

SÉBASTIEN QUIRION

**SEGMENTATION TEMPORELLE DE  
MOUVEMENTS CYCLIQUES HUMAINS À  
PARTIR D'UNE REPRÉSENTATION  
SQUELETTIQUE**

Mémoire présenté  
à la Faculté des études supérieures de l'Université Laval  
dans le cadre du programme de maîtrise en génie électrique  
pour l'obtention du grade de maître ès sciences (M. Sc.)

FACULTÉ DES SCIENCES ET DE GÉNIE  
UNIVERSITÉ LAVAL  
QUÉBEC

JUILLET 2006

©Sébastien Quirion, 2006

# Résumé

L'analyse de mouvements par un système de vision numérique fait l'objet de plusieurs recherches depuis quelques années et plusieurs y ont déjà apporté des solutions intéressantes. Toutefois, dans un bon nombre de ces recherches, les solutions proposées fonctionnent sous l'hypothèse que la séquence vidéo analysée ne comporte qu'un seul mouvement, cyclique ou non cyclique, aussi appelé activité. Par contre, dans une application réelle d'analyse de mouvements, qu'il s'agisse d'un système de surveillance automatisé, d'un système produisant automatiquement des résumés de bandes vidéo ou autre, les séquences vidéo analysées peuvent comporter un grand nombre d'activités exécutées les unes après les autres, entrecoupées ou non de pauses.

Dans ce contexte, cet ouvrage propose, analyse et compare plusieurs variantes d'une méthode visant à extraire automatiquement les mouvements cycliques d'une séquence vidéo. Dans un premier temps, on y présente le problème sous-jacent qu'est la segmentation des parties cycliques d'un signal et on y propose des solutions. Dans un second temps, on y traite du problème de représentation des mouvements par des signaux et de la fusion des segmentations obtenues sur ces signaux.

La méthode présentée utilise l'information fournie par un modèle de squelette représentant l'évolution temporelle d'un être humain dans la séquence vidéo. Elle s'applique à n'importe quel format de squelette décrit en terme de joints reliés par des segments de droite et ne requiert aucune connaissance *a priori* sur les mouvements effectués.

# Abstract

Automated motion analysis and recognition has been the object of numerous researches for a good number of years now. Many of these researches generated interesting solutions to the problem, many of which require that the analyzed video sequence contains a single cyclic or non-cyclic motion, also referred to as an activity. This hypothesis is however not consistent with many real life applications where such algorithms could be of use. For instance, automated surveillance systems, automated video summarizers or other such applications would evidently need to process video sequences containing more than one activity, potentially interleaved with pauses.

In this regard, this dissertation presents, analyses and compares many variations of a general approach to the problem of extracting single cyclic activities from video sequences. To that end, we first address the underlying problem of cyclic component segmentation from one-dimensional signals. We then address the problem of motion representation through such signals and the problem of merging a number of signal segmentations in order to obtain a single segmentation coherent with the activity content of the original video sequence.

The method therein presented uses the information provided by a skeleton sequence representing the motion of a human subject. This method can be applied to any type of skeleton described by nodes linked by straight edges. Moreover, no prior knowledge of the activities to be segmented is required.

# Avant-propos

J'aimerais d'abord remercier mon directeur de recherche, Robert Bergevin, toujours disponible et ne ménageant aucun effort pour me soutenir dans mes grandes questions de recherche comme dans mes difficultés administratives. Au-delà de son aide sur les problèmes techniques de ma recherche, j'ai surtout beaucoup appris de sa vision de la recherche scientifique. J'aimerais également remercier ma codirectrice, Alexandra Branzan Albu, toujours prête à me motiver à me lancer des défis. Elle m'a fait comprendre beaucoup sur les aspects concrets de la vie en recherche et sur les moyens pour bien m'y adapter.

Je voudrais également remercier ma famille, qui m'a encouragé et a cru en moi depuis le tout début. Plus particulièrement mes parents, qui ont sacrifié beaucoup pour moi, mais également mon frère et ma soeur, qui seront toujours des modèles à mes yeux. J'aimerais aussi réserver un merci tout spécial à ma copine, Geneviève. Elle a su me soutenir et m'encourager même lorsque le temps me manquait pour m'en occuper convenablement. Prière de ne pas l'avertir tout de suite que la thèse s'annonce encore plus ardue.

Je me dois aussi de reconnaître la grande importance qu'a eue mon laboratoire adoptif, le laboratoire de vision et de systèmes numériques, dans l'issue de mes recherches. Réunissant des gens ayant des expertises dans un nombre impressionnant de domaines de la science et favorisant l'échange d'idées et de connaissances de multiples façons, ce laboratoire est un lieu où peu de questions restent sans réponse bien longtemps. J'aimerais plus particulièrement remercier mes collègues Frédéric Jean et Keaven Pineau avec qui j'ai collaboré de près tout au long de mes recherches, ainsi que Philippe Lambert, qui a relevé les questions les plus pertinentes à mon projet, qui répond toujours à mes interrogations les plus complexes et qui sait reconnaître lorsqu'une visite au Pub s'impose.

Je voudrais aussi remercier les Fonds de recherche sur la nature et les technologies du Québec, ayant financé mes recherches par le biais d'une bourse de maîtrise en recherche.

Sans ce financement, je n'aurais pu me concentrer entièrement à la recherche comme ce fût le cas.

Finalement, *Rei* à O'sensei Steve Dorion qui m'a introduit au monde du karaté et grâce à qui je peux équilibrer mes efforts dans un univers de connaissances et de réflexions par un enseignement de force physique, de dextérité et de réflexes.

*À tous ceux qui m'ont soutenu et encouragé*

*Dave, although you took thorough  
precautions in the pod against my  
hearing you, I could see your lips  
move.*  
*-HAL, « 2001 : A Space Odyssey »*

# Table des matières

Résumé	ii
Abstract	iii
Avant-propos	iv
Table des matières	vii
Liste des tableaux	ix
Table des figures	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Résumé des travaux de recherche . . . . .	4
1.2 Aspects techniques . . . . .	6
<b>2 Fondements et travaux connexes</b>	<b>11</b>
2.1 Définitions et précisions . . . . .	11
2.2 Analyse de périodicité . . . . .	18
2.3 Segmentation temporelle . . . . .	20
<b>3 Segmentation d'un signal</b>	<b>23</b>
3.1 Autocorrélation . . . . .	26
3.2 Indice de périodicité . . . . .	35
3.2.1 Indice de périodicité . . . . .	36
3.2.2 Indice $l_S$ -normalisé . . . . .	38
3.2.3 Analyse d'un indice basé sur la période . . . . .	39
3.3 Recherche du meilleur segment . . . . .	41
3.3.1 Approche exhaustive . . . . .	41
3.3.2 Sous-échantillonnage . . . . .	42
3.3.3 Optimisation numérique . . . . .	46
3.4 Simplification du problème . . . . .	51
3.4.1 Hypothèses simplificatrices . . . . .	52
3.4.2 Soustraction des pauses . . . . .	52

<b>4</b>	<b>Segmentation d'une séquence de squelettes</b>	<b>55</b>
4.1	Segmentation anatomique . . . . .	56
4.2	Sélection des signaux . . . . .	57
4.3	Segmentation d'un ensemble de signaux . . . . .	58
4.4	Fusion des segmentations . . . . .	61
4.4.1	Algorithme de fusion simple . . . . .	62
4.4.2	Algorithme de fusion par différence . . . . .	63
4.5	Analyse au niveau de l'anatomie entière . . . . .	64
4.6	Alternative à la segmentation anatomique . . . . .	66
<b>5</b>	<b>Étude empirique</b>	<b>68</b>
5.1	Topologies squelettiques utilisées . . . . .	68
5.2	Séquences de squelettes utilisées . . . . .	71
5.3	Métriques d'évaluation . . . . .	74
5.4	Choix des paramètres . . . . .	75
5.4.1	Paramètres de soustraction des pauses . . . . .	76
5.4.2	Seuils sur la périodicité . . . . .	79
5.4.3	Paramètres des algorithmes de segmentation . . . . .	80
5.5	Étude empirique des algorithmes . . . . .	85
5.5.1	Algorithmes de fusion . . . . .	87
5.5.2	Algorithmes de recherche du meilleur segment . . . . .	95
<b>6</b>	<b>Conclusion</b>	<b>108</b>
6.1	Développements et travaux futurs . . . . .	110
	<b>Bibliographie</b>	<b>113</b>
<b>A</b>	<b>Logiciel Développé</b>	<b>115</b>



# Liste des tableaux

5.1	Sommaire des séquences vidéo saisies - mouvements contraints . . . . .	72
5.2	Sommaire des séquences vidéo saisies - mouvements naturels . . . . .	74
5.3	Sommaire des paramètres utilisés par les algorithmes présentés . . . . .	75
5.4	Liste des paramètres avec leur valeur respective. Les valeurs temporaires sont indiquées en italique. . . . .	77
5.5	Indice $I_S$ moyen obtenu pour l'ensemble des tests pour différentes valeurs des paramètres $\beta_P$ et $\epsilon$ . . . . .	78
5.6	Temps cumulatif pris pour l'exécution de l'ensemble des tests pour différentes valeurs des paramètres $\beta_P$ et $\epsilon$ . . . . .	79
5.7	Indice $I_S$ moyen obtenu pour l'ensemble des tests pour différentes valeurs du paramètre $\eta_E$ . . . . .	80
5.8	Indice $I_S$ moyen obtenu pour l'ensemble des tests pour différentes valeurs du paramètre $\eta_B$ . . . . .	81
5.9	Indice $I_S$ moyen et temps cumulatif pour l'ensemble des tests pour différentes valeurs du paramètre $\Delta_U$ . . . . .	82
5.10	Indice $I_S$ moyen obtenu pour l'ensemble des tests pour différentes valeurs des paramètres $D$ et $\gamma$ . . . . .	83
5.11	Temps cumulatif pris pour l'exécution de l'ensemble des tests pour différentes valeurs des paramètres $D$ et $\gamma$ . . . . .	84
5.12	Indice $I_S$ moyen et temps cumulatif pour l'ensemble des tests pour différentes valeurs du paramètre $N$ . . . . .	85
5.13	Liste des paramètres avec leur valeur fixes respectives. . . . .	86
5.14	Avec segmentation anatomique - Indice $I_S$ , précision et rappel obtenu par l'algorithme de fusion simple (Simple) et l'algorithme de fusion par différence (Diff.) appliqué à chaque séquence de test. . . . .	88
5.15	Sans segmentation anatomique - Indice $I_S$ , précision et rappel obtenu par l'algorithme de fusion simple (Simple) et l'algorithme de fusion par différence (Diff.) appliqué à chaque séquence de test. . . . .	91
5.16	Avec segmentation anatomique - Indice $I_S$ obtenu par chacun des algorithmes appliqué à chaque séquence de test. . . . .	98

5.17 Avec segmentation anatomique - Rappel obtenu par chacun des algorithmes appliqué à chaque séquence de test. . . . .	99
5.18 Avec segmentation anatomique - Précision obtenue par chacun des algorithmes appliqué à chaque séquence de test. . . . .	99
5.19 Avec segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test. . . . .	100
5.20 Sans segmentation anatomique - Indice $I_S$ obtenu par chacun des algorithmes appliqué à chaque séquence de test. . . . .	101
5.21 Sans segmentation anatomique - Rappel obtenu par chacun des algorithmes appliqué à chaque séquence de test. . . . .	102
5.22 Sans segmentation anatomique - Précision obtenue par chacun des algorithmes appliqué à chaque séquence de test. . . . .	103
5.23 Sans segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test. . . . .	104
5.24 Sans segmentation anatomique - Indice $I_S$ obtenu par chacun des algorithmes appliqué à un sous-ensemble des séquences de test en utilisant l'approche exhaustive comme segmentation de référence. . . . .	106
5.25 Sans segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test. . . . .	107

# Table des figures

1.1	<b>a)</b> Exemple de cycle simple avec un seul maximum local et un seul minimum local. <b>b)</b> Exemple de cycle simple avec un seul plateau local et une seule vallée locale. <b>c)</b> Un autre exemple de cycle simple où le minimum peut être placé à la fin ou au début du cycle, mais demeure unique. . .	4
1.2	<b>a)</b> Exemple de cycle complexe comportant deux maxima et deux minima locaux. <b>b)</b> Exemple de cycle complexe comportant deux plateaux locaux et deux vallées locales. <b>c)</b> Exemple de cycle complexe comportant un palier.	4
1.3	<b>a)</b> Exemple d'une segmentation de référence pour un signal. La couleur n'est utilisée que pour distinguer des segments adjacents. <b>b)</b> Exemple d'une segmentation quelconque. Sous un seuil $\tau = 75\%$ , elle affiche une précision de 100% et un rappel de 100%. <b>c)</b> Autre exemple de segmentation obtenant une précision de 50% et un rappel de 33%. . . . .	7
1.4	<b>a), b) et c)</b> Trois exemples différents de paires de segments donnant toutes une même valeur d'indice d'appariement. Les zones bleues et rouges représentent les segments en question, les rectangles blancs représentent l'union de ces segments et les rectangles noirs en représentent l'intersection. . . . .	9
1.5	<b>a)</b> Exemple d'une segmentation de référence pour un signal. La couleur n'est utilisée que pour distinguer des segments adjacents. <b>b) et c)</b> Deux exemples distincts de segmentations ayant une similarité de 75% avec la segmentation de référence. <b>d)</b> Exemple de segmentation significativement erronée. Le troisième segment obtient le plus haut indice d'appariement simultanément pour les deux derniers segments de référence, il prend donc la valeur de l'indice le plus élevé des deux. Le segment de référence restant obtient une valeur de pénalité. . . . .	10
2.1	Exemple d'un signal cyclique par modulation temporelle. Le tracé noir dénote le signal. Les flèches noires et rouges dénotent respectivement la durée moyenne $c$ d'un cycle et l'ajustement $\xi_c(t)$ à cette durée pour le temps $t$ spécifié. . . . .	15

2.2	Exemple d'un signal cyclique par modulation d'états. Le tracé noir dénote le signal et le tracé rouge dénote le signal décalé de $c$ . Les zones jaunes représentent l'intégrale des distances entre deux cycles consécutifs. . . . .	16
2.3	Exemple d'un signal cyclique par modulation temporelle et modulation d'états. Le tracé noir dénote le signal et le tracé rouge dénote le signal décalé de $c$ et modulé par les écarts $\xi_c(t)$ . Les zones jaunes représentent l'intégrale des distances entre deux cycles consécutifs temporellement modulés. . . . .	17
3.1	Évolution de la trace de l'autocorrélation $A_S$ en fonction du décalage $d$ pour un signal sinusoïdal. Pour chaque valeur de $d$ , une trace partielle de l'autocorrélation et les signaux $S(x)$ et $S(x + d)$ sont illustrés. La zone ombragée grise correspond au domaine de sommation de l'autocorrélation.	30
3.2	Signaux basés sur des cycles simples et la trace de leur autocorrélation. Chaque signal est précédé de ses propriétés : cycle simple (CS), continuellement croissant/décroissant (CC), symétrie d'amplitude (SA) et symétrie de phase (SP). Un carré vert indique que le signal possède cette propriété alors qu'un carré rouge indique l'inverse. . . . .	32
3.3	Exemples de signaux basés sur des cycles complexes dont l'autocorrélation ne comporte aucun maximum indésirable. <b>a)</b> Signal comportant plus d'un maximum par cycle. <b>b)</b> Signal comportant des paliers. <b>c)</b> Signal comportant plus d'un maximum par cycle et comportant des paliers. . . . .	33
3.4	Exemples de signaux basés sur des cycles complexes dont l'autocorrélation comporte des maxima indésirables. <b>a)</b> Signal comportant plus d'un maximum par cycle. <b>b)</b> Signal comportant un palier. <b>c)</b> Signal comportant plus d'un maximum par cycle et comportant des paliers. . . . .	34
3.5	<b>a)</b> Un exemple de signal cyclique par modulation d'états. <b>b)</b> La trace de l'autocorrélation de ce signal. La zone ombragée recouvre le dernier dixième des valeurs d'autocorrélation. Ces valeurs peuvent être instables et sont donc ignorées. . . . .	35
3.6	<b>a)</b> Un exemple de signal cyclique par modulation temporelle. <b>b)</b> La trace de l'autocorrélation de ce signal. Les flèches indiquent pour chaque maximum où il se trouverait pour un signal périodique. . . . .	36
3.7	<b>a)</b> Un exemple de signal cyclique selon la définition générale. <b>b)</b> La trace de l'autocorrélation de ce signal. . . . .	36
3.8	<b>a)</b> Exemple de signal à segmenter. <b>b)</b> Trace de l'équation (2.12) pour tous les segments $[i, j]$ possibles sur le signal présenté en <b>a)</b> . La correspondance entre les couleurs et les valeurs est présentée sur l'échelle à la droite de cette matrice. <b>c)</b> Trace de l'indice $\Psi'_{S[i,j]}$ pour tous les segments $[i, j]$ possibles sur le signal présenté en <b>a)</b> . Les couleurs correspondent aux mêmes valeurs qu'en <b>b)</b> . . . . .	40

3.9 a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par une approche exhaustive sur ce signal. b) Matrice des valeurs calculées par l'approche exhaustive, correspond à la trace de l'indice  $l_S$ -normalisé pour tous les segments possibles. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches. 42

3.10 a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par sous-échantillonnage uniforme avec  $\Delta_U = 5$ . b) Matrice des valeurs calculées par sous-échantillonnage uniforme avec  $\Delta_U = 5$ . Les points ont été grossis pour en améliorer la lisibilité. La zone noire représente les valeurs inconnues. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches. . . . . 44

3.11 a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par sous-échantillonnage sélectif avec  $D = 4$  et  $\gamma = 0.75$ . b) Matrice des valeurs calculées par sous-échantillonnage sélectif. La zone noire représente les valeurs inconnues. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches. . . . . 47

3.12 a) Indice  $l_S$ -normalisé calculé pour tous les segments d'un signal sinusoïdal. Nous remarquons que le maximum se retrouve aux limites de la fonction. b) Schéma illustrant comment la réflexion de la fonction est effectuée. Seules ces trois réflexions sont nécessaires au bon fonctionnement de notre algorithme. c) Le résultat des réflexions appliqué à la matrice présentée en a). Nous remarquons que le maximum ne se retrouve plus sur une frontière. . . . . 49

3.13 a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par une itération de l'algorithme DHC. b) Trace de la fonction  $Y_{S_{[i,j]}}$  pour le signal donné en a). Le tracé blanc représente les étapes d'optimisation de l'algorithme DHC. c) Tracé partiel de la fonction, permettant de mieux voir les étapes effectuées par l'algorithme DHC. . . . 51

4.1 En a), les sommets et arcs tracés en bleu représentent le sous-graphe défini pour le torse. En b), les sommets et arcs tracés en bleu représentent les sous-graphes définis pour chaque membre de la topologie squelettique. 57

4.2 Illustration des signaux utilisés pour caractériser les mouvements globaux. a) Mouvement de saut vers la droite. b) Approximation des signaux formés à partir des positions en abscisse et en ordonnée pour chaque noeud du torse. c) Approximation du signal formé à partir de l'angle absolu de l'arc contenu dans le torse. . . . . 59

4.3	Illustration des signaux utilisés pour caractériser les flexions et les extensions d'une articulation. <b>a)</b> Flexion de l'épaule. <b>b)</b> Approximation des signaux extraits de l'articulation de l'épaule pour le mouvement <b>a)</b> . <b>c)</b> Extension du coude. <b>d)</b> Approximation des signaux extraits de l'articulation du coude pour le mouvement <b>c)</b> . . . . .	60
5.1	Topologie squelettique proposée dans [14]. Une silhouette grise est placée en arrière-plan en référence. . . . .	69
5.2	Topologie squelettique issue de l'approche de suivi des membres proposée dans [6]. Une silhouette grise est placée en arrière-plan en référence. . .	70
5.3	Exemples de séquences vidéo pour chaque type de séquence. Pour chaque trame clé montrée, quelques trames précédentes ont été superposées avec un niveau de transparence pour aider à percevoir le mouvement effectué.	73
5.4	Segmentation anatomique de la topologie squelettique proposée dans [14]. Les éléments en rouge représentent les membres alors que les éléments en bleu composent le torse. La base du cou fait à la fois partie du torse et du <i>membre</i> de la tête. . . . .	87
5.5	Segmentation de référence, segmentation des signaux et segmentation finale pour les mouvements du bras gauche dans la séquence <i>VignA3</i> . Le bleu et le rouge sont utilisés en alternance comme couleur de fond pour représenter les segments trouvés lors de la segmentation des signaux. Les crochets verts indiquent les segments retenus lors de la fusion alors que les <i>X</i> rouges indiquent les segments éliminés. . . . .	90
5.6	Segments cycliques trouvés par la segmentation automatique sur des signaux représentant le mouvement des jambes dans la séquence <i>VignA3</i> .	92
5.7	Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence <i>VignC3</i> en utilisant l'algorithme de fusion simple. Les crochets verts indiquent les segments retenus lors de la fusion alors que les <i>X</i> rouges indiquent les segments éliminés. . .	93
5.8	Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence <i>VignC3</i> en utilisant l'algorithme de fusion par différence. L'encadré vert représente un segment retenu qui a été obtenu par l'algorithme de différence. . . . .	94
5.9	Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence <i>JeanC3</i> en utilisant l'algorithme de fusion simple. . . . .	96
5.10	Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence <i>JeanC3</i> en utilisant l'algorithme de fusion par différence. L'encadré vert représente un segment retenu qui a été obtenu par l'algorithme de différence. . . . .	97

A.1	Menu principal . . . . .	116
A.2	Page d'analyse d'indices de périodicité . . . . .	117
A.3	Page de sélection de signal . . . . .	117
A.4	Étape par étape - Page de sélection de la séquence de squelettes . . . . .	118
A.5	Étape par étape - Page de segmentation des signaux . . . . .	119
A.6	Étape par étape - Page de fusion . . . . .	120
A.7	Page de traitements de lot . . . . .	121
A.8	Page dynamique d'ajustement des paramètres d'un algorithme . . . . .	121

# Chapitre 1

## Introduction

Bien que l'analyse automatisée d'images et de séquences vidéo ne soit pas un domaine des plus récents, il a beaucoup gagné en popularité auprès des chercheurs et des compagnies au cours de la dernière décennie. Cela est sans doute attribuable à l'évolution de la puissance de calcul des ordinateurs au cours des dernières années, qui a rendu ce domaine à la fois accessible et profitable. Alors qu'il est relativement simple de doter un ordinateur d'un sens de la vue comparable à celui de l'humain, en lui annexant une caméra vidéo par exemple, de plus en plus de scientifiques des quatre coins du monde s'affairent aujourd'hui à lui développer la capacité d'extraire des informations utiles de ce sens : ils s'affairent à développer la vision numérique. Une motivation générale derrière ce mouvement est certainement que la vision numérique permet d'envisager l'automatisation éventuelle des tâches de vision cognitive, un ensemble de tâches autrefois réservées aux humains.

Alors que l'analyse d'images connaît de plus en plus d'applications stables et d'utilisation courante, telles que la reconnaissance de visages, l'analyse de séquences vidéo en est, pour la plus grande partie, toujours à ses premiers balbutiements. Dans ce sous-domaine de la vision numérique, une tangente de recherche de plus en plus exploitée est l'analyse de mouvements. L'engouement croissant pour de telles recherches est en grande partie attribuable au grand nombre d'applications pratiques qui pourraient en découler. Il existe déjà des prototypes de systèmes de surveillances automatisés pour assister des opérateurs humains [5], de systèmes produisant automatiquement des résumés de bandes vidéo [4] et de systèmes de reconnaissance de personnes basés sur la démarche [13] pour n'en nommer que quelques-uns. De plus, lorsque ces technologies deviendront suffisamment stables, il sera possible d'envisager des systèmes beaucoup plus complexes tels que des systèmes d'assistance à la réadaptation motrice ou encore des entraîneurs sportifs virtuels qui pourront corriger les mouvements de leurs usagers.



À ce jour, une limitation couramment rencontrée dans les diverses solutions proposées par les recherches en analyse de mouvements est que la séquence vidéo analysée ne doit contenir qu'une seule activité se déroulant du début à la fin de la séquence. Cette condition est particulièrement fréquente dans les systèmes de description et de reconnaissance d'activités [10][2][1]. Elle ne correspond toutefois pas au cas général d'utilisation de telles applications, où les séquences vidéo peuvent comporter un grand nombre d'activités exécutées les unes après les autres, entrecoupées ou non de pauses. Dans plusieurs cas, l'intervention humaine requise pour isoler les activités dans les séquences d'origine viendrait annuler tout gain de productivité espéré par l'automatisation de la tâche. Dans cette optique, il devient important de se doter d'un mécanisme de segmentation automatique d'activités.

Notons que, puisque nous traitons de mouvements humains, nous utiliserons souvent le terme *activité* en référence à ces mouvements. Bien que Polana et Nelson [9] l'utilisent exclusivement pour parler de mouvements cycliques, nous utiliserons ce terme indépendamment en le spécifiant à chaque fois lorsque nous traitons d'activités cycliques.

Alors que peu de chercheurs se sont attardés spécifiquement à ce problème, certains y ont apporté des solutions adaptées à leurs besoins spécifiques. À titre d'exemple, Bobick et Davis présentent dans [2] une méthode de segmentation temporelle basée sur une base de connaissances. Cette approche de segmentation temporelle, fortement couplée à la reconnaissance d'activités, consiste à comparer plusieurs segments de durées variables et de comparer leur contenu en mouvement aux modèles d'activités connus. La segmentation temporelle est alors implicite dans la reconnaissance. Min et Kasturi, quant à eux, présentent dans [8] une segmentation temporelle préalable basée sur le nombre de membres en mouvement à un temps donné. Ils considèrent que le nombre de membres en mouvement demeure constant à l'intérieur d'une activité et forment des segments temporels avec les trames contiguës comportant un même nombre de membres en mouvement. Cette segmentation est ensuite raffinée par une approche couplant la segmentation temporelle et la reconnaissance de mouvements, à l'image de l'approche de Bobick et Davis.

Puisqu'elles se basent sur des bases de connaissances, ces solutions ne peuvent remplir une tâche générique de segmentation temporelle d'activités. Parmi la littérature scientifique accessible, peu nombreux sont les chercheurs qui ont attaqué ce problème précis. Bien que les solutions proposées soient intéressantes, elles posent tout de même certaines limitations, notamment sur le type d'activité. Nous sommes d'avis que l'utilisation d'une approche de plus haut niveau serait plus à même de contourner ces limitations.

Suivant ces observations, notre objectif principal est de répondre à la problématique posée, soit de fournir une segmentation temporelle précise des activités dans une séquence vidéo avec des résultats comparables à ceux qu'un opérateur humain obtiendrait. Bien que l'approche proposée dans cet ouvrage soit suffisamment générique, il existe tout de même quelques contraintes sur le contexte dans lequel elle est applicable :

**Séquences de squelettes** L'approche proposée ne procède pas à la segmentation temporelle directement à partir de la séquence vidéo, mais segmente plutôt une séquence de squelettes représentant les mouvements d'une personne dans la séquence. Ceci pose donc la contrainte qu'il doit être possible d'extraire un squelette représentatif d'un sujet humain pour chaque image de la séquence. Cette approche de haut niveau offre plusieurs avantages, notamment elle assure une robustesse aux autres mouvements dans la scène et elle permet de traiter les mouvements de plusieurs personnes indépendamment. Il va de soi que le problème n'en est pas plus simple, mais cette approche de haut niveau vient scinder le problème en deux : localiser, isoler et décrire la position de la personne dans un premier temps, puis procéder à la segmentation des activités dans un deuxième temps<sup>1</sup>.

**Mouvements cycliques** Notre approche ne s'applique qu'à la segmentation de mouvements cycliques. En effet, comme nous ne connaissons pas *a priori* les mouvements à segmenter, nous n'en possédons pas de modèles. Sans de tels modèles, il est virtuellement impossible de procéder à une segmentation précise ou même approximative de mouvements non cycliques. Notons toutefois qu'un grand nombre d'activités humaines d'intérêt sont cycliques, surtout au niveau de la locomotion. Par contre, lorsque les activités à segmenter sont connues et en petit nombre, il peut être préférable d'utiliser une méthode adaptée, telle que celles présentées dans [2] et [8]. Une combinaison de notre approche avec une telle méthode pourrait aussi être envisagée. Une telle approche hybride permettrait alors de segmenter tout mouvement cyclique ainsi qu'un certain nombre de mouvements non cycliques, en fonction des activités pour lesquelles des modèles sont connus. Ces approches hybrides ne sont toutefois pas étudiées davantage dans cet ouvrage.

**Nombre minimum de cycles** Les mouvements pouvant être segmentés doivent comporter un minimum de trois répétitions d'un cycle. Bien que la définition théorique stipule que seul un minimum de deux répétitions est requis pour décider si un mouvement est cyclique<sup>2</sup>, en pratique, nos méthodes ne sont pas assez précises pour atteindre cette limite théorique. Alors qu'un bon nombre d'erreurs peuvent être commises avec notre approche lorsque nous acceptons un minimum de deux répétitions, poser une limite de trois répétitions permet une redondance importante

---

<sup>1</sup>Le cadre de cette recherche ne couvre pas l'élaboration d'un modèle de squelette, mais utilise plutôt des modèles existants. Notre recherche se concentre donc uniquement sur la deuxième étape, soit la segmentation temporelle précise à partir d'une séquence de squelettes.

<sup>2</sup>voir section 2.1

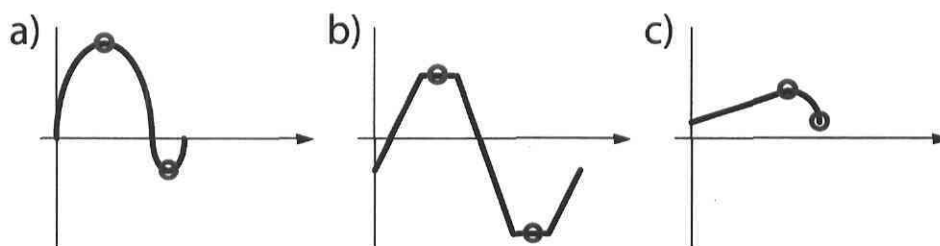


FIG. 1.1 – a) Exemple de cycle simple avec un seul maximum local et un seul minimum local. b) Exemple de cycle simple avec un seul plateau local et une seule vallée locale. c) Un autre exemple de cycle simple où le minimum peut être placé à la fin ou au début du cycle, mais demeure unique.

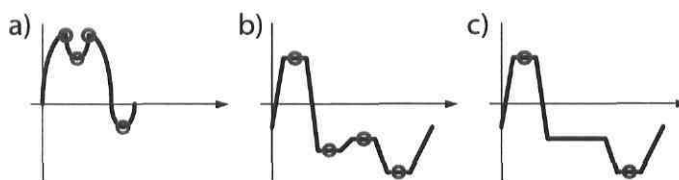


FIG. 1.2 – a) Exemple de cycle complexe comportant deux maxima et deux minima locaux. b) Exemple de cycle complexe comportant deux plateaux locaux et deux vallées locales. c) Exemple de cycle complexe comportant un palier.

dans notre analyse et élimine presque entièrement ces erreurs.

**Mouvements simples** Le cycle de chaque mouvement doit être simple, c'est-à-dire qu'il doit comporter un seul maximum (ou plateau) local, un seul minimum (ou vallée) local et ne comporter aucun palier. Bien que l'approche puisse donner des résultats satisfaisants pour des mouvements cycliques complexes, elle n'en fournit aucune garantie. La section 3.1 étudie cette question plus en détail. Les figures 1.1 et 1.2 illustrent respectivement des exemples de cycles simples et des exemples de cycles complexes pour un signal<sup>3</sup>.

## 1.1 Résumé des travaux de recherche

Nos travaux de recherche sont fortement basés sur l'analyse de signaux. L'idée directrice est plutôt simple : dans la mesure où nous pouvons obtenir un signal représentant fidèlement les mouvements, le problème énoncé se réduit alors à segmenter les portions cycliques d'un signal. Dans les faits, il serait surprenant de pouvoir trouver un

<sup>3</sup>Dans le cadre de cet ouvrage, le terme *signal* désignera toujours un signal discret unidimensionnel de longueur finie.

signal représentant à lui seul les mouvements des divers membres d'une personne. Nous avons donc décidé d'extraire plusieurs signaux pour représenter ces mouvements le plus adéquatement possible.

Pour arriver à nos fins à partir d'un tel ensemble de signaux, nous devons pouvoir segmenter adéquatement les portions cycliques des signaux et fusionner l'information de ces segmentations de façon à ce qu'elle puisse composer une segmentation pour l'ensemble de la séquence. Nos recherches sur le problème de la segmentation des portions cycliques d'un signal sont présentées au chapitre 3 alors que nos recherches sur le problème de la fusion des segmentations et sur le problème de sélection de l'ensemble de signaux sont présentées au chapitre 4.

Au lieu de proposer une solution unique, nous avons préféré développer et tester plusieurs algorithmes différents pour résoudre le problème de la segmentation d'un signal ainsi que celui de la fusion des segmentations. Cette approche nous a permis de conserver une vision plus globale de chacun de ces deux problèmes et de mieux en comprendre les subtilités. Elle nous a également permis de confirmer nos idées premières quant aux algorithmes idéaux ou nous a invités à les réviser.

Lorsque venait le temps de planifier nos recherches, deux règles d'or ont guidé nos diverses décisions. Premièrement, la solution développée devait être la plus générique possible. Elle ne devait comporter qu'un nombre minimum de contraintes, permettre l'utilisation de n'importe quel modèle de squelette et pouvoir détecter n'importe quelle activité cyclique. Deuxièmement, les algorithmes retenus devaient utiliser aussi peu de seuils que possible. Étant donné que la nature du problème est d'automatiser une tâche, il ne serait pas vraiment profitable qu'un opérateur humain doive ajuster des seuils pour chaque séquence traitée. Bien que les seuils soient pratiquement inévitables dans un contexte de vision numérique, nous avons tenté d'illustrer, à l'aide d'expérimentations, que les seuils utilisés dans nos divers algorithmes peuvent conserver une valeur fixe et fournir de bons résultats, peu importe la séquence traitée.

Bien qu'une solution idéale de segmentation temporelle puisse s'exécuter en temps réel, il ne s'agissait pas d'un des critères guidant nos recherches. Étant donné que nos recherches sont relativement exploratrices sur le problème de la segmentation temporelle d'activités cycliques, peu de recherches ayant été menées sur le sujet, nous avons jugé qu'il était plus important de s'attarder à assurer une bonne efficacité de la solution que d'assurer une exécution rapide.

## 1.2 Aspects techniques

Les algorithmes présentés plus loin ont été testés empiriquement. Pour ce faire, plusieurs séquences vidéo d'une personne en mouvement ont été saisies. En fonction de la difficulté de segmentation qu'elles présentent, ces séquences ont été classées en trois groupes distincts. Le type A regroupe les séquences où toutes les activités cycliques sont entrecoupées de pauses. Il représente évidemment le type de séquence le plus simple à segmenter. Le type B regroupe les séquences où au moins une activité cyclique est temporellement adjacente à une autre activité, cyclique ou non. Finalement, le type C regroupe les séquences où au moins une activité cyclique se transforme en une autre activité cyclique avec une continuité dans le mouvement. Par exemple, une séquence où une personne salue d'une main immédiatement avant de saluer des deux mains serait de type C.

Afin de pouvoir évaluer la qualité des segmentations obtenues automatiquement, trois métriques d'évaluation ont été développées : la précision, le rappel et un indice de similarité. Ces trois métriques se basent sur une segmentation de référence. Puisque nous désirons obtenir une segmentation comparable à ce qu'un opérateur humain pourrait faire, il est normal d'utiliser une segmentation manuelle effectuée par un humain comme segmentation de référence. Afin de s'assurer qu'une telle segmentation humaine était suffisamment objective, qu'elle ne variait pas significativement selon les personnes, chaque séquence a été temporellement segmentée manuellement par dix volontaires. Comme les écarts-types des débuts et fins des segmentations manuelles ne différaient pas significativement, ayant un écart-type toujours inférieur à un cinquième de seconde, nous avons conclu que la moyenne de ces segmentations manuelles constituait une segmentation de référence adéquate.

Pour un pourcentage  $\tau$  donné, si au-delà de  $\tau\%$  de la couverture temporelle d'un segment obtenu recouvre un segment de référence, l'activité cyclique représentée par le segment de référence est alors considérée détectée. Les activités cycliques contenues dans la séquence ne faisant pas l'objet d'une telle détection sont désignées détections manquées. Leur compte est toujours égal au nombre d'activités cycliques dans la segmentation de référence soustrait du nombre de bonnes détections. Finalement, les segments obtenus qui ne forment pas une bonne détection ou qui détectent une activité cyclique déjà détectée par un autre segment sont considérés être des fausses détections. Leur compte est toujours égal au nombre de segments obtenus soustrait du nombre de bonnes détections. Les décomptes peuvent être faits pour différentes valeurs de  $\tau$  pour obtenir une meilleure idée de la performance de l'algorithme étudié.

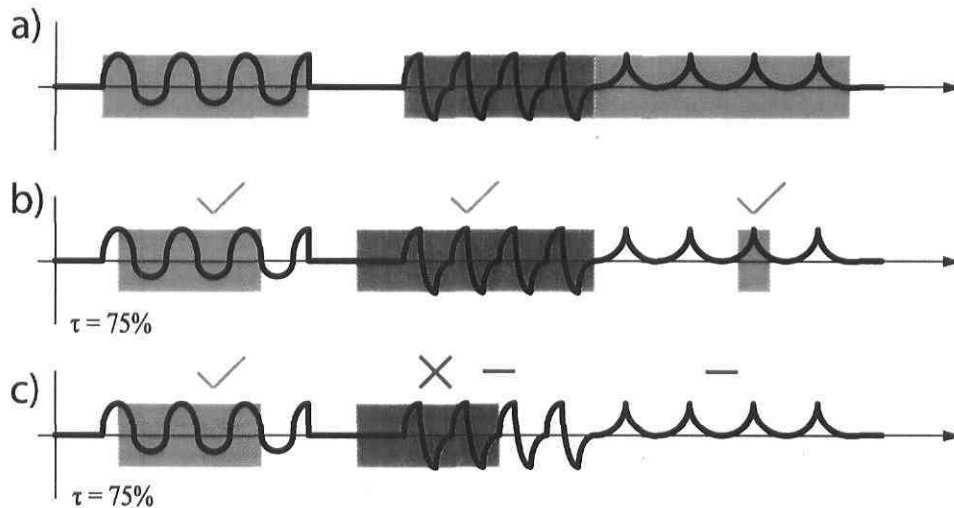


FIG. 1.3 – a) Exemple d'une segmentation de référence pour un signal. La couleur n'est utilisée que pour distinguer des segments adjacents. b) Exemple d'une segmentation quelconque. Sous un seuil  $\tau = 75\%$ , elle affiche une précision de 100% et un rappel de 100%. c) Autre exemple de segmentation obtenant une précision de 50% et un rappel de 33%.

Notre première métrique, la précision, est un dérivé de la mesure du même nom dans la littérature en informatique. Elle indique le rapport entre le nombre de bonnes détections et le nombre total de segments contenus dans la segmentation. Le rappel, quant à lui, indique le rapport entre le nombre de bonnes détections et le nombre total de segments dans la segmentation de référence. En d'autres termes, la précision nous indique à quel point les segments obtenus sont fiables alors que le rappel nous indique à quel point toutes les activités cycliques de la séquence ont été détectées. La figure 1.3 illustre ces métriques.

Ces métriques sont intéressantes lorsque nous cherchons à savoir si une activité cyclique a lieu sans vouloir connaître précisément à quel moment elle commence et à quel moment elle se termine.

La dernière métrique est un indice de similarité entre la segmentation obtenue et la segmentation de référence. Cet indice est contenu dans l'intervalle  $[0, 1]$ , la valeur de 1 représentant une équivalence parfaite entre la segmentation obtenue et la segmentation de référence. En quelques mots, il indique à quel point toutes les activités cycliques ont été localisées précisément, complètement et sans erreurs. Cet indice intègre donc une notion de rappel et de précision, au sens du nombre de bonnes détections sur le nombre total des détections, mais également une notion de localisation temporelle précise des

segments. Considérons l'indice d'appariement  $I_A$  entre deux segments  $s_1$  et  $s_2$ <sup>4</sup> :

$$I_A(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|} \quad (1.1)$$

Cet indice est donc le cardinal de l'intersection des deux segments divisé par le cardinal de l'union des deux segments. La figure 1.4 illustre différents aspects de l'indice  $I_A$ . En utilisant cet indice d'appariement  $I_A$ , nous pouvons calculer l'indice de similarité  $I_S$  pour  $SEG$ , une segmentation obtenue automatiquement, et  $SEG^*$ , la segmentation de référence, selon les étapes suivantes :

1. Pour chaque segment de référence  $s^* \in SEG^*$ , trouver le segment  $s \in SEG$  maximisant l'indice  $I_A(s, s^*)$ . Pour de tels segments  $s$ , définir une valeur  $V_s$  comme suit :

$$V_s = \frac{I_A(s, s^*)}{|SEG^*|} \quad (1.2)$$

où  $|SEG^*|$  est le nombre de segments dans  $SEG^*$ . Si une valeur  $V_s$  est ainsi définie par plus d'un  $s^* \in SEG^*$ , conserver la valeur la plus élevée des deux ;

2. Pour tous les autres  $s \in SEG$ , définir une valeur de pénalité  $V_s$  comme suit :

$$V_s = \frac{-1}{2 \times |SEG^*|} \quad (1.3)$$

3. Sommer  $V_s$  pour tous les  $s \in SEG$ . Si la somme est négative, lui attribuer la valeur de 0 :

$$I_S(SEG, SEG^*) = \begin{cases} \sum_{s \in SEG} V_s & \text{si } \sum_{s \in SEG} V_s > 0; \\ 0 & \text{sinon.} \end{cases} \quad (1.4)$$

Cette métrique, quant à elle, est intéressante lorsque nous désirons segmenter parfaitement les activités cycliques d'une séquence vidéo. Dans le cas où la segmentation obtenue comporte autant de segments que la segmentation de référence et s'en approche suffisamment, l'indice donne une bonne idée de la similarité des deux segmentations. Lorsque la segmentation obtenue comporte moins de segments que la référence, la définition de  $V_s$  à la première étape indique que l'indice de similarité ne peut dépasser la fraction des activités cycliques détectées par rapport au nombre total d'activités cycliques. Par exemple, si un seul segment a été obtenu et que la segmentation de référence en comporte quatre, l'indice  $I_S$  ne dépassera pas 25%. Lorsque la segmentation obtenue comporte plus de segments que la segmentation de référence, l'indice de similarité ne peut dépasser 1 moins la moitié du quotient des segments supplémentaires sur le nombre de segments de référence. Selon ces observations, nous pouvons anticiper que l'indice de similarité nous offrira une mesure plus précise pour les segmentations s'approchant

<sup>4</sup>Un segment est ici considéré comme l'ensemble des trames qu'il recouvre.

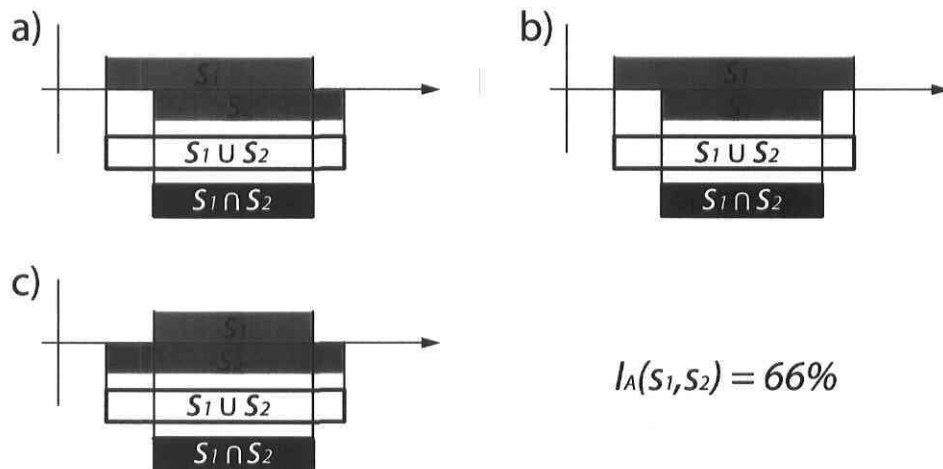


FIG. 1.4 – a), b) et c) Trois exemples différents de paires de segments donnant toutes une même valeur d'indice d'appariement. Les zones bleues et rouges représentent les segments en question, les rectangles blancs représentent l'union de ces segments et les rectangles noirs en représentent l'intersection.

suffisamment de la segmentation de référence que pour celles comportant des lacunes importantes, telles qu'un nombre trop petit ou trop grand de segments. Bien que cela puisse être vu comme une faille de cet indice, une plus grande subtilité est généralement plus utile dans les solutions satisfaisantes, alors que nous ne nous intéressons généralement pas à savoir quelle solution est la meilleure entre deux solutions significativement erronées. La figure 1.5 illustre cette métrique.



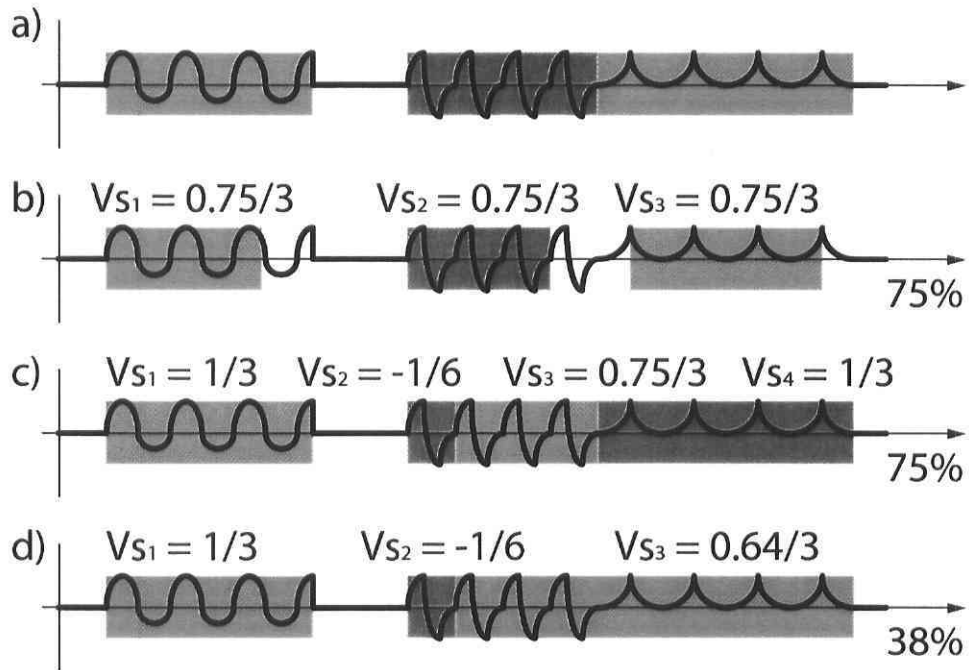


FIG. 1.5 – a) Exemple d'une segmentation de référence pour un signal. La couleur n'est utilisée que pour distinguer des segments adjacents. b) et c) Deux exemples distincts de segmentations ayant une similarité de 75% avec la segmentation de référence. d) Exemple de segmentation significativement erronée. Le troisième segment obtient le plus haut indice d'appariement simultanément pour les deux derniers segments de référence, il prend donc la valeur de l'indice le plus élevé des deux. Le segment de référence restant obtient une valeur de pénalité.

# Chapitre 2

## Fondements et travaux connexes

Ce chapitre se veut la pierre d'assise de cet ouvrage. Il explicite les principes et les connaissances nécessaires à une bonne compréhension des chapitres ultérieurs. Il tente également de bien situer le lecteur quant au contexte dans lequel nos recherches s'inscrivent.

### 2.1 Définitions et précisions

Plusieurs termes reviendront régulièrement dans le présent ouvrage. Pour éviter de susciter différentes interprétations lors de leur usage de même que pour se familiariser avec ces concepts, il convient de définir formellement les termes les plus importants et de les expliquer avant de procéder au vif du sujet.

Tout d'abord, définissons ce qu'est une fonction temporelle, soit un concept regroupant tout ce que nous sommes susceptibles de vouloir segmenter.

#### **Définition 2.1.1 : Fonction Temporelle**

*Dans le cadre de cet ouvrage, nous utiliserons le terme fonction temporelle pour désigner toute fonction dont l'état dépend du temps. Une fonction temporelle  $f$  est définie pour toutes les valeurs d'un intervalle  $\mathbb{T}_f$ . L'intervalle  $\mathbb{T}_f$  peut être fini ou infini et est défini soit sur  $\mathbb{N}$  ou  $\mathbb{Z}$ , dans le cas discret, ou sur  $\mathbb{R}$  dans le cas continu. L'état d'une fonction temporelle  $f$  au temps  $t \in \mathbb{T}_f$  s'exprime  $f(t)$ . Puisqu'il s'agit d'une fonction,  $f(t)$  est toujours unique. Dans le cas générique,  $\mathbb{E}_f$  désigne le domaine des états de  $f$ .*

Voici quelques exemples de fonctions temporelles auxquelles nous référerons :

**Signal** Un signal est une suite de nombres réels. Dans le cadre de cet ouvrage, son domaine sera toujours discret. Un signal se représente sous la forme  $f_S : \mathbb{T}_f \mapsto \mathbb{R}$  où  $\mathbb{T}_f$  est soit un intervalle sur  $\mathbb{R}$  ou sur  $\mathbb{Z}$ ;

**Séquence vidéo** Une séquence vidéo est une suite d'images. Son domaine est réputé discret d'emblée. Soit  $\mathbb{I}_{n \times m}$ , l'ensemble des images d'une résolution de  $n \times m$ , nous pouvons représenter la séquence vidéo sous la forme  $f_V : \mathbb{T}_f \mapsto \mathbb{I}_{n \times m}$  où  $\mathbb{T}_f$  est un intervalle sur  $\mathbb{Z}$ ;

**Ensemble de  $n$  signaux** L'état d'un ensemble de  $n$  signaux est un vecteur de  $n$  nombres réels. Nous représentons donc un tel ensemble sous la forme  $f_E : \mathbb{T}_f \mapsto \mathbb{R}^n$  où  $\mathbb{T}_f$  est soit un intervalle sur  $\mathbb{R}$  ou sur  $\mathbb{Z}$ .

Comme nous nous attarderons à segmenter des activités cycliques ou périodiques, il convient de bien définir ces termes et leurs dérivés. Définissons d'abord le terme *cycle* puisque d'autres définitions en découlent.

### Définition 2.1.2 : Cycle

*Un cycle est une séquence finie et contiguë d'états d'une fonction temporelle  $f$  qui se répète identiquement et de façon contiguë tout au long de  $f$ . Pour s'assurer de l'existence d'un cycle,  $f$  doit contenir plus d'une occurrence dudit cycle.*

Cette définition correspond bien au cas périodique. Dans le cas cyclique, par contre, nous verrons que le cycle peut être modulé de diverses façons. Même s'il s'agit alors plutôt d'une approximation de cycle, nous utiliserons tout de même le terme *cycle* pour y référer. Il est aussi important de préciser, bien que l'usage se fasse généralement en ce sens, qu'un décompte de cycles dans une fonction temporelle ne réfère pas à plusieurs cycles différents, mais bien à plusieurs occurrences du même cycle. Par exemple, si nous disons d'un signal qu'il comporte cinq cycles, cela signifie qu'un cycle  $y$  est répété cinq fois.

### Définition 2.1.3 : $p$ -Périodique

*Une fonction temporelle  $f$  est dite  $p$ -périodique si elle respecte le prédicat suivant :*

$$p \in ]0, \frac{\mathcal{I}\mathbb{T}_f}{2}] \wedge \forall (t, t+p) \in \mathbb{T}_f^2, f(t) = f(t+p) \quad (2.1)$$

où  $\mathcal{I}\mathbb{T}_f$  est l'étendue temporelle de l'intervalle  $\mathbb{T}_f$ .

### Définition 2.1.4 : Étendue temporelle

*L'étendue temporelle  $\mathcal{I}\mathbb{T}_f$  d'un intervalle  $\mathbb{T}_f = [a, b]$  est défini comme la différence entre*

a le plus petit élément du domaine de  $\mathbb{T}_f$  qui soit plus grand que  $b$ . Lorsque  $\mathbb{T}_f$  est défini sur  $\mathbb{R}$ , la définition suivante s'applique :

$$\lceil a, b \rceil = (b - a) + 0^+ \quad (2.2)$$

Alors que si  $\mathbb{T}_f$  est défini sur  $\mathbb{N}$  ou sur  $\mathbb{Z}$ , la définition suivante s'applique :

$$\lceil a, b \rceil = (b - a) + 1 \quad (2.3)$$

En d'autres mots, cela signifie qu'une fonction temporelle  $f$  est dite  $p$ -périodique si tout état de  $f$  est répété identiquement  $p$  unités de temps plus tard dans les limites de  $\mathbb{T}_f$ . La notation  $\mathbb{T}_f^2$  est utilisée comme forme abrégée pour désigner le domaine bidimensionnel  $\mathbb{T}_f \times \mathbb{T}_f$ . La limite supérieure sur  $p$  assure que l'intervalle  $\mathbb{T}_f$  contienne au moins deux cycles. En effet, pour considérer une fonction temporelle comme  $p$ -périodique, il faut pouvoir avoir l'assurance que chaque état de son cycle soit répété identiquement et entièrement, ce qui demande au moins deux cycles. De façon générale, s'il existe un  $p$  non nul tel que  $f$  est  $p$ -périodique, nous dirons de  $f$  qu'elle est périodique.

Une fonction temporelle  $p$ -périodique est également  $2p$ -périodique dans la mesure où  $p$  est suffisamment petit par rapport à l'étendue temporelle de la fonction  $f$ . La démonstration suivante explicite la situation :

$$\begin{aligned} & p \in ]0, \frac{\lceil \mathbb{T}_f \rceil}{4}] \wedge \forall (t, t+p) \in \mathbb{T}_f^2, f(t) = f(t+p) \\ \Rightarrow & p \in ]0, \frac{\lceil \mathbb{T}_f \rceil}{4}] \wedge \forall (t+p, t+2p) \in \mathbb{T}_f^2, f(t+p) = f(t+2p) \\ \Rightarrow & p \in ]0, \frac{\lceil \mathbb{T}_f \rceil}{4}] \wedge \forall (t, t+2p) \in \mathbb{T}_f^2, f(t) = f(t+2p) \\ \Rightarrow & \langle \text{en substituant } \rho = 2p \rangle \\ & \rho \in ]0, \frac{\lceil \mathbb{T}_f \rceil}{2}] \wedge \forall (t, t+\rho) \in \mathbb{T}_f^2, f(t) = f(t+\rho) \end{aligned}$$

Nous constatons que si  $p$  remplit la condition,  $2p$  la remplit également dans la mesure où  $p$  est compris dans l'intervalle  $]0, \lceil \mathbb{T}_f \rceil / 4]$ . En fait, cette démonstration pourrait facilement être généralisée pour tout  $np > 0$ , avec  $n \geq 1$ , dans la mesure où  $p$  est compris dans l'intervalle  $]0, \lceil \mathbb{T}_f \rceil / (2n)]$ . Ceci nous motive donc à bien spécifier ce que nous entendons par le terme période.

### Définition 2.1.5 : Période

Si une fonction temporelle  $f$  est périodique, la plus petite valeur de  $p$  pour laquelle  $f$  est  $p$ -périodique est dite la période  $\pi$  de  $f$  :

$$\pi = \min_{p \in P} (p) \quad (2.4)$$

où  $P$  est l'ensemble des valeurs de  $p$  pour lesquelles  $f$  est  $p$ -périodique.

Il est important de noter que toutes les valeurs  $p \in P$  sont des multiples entiers de cette période  $\pi$ . Aussi, il est fréquent en traitement de signaux d'admettre toutes ces valeurs de  $p$  comme période<sup>1</sup>. Notre décision de ne considérer que la plus petite d'entre elles se base sur le fait que nous désirons avoir des cycles simples. En effet, pour une fonction  $f$  de période  $\pi$ , si un intervalle  $\pi$  sur  $f$  admet exactement un maximum et exactement un minimum,  $n\pi$  en admettra exactement  $n$ , ce qui ne respecte pas notre définition de cycle simple. Il n'y a donc que la période  $\pi$  qui présentera un cycle simple si la fonction en comporte.

À la suite de ces définitions du cas périodique, attardons-nous maintenant au cas cyclique. De façon non formelle, une fonction temporelle est dite cyclique si elle n'est pas tout à fait périodique, mais s'en approche suffisamment. Considérons une fonction temporelle périodique. Nous pouvons la transformer de deux façons pour en faire une fonction temporelle cyclique non périodique. La première méthode de transformation consiste à moduler la période instantanée à chaque temps  $t$  en lui ajoutant ou soustrayant une valeur arbitraire.

### Définition 2.1.6 : Cyclique par Modulation Temporelle

Une fonction temporelle  $f$  est dite cyclique par modulation temporelle si elle respecte le prédicat suivant pour une durée moyenne de cycle  $c \in ]0, \frac{T_f}{2}]$  :

$$\exists \xi_c : \mathbb{T}_f \mapsto \mathbb{R}, \forall t \in \mathbb{T}_{\xi_c}, f(t) = f(t + c + \xi_c(t)) \quad (2.5)$$

sous les conditions suivantes :

$$c + \xi_c(t) > 0 \quad (2.6)$$

$$t_1 < t_2 \Leftrightarrow (t_1 + \xi_c(t_1)) < (t_2 + \xi_c(t_2)) \quad (2.7)$$

où  $\xi_c$  est une fonction d'écarts définie sur l'intervalle  $\mathbb{T}_f$  et  $\mathbb{T}_{\xi_c}$  est le sous-intervalle des temps  $t \in \mathbb{T}_f$  pour lesquels  $t + c + \xi_c(t)$  est également inclus dans  $\mathbb{T}_f$ .

À quelques détails près, cette définition est similaire à celle présentée par Seitz et Dyer dans [12]. Les différences les plus importantes sont :

- la définition proposée ici s'applique à un domaine infini comme à un domaine fini alors que la définition donnée dans [12] ne s'applique que pour un domaine infini ;
- la définition proposée ici sépare explicitement la durée moyenne d'un cycle  $c$  de la fonction d'écarts  $\xi_c$ , ce qui est important dans notre analyse puisqu'il s'agit de deux informations bien distinctes.

---

<sup>1</sup>La plus petite période est alors désignée *période fondamentale*

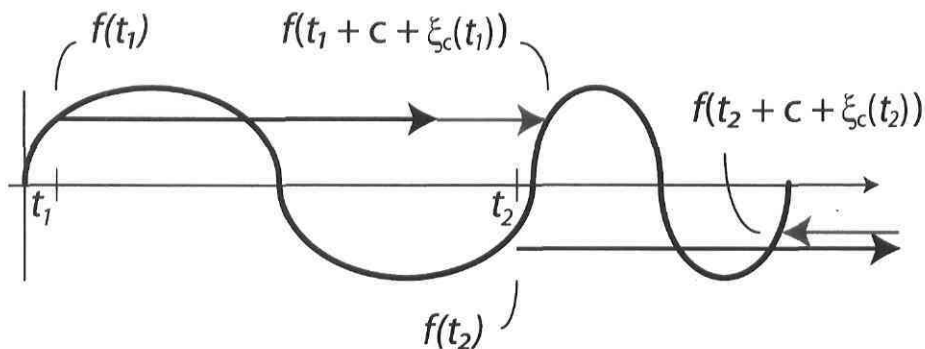


FIG. 2.1 – Exemple d’un signal cyclique par modulation temporelle. Le tracé noir dénote le signal. Les flèches noires et rouges dénotent respectivement la durée moyenne  $c$  d’un cycle et l’ajustement  $\xi_c(t)$  à cette durée pour le temps  $t$  spécifié.

En observant notre définition plus attentivement, nous remarquons que les bornes posées sur  $c$  sont similaires aux bornes posées sur  $p$  dans l’équation (2.1) : elles assurent que plus d’un cycle est présent dans la fonction temporelle<sup>2</sup>. La condition (2.6) posée sur la fonction d’écarts assure que la durée de cycle demeure toujours positive, même après l’ajustement. La condition (2.7), quant à elle, vient soutenir que l’ordre des états d’un cycle reste constant tout au long de la fonction temporelle. Notons que si tous les écarts  $\xi_c(t)$  sont nuls, les conditions (2.6) et (2.7) deviennent triviales, l’équation (2.5) est équivalente à l’équation (2.1) et nous retrouvons le cas périodique. Le cas cyclique par modulation temporelle est donc une généralisation du cas périodique. Ceci nous fait remarquer que les écarts  $\xi_c(t)$  nous indiquent à quel point nous sommes prêts ou loin du cas périodique. La figure 2.1 illustre un signal cyclique par modulation temporelle ainsi que des exemples d’écarts  $\xi_c(t)$ .

La seconde méthode pour transformer une fonction temporelle périodique en fonction temporelle cyclique non périodique consiste à modifier les états de façon à ce que deux états situés à une période d’intervalle ne soient pas forcément identiques, mais seulement ressemblants.

### Définition 2.1.7 : Cyclique par Modulation d’États

Soient  $f$ , une fonction temporelle, et  $d(f(t_1), f(t_2))$ , une mesure de distance entre les états  $f(t_1)$  et  $f(t_2)$  prenant une valeur nulle pour deux états identiques et une valeur positive pour deux états différents.  $f$  est dite cyclique par modulation d’états si elle

<sup>2</sup>Dans le cas où  $c$  n’est pas connu *a priori*, il peut être estimé en divisant la durée  $T_f$  de la fonction temporelle par le nombre de cycles qu’elle contient.

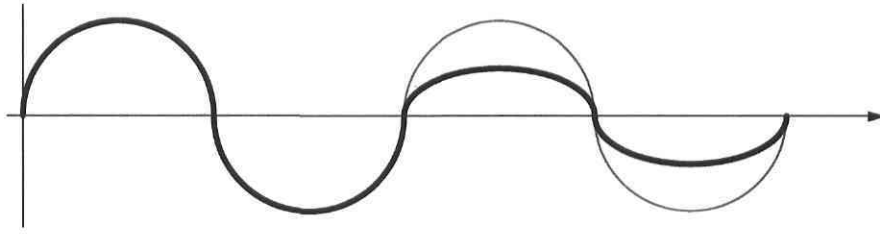


FIG. 2.2 – Exemple d’un signal cyclique par modulation d’états. Le tracé noir dénote le signal et le tracé rouge dénote le signal décalé de  $c$ . Les zones jaunes représentent l’intégrale des distances entre deux cycles consécutifs.

respecte l’inégalité suivante pour une durée moyenne de cycle  $c \in ]0, \frac{\mathbb{T}_f}{2}]$  :

$$\int_{(t,t+c) \in \mathbb{T}_f^2} \frac{d(f(t), f(t+c))}{\mathbb{T}_f - c} dt \leq \delta \quad (2.8)$$

où  $\delta$  est une borne supérieure sur la distance moyenne tolérée pour considérer une fonction temporelle comme cyclique par modulation d’états.

Spécifions que le dénominateur  $\mathbb{T}_f - c$  est en fait l’étendue temporelle du domaine d’intégration, utilisé ici afin d’obtenir la moyenne du dénominateur. Selon cette définition, toute fonction temporelle peut être cyclique par modulation d’états pour une valeur suffisamment grande de  $\delta^3$ . Toutefois, plus  $\delta$  est petit, plus nous nous rapprochons du cas périodique. Pour une valeur de  $c$  donnée, le plus petit  $\delta$  toléré nous indique donc à quel point nous sommes distant du cas périodique. Notons que pour  $\delta = 0$ , l’équation (2.8) est équivalente à l’équation (2.1). Le cas cyclique par modulation d’états est donc également une généralisation du cas périodique. La figure 2.2 illustre un signal cyclique par modulation d’états ainsi que le calcul de la distance entre deux cycles.

Dans une situation réelle telle qu’en analyse de mouvements humains, il faut s’attendre à rencontrer les deux types de modulations simultanément. En fait, à l’exception d’exemples théoriques ou synthétiques, rares sont les exemples de modulation temporelle stricte ou de modulation d’états stricte. Il faut donc pouvoir généraliser notre définition de *cyclique* pour mettre en évidence les deux types de modulation.

### Définition 2.1.8 : Cyclique (définition généralisée)

Soient  $f$ , une fonction temporelle, et  $d(f(t_1), f(t_2))$ , une mesure de distance entre les

<sup>3</sup>Notons que ceci est propre à la modulation d’états. Toute fonction temporelle n’est pas forcément cyclique par modulation temporelle.

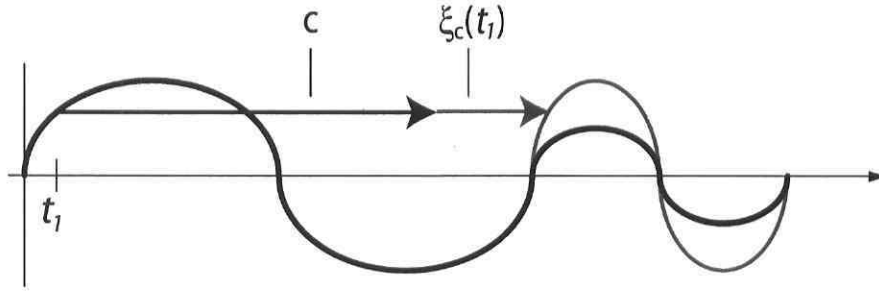


FIG. 2.3 – Exemple d'un signal cyclique par modulation temporelle et modulation d'états. Le tracé noir dénote le signal et le tracé rouge dénote le signal décalé de  $c$  et modulé par les écarts  $\xi_c(t)$ . Les zones jaunes représentent l'intégrale des distances entre deux cycles consécutifs temporellement modulés.

états  $f(t_1)$  et  $f(t_2)$  prenant une valeur nulle pour deux états identiques et une valeur positive pour deux états différents.  $f$  est dite cyclique si elle respecte le prédicat suivant pour une durée moyenne de cycle  $c \in ]0, \frac{\mathbb{T}_f}{2}]$  :

$$\exists \xi_c : \mathbb{T}_f \mapsto \mathbb{R}, \int_{t \in \mathbb{T}_{\xi_c}} \frac{d(f(t), f(t + c + \xi_c(t)))}{\mathbb{T}_{\xi_c}} dt \leq \delta \quad (2.9)$$

sous les conditions suivantes :

$$c + \xi_c(t) > 0 \quad (2.10)$$

$$t_1 < t_2 \Leftrightarrow (t_1 + \xi_c(t_1)) < (t_2 + \xi_c(t_2)) \quad (2.11)$$

Dans cette définition généralisée, nous remarquons les termes d'écarts  $\xi_c(t)$  qui font état de la modulation temporelle ainsi que la borne  $\delta$  qui fait état de la modulation d'états. En portant à 0 les  $\xi_c(t)$  ou la borne  $\delta$ , nous retrouvons respectivement les équations (2.8) et (2.5). Cette nouvelle définition généralise donc les définitions de *cyclique par modulation temporelle* et *cyclique par modulation d'états*. Par conséquent, elle généralise également la définition de *périodique*. La figure 2.3 illustre un signal cyclique simultanément par les deux types de modulation.

Bien que non formel, nous utiliserons aussi à l'occasion le terme *non-cyclique* pour signifier qu'une fonction temporelle est trop éloignée d'une forme périodique, soit qu'elle répond à l'équation (2.9) avec un  $\delta$  trop grand, avec une trop grande variance pour les termes d'écarts  $\xi_c(t)$  ou les deux.



## 2.2 Analyse de périodicité

Afin de pouvoir analyser convenablement la périodicité d'une séquence vidéo, il est crucial de se doter d'une mesure de périodicité.

Polana et Nelson [9] proposent un indice de périodicité qui peut être calculé sur un signal et généralisé à un ensemble de signaux. Cet indice se base sur la Transformée de Fourier du signal analysé. Nous calculons d'abord la sommation des amplitudes aux harmoniques de la fréquence fondamentale soustraite de la sommation des amplitudes trouvées à mi-chemin entre deux harmoniques. Cette différence est ensuite normalisée par la somme des deux sommations précédentes. L'indice ainsi obtenu est contenu dans l'intervalle  $[0, 1]$ , tendant vers 0 lorsque le signal n'est pas cyclique et prenant la valeur de 1 pour un signal parfaitement périodique. Pour un signal  $s$ , cet indice  $p_s$  s'exprime par l'équation suivante :

$$p_s = \frac{\sum_{h=1}^{\Omega} F_s(h\omega_s) - \sum_{h=1}^{\Omega} F_s(h\omega_s + \omega_s/2)}{\sum_{h=1}^{\Omega} F_s(h\omega_s) + \sum_{h=1}^{\Omega} F_s(h\omega_s + \omega_s/2)} \quad (2.12)$$

où  $F_s$  est la Transformée de Fourier du signal  $s$ ,  $\omega_s$  est sa fréquence fondamentale,  $h$  est l'harmonique et  $\Omega$  est le nombre d'harmoniques utilisées.

Pour généraliser cet indice à un ensemble de signaux  $S$ , nous regroupons les signaux selon leur fréquence fondamentale et nous calculons l'indice de périodicité moyen pour chaque tel groupe :

$$P_{S,\omega} = \sum_{s \in S_\omega} p_s / |S_\omega| \quad (2.13)$$

où  $S_\omega$  est l'ensemble des signaux  $s \in S$  ayant  $\omega$  pour fréquence fondamentale. L'indice de périodicité moyen le plus élevé est ensuite associés à l'ensemble de signaux  $S$  :

$$P_S = \max_{\omega} (P_{S,\omega}) \quad (2.14)$$

Cet indice a été appliqué à des séquences vidéo cycliques ainsi qu'à des séquences vidéo non cycliques en utilisant comme signaux les valeurs de gris prises par chaque pixel tout au long de la séquence. Les résultats présentés dans [9] illustrent une dichotomie marquée entre les deux types de séquences.

Seitz et Dyer [12] présentent une méthode d'évaluation des anomalies dans un mouvement périodique : la *period trace*. Cette courbe indique la période instantanée pour

chaque temps  $t$ , soit le nombre d'unités de temps à parcourir pour revenir à la même phase dans le cycle. Dans le cas parfaitement périodique, la période instantanée est constante et la trace est donc une droite parallèle à l'axe temporel. Moins le signal est cyclique, plus nous verrons des irrégularités apparaître sur la trace. Cette analyse peut être appliquée à un signal simple, à une séquence vidéo ou à toute autre fonction temporelle pour laquelle nous pouvons définir une mesure de distance entre deux états. Cette mesure de distance sert à évaluer la période instantanée en appariant les états les moins distants. Nous calculons aussi, à l'aide du test Kolmogorov-Smirnov (K-S) [11], une borne supérieure sur la probabilité que le signal, la séquence vidéo ou autre fonction ne soit pas périodique.

Lin et coll. [7] proposent une analyse de la périodicité de textures bidimensionnelles par le biais de l'autocorrélation. Cette publication fait état de la relation mathématique entre la Transformée de Fourier d'une texture et son autocorrélation. Elle souligne également que la périodicité d'une texture est plus facile à constater à l'aide de son autocorrélation plutôt que de sa Transformée de Fourier.

Cutler et Davis [3] proposent une analyse de la périodicité d'une séquence vidéo en développant l'approche proposée par Lin et coll. [7] et en utilisant l'autocorrélation normalisée. À partir d'une séquence vidéo, ils calculent d'abord une matrice de similarité en comparant deux à deux toutes les images de la séquence. Si elle est issue d'une séquence vidéo périodique, la matrice de similarité prendra la forme d'une texture périodique. Cette forme est explicitée en procédant à l'autocorrélation normalisée de cette matrice de similarité. Pour une matrice de similarité  $S$  avec un décalage  $(d_x, d_y)$ , la fonction d'autocorrélation utilisée s'exprime comme suit :

$$A_S(d_x, d_y) = \frac{\sum_{(x,y) \in R} (S(x, y) - \overline{S_R})(S(x + d_x, y + d_y) - \overline{S_{R_L}})}{\sqrt{\sum_{(x,y) \in R} (S(x, y) - \overline{S_R})^2 \sum_{(x,y) \in R} (S(x + d_x, y + d_y) - \overline{S_{R_L}})^2}} \quad (2.15)$$

où  $\overline{S_R}$  est la valeur moyenne de  $S$  dans la région  $R$  et  $\overline{S_{R_L}}$  est la valeur moyenne de  $S$  dans la région  $R$  décalée de  $(d_x, d_y)$ . Le numérateur est en fait l'autocorrélation de l'image soustraite de sa moyenne alors que le dénominateur vient normaliser cette valeur dans l'intervalle  $[0, 1]$ .

Pour une texture périodique de période  $\pi$ , la matrice formée par tous les points calculables de  $A_S$  comportera des maxima locaux pour tous les décalages  $(d_x, d_y) \in (\pi\mathbb{Z})^2$ , où  $\pi\mathbb{Z}$  est l'ensemble formé par les multiples entiers positifs et négatifs de  $\pi$ . Les auteurs utilisent cette idée en ajustant un treillis de points équidistants sur ces maxima, entre autres afin de déterminer si la séquence est cyclique ou non et de trouver

la période de cette séquence.

Tel que nous le verrons au chapitre 3, nous devons pouvoir comparer deux signaux sur leur niveau de périodicité. Pour ce faire, nous devons utiliser une mesure quantitative de périodicité. L'équation (2.12) remplit directement cet objectif, mais comme nous le verrons, son approche basée sur le domaine fréquentiel n'est pas idéale pour nos besoins. Dans l'approche proposée par Seitz et Dyer[12], l'information quant au niveau de périodicité se retrouve à la fois dans la forme de la trace et dans la probabilité calculée à l'aide du test K-S. Il n'est donc pas trivial d'en déduire un taux de périodicité. L'approche proposée par Cutler et Davis[3], quant à elle, peut être facilement adaptée à nos besoins. Nous y reviendrons au prochain chapitre.

## 2.3 Segmentation temporelle

Puisque notre objectif premier est de segmenter une séquence vidéo en segments cycliques, il est naturel de s'attarder à ce que d'autres chercheurs ont pu proposer sur le sujet de la segmentation temporelle.

Bobick et Davis [2] ont développé une méthode de segmentation temporelle adaptée à leur problématique. Cette méthode dépend d'une base de données contenant des représentations compactes, soient les *motion-energy image* (MEI) et *motion-history image* (MHI), de chaque mouvement pouvant être détecté. Lors de l'analyse d'une séquence vidéo, des MEI et MHI sont formées à chaque temps  $t$  à l'aide des  $\tau$  dernières images de la séquence pour différentes durées  $\tau$ . Les MEI et MHI ainsi formées sont comparées aux activités contenues dans la base de données. Une similarité suffisamment élevée indique que le mouvement concerné a été détecté du temps  $t - \tau$  au temps  $t$ .

Bien que l'approche précédente procède à une forme de segmentation temporelle d'activités, elle s'applique difficilement à nos objectifs. En effet, ce type d'approches se base sur une liste exhaustive de modèles de mouvements possibles afin de procéder à la segmentation de séquences vidéo. Comme nous espérons pouvoir traiter des séquences vidéo contenant toutes sortes d'activités de durées indéfinies, il est impensable de générer une base de données contenant toutes les activités pouvant être segmentées. Toutefois, ces approches ont l'avantage de pouvoir segmenter des activités non cycliques, telles que s'asseoir ou saisir un objet, ce que nous ne pouvons espérer faire à l'aide d'une méthode n'utilisant pas de modèles. En effet, si les activités cycliques ont une caractéristique commune, soit que leurs états se répètent à intervalles réguliers, les activités non cycliques n'ont pas de telle caractéristique facile à détecter *a priori*. Plus

précisément, sans modèle ou autre connaissance, il est difficile, voire impossible, de savoir si une activité non cyclique n'est pas en fait deux activités non cycliques effectuées l'une après l'autre. Par cette capacité, ces approches viennent combler les lacunes dont une approche de segmentation générique ne pourra jamais se défaire.

Min et Kasturi [8] viennent découpler davantage le processus de segmentation d'activités non cycliques d'une base de données d'activités. Leur approche décrit les mouvements en termes d'un nombre variable de trajectoires de mains, coudes, genoux et pieds. La variabilité du nombre de trajectoires vient du fait qu'un membre ne produit pas de trajectoires s'il n'est pas en mouvement. Les activités sont traitées séparément au niveau des bras et des jambes et sont décrites par des Modèles de Markov Cachés (MMC). Bien que la segmentation finale d'une séquence analysée se fasse par un processus de comparaison avec une base de données d'activités connues, des étapes de segmentation préalables sont effectuées sans aucune forme de modèle. Premièrement, les trajectoires appartenant aux mouvements des bras sont séparées de ceux appartenant aux mouvements des jambes. Cette segmentation anatomique se fait en trouvant la position moyenne des trajectoires. Elle s'appuie sur l'hypothèse que la personne se tient debout, ce qui est généralement acceptable en analyse de mouvements selon l'application envisagée. Deuxièmement, une segmentation temporelle approximative se fait en groupant les trames consécutives contenant le même nombre de trajectoires. Bien que nous puissions y trouver des contre-exemples, l'hypothèse d'un nombre de trajectoires constant à l'intérieur d'une activité couvre un grand nombre d'activités quotidiennes.

Bien que la dernière approche présentée se dote de méthodes de segmentation automatique sans modèles, la segmentation n'y est pas complètement découplée de la reconnaissance. Nécessairement, cela implique qu'elle n'est pas complètement découplée d'une base de connaissances. En fait, rares sont les recherches qui traitent uniquement de la segmentation temporelle d'activités sans également traiter de la reconnaissance d'activités. Parmi ces rares recherches, nous comptons celle de Yazdi, Branzan Albu et Bergevin [15] qui propose une méthode de bas niveau pour segmenter temporellement les activités cycliques symétriques. Cette méthode est basée sur une analyse topologique d'une matrice de similarité [3] générée à partir d'une séquence d'images binaires représentant la silhouette d'une personne. Les groupes de carrées similaires trouvés le long de la diagonale principale y sont entourés d'une boîte englobante carrée dont les coordonnées correspondent aux indices temporels du début et de la fin d'une activité cyclique symétrique.

Les résultats obtenus par Yazdi et coll. [15] sont prometteurs. Toutefois, deux limitations importantes y sont mentionnées. Premièrement, cette approche ne considère que les activités cycliques symétriques, soient les activités où la deuxième moitié du cycle

correspond à l'inversion de la première moitié du cycle. Cette limitation n'est pas trop gênante lorsque nous désirons analyser des activités telles que la marche où le mouvement relatif des pieds et des mains est celui d'un pendule, mais, à titre d'exemple, elle risque de poser problème dans un contexte de course où ce mouvement est plutôt circulaire. Deuxièmement, elle ne considère que les cycles entiers. Dans le cas d'une activité cyclique dont le dernier cycle n'est pas tout à fait complet, cela risque de nuire fortement à l'exactitude de la segmentation.

Comme nous le verrons, l'utilisation d'une approche basée sur un modèle de squelette nous permettra de combler les deux lacunes de cette dernière méthode et nous permettra également de procéder à une segmentation anatomique des mouvements, un peu à l'image de celle faite par Min et coll. dans [8].

# Chapitre 3

## Segmentation d'un signal

Ce chapitre présente, analyse et compare plusieurs variantes d'une méthode visant à extraire de façon précise les segments cycliques d'un signal unidimensionnel. Nous y présentons aussi quelques hypothèses simplificatrices sur le problème afin d'obtenir des résultats plus intéressants dans une grande majorité de cas.

La segmentation précise des parties cycliques d'un signal est essentielle à la réussite de notre approche, s'il est impossible d'obtenir une segmentation adéquate d'un signal, nous ne pouvons espérer pouvoir en faire autant pour une séquence vidéo. Étant donné la généralité de la définition du terme cyclique, la segmentation automatique est beaucoup plus ardue que nous ne pourrions le croire *a priori*. Il s'agit donc à la fois du problème le plus important et du problème le plus complexe de notre approche. La majorité de nos recherches ont donc visé à résoudre ce problème.

Premièrement, analysons rapidement la complexité du problème. Pour un signal  $S$  de longueur  $l_S$ , il existe  $2^{l_S-1}$  segmentations différentes possibles. En effet, chaque trame est soit une trame de début ou fin d'une activité cyclique, désignée trame de transition, soit une trame autre<sup>1</sup>. Il doit toutefois y avoir un nombre pair de trames de transition afin de s'assurer que chaque activité cyclique a un début et une fin. Le dénombrement des solutions possibles est donc un problème isomorphe au dénombrement des nombres binaires de longueur  $l_S$  ayant un nombre pair de 1. Formellement, nous pouvons procéder au dénombrement à l'aide des séries génératrices exponentielles. Élaborons d'abord la série  $h_0$ , où le coefficient  $a_n$  de chaque terme  $a_n \frac{x^n}{n!}$  représente le nombre de séquences

---

<sup>1</sup>Les trames autres sont soit des trames de continuité de pauses ou d'activités cycliques ou non cycliques, soit des trames de transition entre pauses et activités non cycliques

possibles de longueur  $n$  ne contenant que des 0 :

$$\begin{aligned} h_0(x) &= 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \\ h_0(x) &= e^x \end{aligned}$$

Nous élaborons ensuite la série  $h_1$ , où le coefficient  $a_n$  de chaque terme  $a_n \frac{x^n}{n!}$  représente le nombre de séquences possibles de longueur  $n$  ne contenant que des 1 et respectant la contrainte de parité :

$$\begin{aligned} h_1(x) &= \frac{1}{2} \left( 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \right) + \frac{1}{2} \left( 1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots \right) \\ h_1(x) &= \frac{e^x + e^{-x}}{2} \end{aligned}$$

Nous pouvons ensuite combiner ces séries en une autre série  $g$ , où le coefficient  $a_n$  de chaque terme  $a_n \frac{x^n}{n!}$  représente le décompte des nombres binaires de longueur  $n$  ayant un nombre pair de 1 :

$$\begin{aligned} g(x) &= h_0(x) \cdot h_1(x) \\ g(x) &= e^x \left( \frac{e^x + e^{-x}}{2} \right) \\ g(x) &= \frac{1}{2} (e^{2x} + e^{0x}) \\ g(x) &= \sum_{l_S=0}^{\infty} \frac{1}{2} (2^{l_S} + 0^{l_S}) \frac{x^{l_S}}{l_S!} \end{aligned}$$

Nous en déduisons donc que nous pouvons former  $(1+1)/2 = 1$  tel nombre de longueur nulle, ce qui est de peu d'importance pour nous, et  $2^{l_S}/2 = 2^{l_S-1}$  tels nombres de longueur  $l_S > 0$ . Ce résultat n'est pas surprenant étant donné notre contrainte de parité. En effet, il suffit de voir que les  $l_S - 1$  premiers chiffres peuvent prendre n'importe quelle valeur et que le dernier chiffre servira à conserver la parité, en prenant la valeur de 0 s'il y a déjà parité pour les 1, ou à obtenir la parité, en prenant la valeur de 1 dans l'autre cas. Il y a donc  $l_S - 1$  choix binaires à prendre, ce qui nous donne bien  $2^{l_S-1}$  possibilités. Ce résultat se transfère directement au nombre de segmentations temporelles différentes possibles sur un signal de durée  $l_S$ .

Il est possible de réduire ce nombre en posant une contrainte sur la durée minimum d'une activité cyclique. L'expression mathématique du nombre de solutions est alors

considérablement plus complexe et l'explosion du nombre de solutions demeure toujours exponentielle. En effet, une version simplifiée de ce problème consiste à regrouper toutes les valeurs du signal par groupes de  $k$  valeurs consécutives. Ainsi, nous posons une contrainte sur la durée des activités cycliques, qui doit être un multiple de  $k$  supérieur à  $k$ . Nous posons aussi une contrainte sur la position des activités cycliques, qui ne peuvent débuter que sur les trames dont l'indice est un multiple de  $k$ . Avec ce problème, beaucoup plus contraint que ce que nous proposons, nous voyons facilement qu'il y a  $2^{\lfloor l_S/k \rfloor - 1}$  solutions possibles, ce qui est toujours exponentiel en  $l_S$ .

Puisque nous faisons face à un problème de complexité exponentielle, nous ne pouvons envisager de résoudre ce problème par force brute. L'approche que nous allons privilégier est donc un algorithme de type *glouton*. En considérant le signal entier, l'algorithme de segmentation cherchera le segment le plus prometteur et le conservera. Il procédera ensuite de la même façon sur les parties restantes du signal jusqu'à ce qu'il ne soit plus possible de trouver un segment suffisamment long ou suffisamment intéressant. Cette procédure est décrite en pseudo-code par l'algorithme 3.1.

---

**Algorithme 3.1** *SegmenterSignalGlouton*( $S$ )

---

```

1: SEG ← ∅
2: X ← {(1, lS)}
3: tant que X ≠ ∅ faire
4:   (I, J) ← élément arbitraire de X
5:   si J - I + 1 ≥ β alors
6:     (i, j) ← MeilleurSegment(S[I,J])
7:     si IndiceP(S[i,j]) > ηE alors
8:       SEG ← SEG ∪ {(i, j)}
9:     fin si
10:    si i - 1 > I alors
11:      X ← X ∪ {(I, i - 1)}
12:    fin si
13:    si j + 1 < J alors
14:      X ← X ∪ {(j + 1, J)}
15:    fin si
16:  fin si
17:  X ← X / {(I, J)}
18: fin tant que
19: retourner SEG

```

---

Prenons le temps d'expliquer les éléments de l'algorithme 3.1. Tout d'abord, signalons que la forme  $S_{[i,j]}$  est utilisée pour désigner le sous-signal compris entre le  $i^{\text{ème}}$  et le  $j^{\text{ème}}$  élément du signal  $S$  inclusivement, alors que  $l_S$  désigne la longueur du signal



$S$ .  $\text{SEG}$  est la segmentation retournée, soit un ensemble de paires d'indices de début et de fin de segments de  $S$ , alors que  $\mathbb{X}$  est un ensemble de segments temporaires. La constante  $\beta$  est un seuil sur la durée minimum d'une activité cyclique et la constante  $\eta_E$  est un seuil sur l'indice de périodicité<sup>2</sup> minimum d'une activité cyclique. Spécifions que l'indice  $E$  de  $\eta_E$  indique qu'il s'agit d'un seuil *élevé* sur la périodicité, par rapport au seuil *bas*  $\eta_B$  qui sera présenté plus loin.

L'algorithme 3.1 fait également appel à deux autres algorithmes :  $\text{MeilleurSegment}(S)$  et  $\text{IndiceP}(S)$ . L'algorithme  $\text{MeilleurSegment}(S)$  retourne le segment de  $S$  qui semble le plus cyclique parmi tous les segments possibles. Cet algorithme sera détaillé à la section 3.3. L'algorithme  $\text{IndiceP}(S)$ , quant à lui, retourne une estimation de la périodicité de signal  $S$ . La valeur retournée est dans l'intervalle  $[0, 1]$ , une valeur de 1 indiquant un signal parfaitement périodique. Cet algorithme sera détaillé à la section 3.2.

L'utilisation d'un algorithme glouton est tout indiquée pour notre problème. Une faille souvent rencontrée avec de tels algorithmes est qu'ils ne procèdent qu'à une évaluation locale du problème et qu'ils risquent donc de prendre une direction qui rendra impossible l'atteinte de la solution optimale. Toutefois, comme les premiers segments retournés par l'algorithme  $\text{MeilleurSegment}(S)$  sont ceux pour lesquels nous avons la plus grande certitude d'avoir bien segmenté précisément une partie cyclique du signal, ce choix ne nuira généralement pas à la segmentation des parties cycliques restantes.

### 3.1 Autocorrélation

Comme nous le verrons plus tard, les caractéristiques de l'autocorrélation en font un outil très intéressant pour analyser la périodicité d'un signal. Prenons donc le temps de nous familiariser davantage avec cette opération.

L'autocorrélation normalisée d'un signal discret  $S$  de longueur finie soustrait de sa moyenne s'exprime par l'équation suivante :

$$A_S(d) = \frac{\sum_{x=1}^{l_S-d} (S(x) - \bar{S})(S(x+d) - \bar{S})}{\sqrt{\sum_{x=1}^{l_S-d} (S(x) - \bar{S})^2 \sum_{x=1}^{l_S-d} (S(x+d) - \bar{S})^2}} \quad (3.1)$$

---

<sup>2</sup>Le concept d'indice de périodicité sera détaillé dans la section 3.2.

où  $d$  est la valeur de décalage<sup>3</sup>,  $l_S$  est la longueur du signal  $S$  et  $\bar{S}$  est la valeur moyenne de  $S$ .

Cette équation s'apparente beaucoup à l'équation (2.15). Toutefois, contrairement à cette dernière qui s'applique à un signal bidimensionnel, l'équation présentée ici s'applique à un signal unidimensionnel. De plus, la moyenne utilisée est calculée sur la totalité du signal au lieu d'être calculée seulement sur une région telle que dans l'équation (2.15). Dans le cadre du présent chapitre et du chapitre suivant, le terme autocorrélation fera toujours référence à l'équation (3.1), puisque nous manipulerons des signaux unidimensionnels, discrets et finis.

Il s'avère que l'autocorrélation d'un signal est un outil robuste pour analyser la périodicité d'un signal. Intuitivement, puisque l'autocorrélation procède à la sommation de chaque valeur du signal multiplié d'une valeur distante d'un décalage  $d$  et puisque ces valeurs sont soustraites de leurs moyennes, cette sommation sera maximisée lorsque toutes les valeurs du signal seront identiques ou très similaires aux valeurs trouvées après le décalage. Ceci nous donne l'idée que la trace de l'autocorrélation en fonction des décalages  $d$  d'un signal périodique comportera des maxima chacun espacé d'une période.

Partant de cette idée, analysons l'équation (3.1) afin de valider notre hypothèse. Nous pouvons premièrement établir que, peu importe le signal  $S$ ,  $A_S(0)$  sera toujours égal à 1 puisqu'un signal est toujours identique à lui-même. Pour simplifier la notation, posons  $\tilde{S}(x) = S(x) - \bar{S}$ .

$$\begin{aligned}
 A_S(0) &= \frac{\sum_{x=1}^{l_S} \tilde{S}(x)\tilde{S}(x)}{\sqrt{\sum_{x=1}^{l_S} \tilde{S}(x)^2 \sum_{x=1}^{l_S} \tilde{S}(x)^2}} \\
 \Leftrightarrow \\
 A_S(0) &= \frac{\sum_{x=1}^{l_S} \tilde{S}(x)^2}{\sqrt{\left(\sum_{x=1}^{l_S} \tilde{S}(x)^2\right)^2}} \\
 \Leftrightarrow \\
 A_S(0) &= 1
 \end{aligned}$$

---

<sup>3</sup>Les valeurs de décalage négatives ne sont pas considérées puisqu'elles donnent toujours la même valeur d'autocorrélation que leur contrepartie positive.

De plus, pour un signal périodique  $S$  de période  $\pi_S^4$ , nous obtenons que  $A_S(d) = 1$  pour tous les décalages  $d$  étant un multiple entier de  $\pi_S$  :

$$\begin{aligned}
 A_S(i \cdot \pi_S) &= \frac{\sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x) \tilde{S}(x + (i \cdot \pi_S))}{\sqrt{\sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x)^2 \sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x + (i \cdot \pi_S))^2}} \\
 \Leftrightarrow &< \text{par la définition 2.1} > \\
 A_S(i \cdot \pi_S) &= \frac{\sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x) \tilde{S}(x)}{\sqrt{\sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x)^2 \sum_{x=1}^{l_S - (i \cdot \pi_S)} \tilde{S}(x)^2}} \\
 \Leftrightarrow & \\
 A_S(i \cdot \pi_S) &= 1
 \end{aligned}$$

dans la mesure où  $i$  est un entier non négatif et que  $i \cdot \pi_S < l_S$ .

Nous voyons donc que l'autocorrélation d'un signal périodique présente une valeur de 1 pour tous les décalages étant des multiples de sa période  $\pi_S$  et uniquement pour ces décalages. Ainsi, tous les autres décalages ont une valeur d'autocorrélation strictement inférieure à 1, ce qui nous indique que les décalages  $d$  multiples entiers de  $\pi_S$  formeront des maxima sur la trace de l'autocorrélation.

Pour assurer la validité des algorithmes présentés dans ce chapitre, il est également important que la trace de l'autocorrélation ne comporte pas d'autres maxima locaux que ceux aux décalages multiples entiers de la période. Comme nous le verrons, les signaux périodiques formés de cycles simples ne génèrent pas de maxima locaux indésirables dans la trace de l'autocorrélation. Un signal périodique formé de cycles complexes est toutefois propice à en générer.

Commençons par rappeler la définition de cycle simple.

### Définition 3.1.1 : Cycle simple

*Un cycle est dit cycle simple s'il ne comporte qu'un seul maximum, ou plateau, qu'un seul minimum, ou vallée, et ne comporte aucun palier.*

Les sinusoïdales sont de bons exemples de fonctions basées sur un cycle simple.

<sup>4</sup>Afin de simplifier les démonstrations,  $\pi_S$  sera toujours considéré comme une valeur entière.

Observons de plus près comment se forme la trace de l'autocorrélation d'une sinusoïdale. Sur l'intervalle d'une période, la valeur initiale de 1 diminuera continuellement jusqu'à l'atteinte d'un minimum à un décalage exactement égal à la demi-période, puis augmentera continuellement pour retrouver une valeur de 1 à un décalage correspondant exactement à une période. La figure 3.1 illustre cette évolution de la trace de l'autocorrélation.

En observant de plus près la figure 3.1, il est plus facile de visualiser le processus. Nous pouvons y observer une relation directe entre la distance séparant les points de  $S(x)$  des points de  $S(x + d)$  et la valeur de l'autocorrélation. En effet, plus cette distance augmente, moins les produits des points de  $S(x)$  et de  $S(x + d)$  est grand, plus l'autocorrélation diminue. Plus le décalage tend vers  $d = 40$ , plus ces points sont distants. Nous observons aussi qu'à un décalage  $d = 40$ , chaque point de  $S(x)$  est multiplié avec sa négation dans  $S(x + d)$ . La somme de ces produits compose donc naturellement le minimum de l'autocorrélation. Étant donné la symétrie de la sinusoïdale, il est trivial de voir qu'en augmentant le décalage de  $d = 40$  à  $d = 80$ , le processus inverse se produit pour retrouver une valeur d'autocorrélation de 1 au décalage  $d = 80$ . Notons que la transition entre les extremums de l'autocorrélation se fait de façon continue, sans aucun maximum local.

Nous sommes d'avis que cette transition continue entre les extremums de l'autocorrélation est attribuable au fait que la sinusoïdale est un signal périodique basé sur un cycle simple. En effet, en considérant des signaux périodiques ne partageant, selon toutes vraisemblances, aucune propriété de la sinusoïdale autre que d'être basé sur un cycle simple, nous observons des traces d'autocorrélation comparables. Pour approfondir cette idée, notons les propriétés principales d'une sinusoïdale en rapport avec l'autocorrélation :

**Cycle simple :** Le cycle ne comporte qu'un maximum, qu'un minimum et aucun palier.

**Continuellement croissante / décroissante :** Entre un maximum et un minimum de la sinusoïdale, les valeurs vont toujours en décroissant et, entre un minimum et un maximum, les valeurs sont toujours croissantes. Elle ne comporte donc aucun plateau ou vallée.

**Symétrique d'amplitude :** Le cycle comporte autant de valeurs positives que négatives.

**Symétrique de phase :** La distance entre un maximum et son minimum précédent de la sinusoïdale est égale à la distance entre un maximum et son minimum suivant.

Or, l'autocorrélation de signaux ne respectant pas une ou plusieurs des trois dernières propriétés mentionnées semblent toujours générer une autocorrélation dont la trace ne

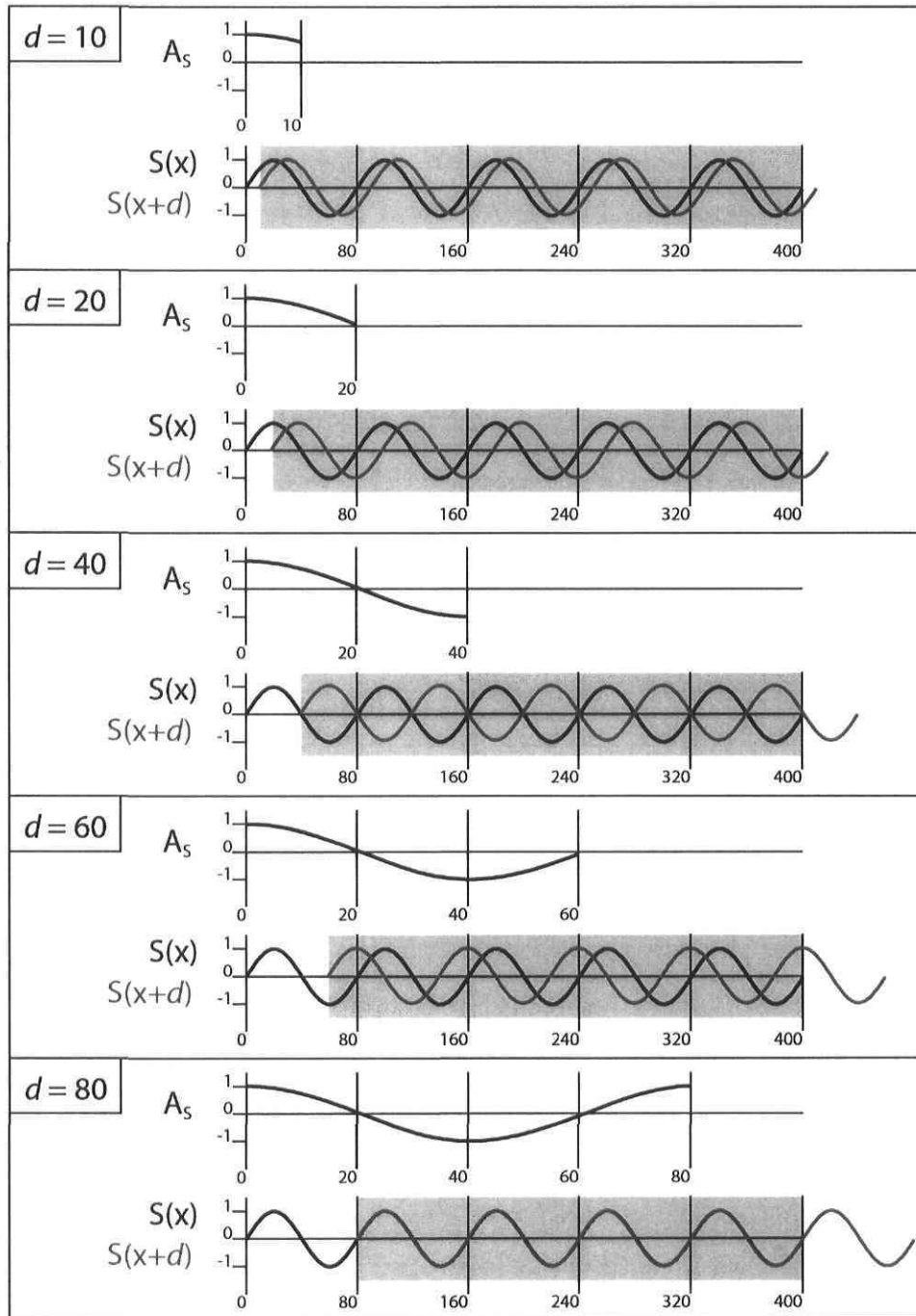


FIG. 3.1 – Évolution de la trace de l'autocorrélation  $A_S$  en fonction du décalage  $d$  pour un signal sinusoïdal. Pour chaque valeur de  $d$ , une trace partielle de l'autocorrélation et les signaux  $S(x)$  et  $S(x+d)$  sont illustrés. La zone ombragée grise correspond au domaine de sommation de l'autocorrélation.

comporte pas de maximums locaux indésirables. Le figure 3.2 illustre cette observation par le biais de cycles aussi variés que possible<sup>5</sup>. À l'aide de cette figure, nous pouvons également remarquer que, sur l'ensemble des signaux, l'autocorrélation semble devenir irrégulière pour des décalages  $d$  s'approchant de la longueur du signal  $l_S$ . Ceci s'explique par le fait que la sommation se fait sur un nombre restreint d'éléments du signal. Les parties sommées du signal ne constituent pas un cycle entier et une composante cyclique n'est pas forcément présente. Pour un signal périodique de période  $\pi_S$ , ceci peut faire apparaître un maximum local indésirable dans les dernières  $\pi_S/2$  valeurs de son autocorrélation. À défaut de pouvoir connaître la période  $\pi_S$  *a priori*, nous tenterons d'amoinrir les effets de cette situation, bien que peu fréquente, en ne considérant pas les valeurs du le dernier 10% de la trace de l'autocorrélation lors de nos analyses.

Ces observations nous incitent à poser l'hypothèse réaliste que l'autocorrélation d'un signal périodique  $S$  de période  $\pi_S$  ne comporte des maxima locaux qu'à des décalages multiples entiers de  $\pi_S$ , dans la mesure où  $S$  est basé sur un cycle simple. Bien que nos observations ne constituent pas une preuve formelle, elles montrent tout de même que des tests sur un ensemble très hétérogène de cycles simples appuient notre hypothèse. Nous croyons donc qu'il est important de souligner que, bien qu'une approche se basant sur cette hypothèse, telles que la notre et celle proposée dans [3], ne soit pas formellement validée par ces observations, elle en est au moins empiriquement validée de cette façon.

Il est également possible pour un cycle comportant plusieurs maxima, ou comportant des paliers, de former une autocorrélation dont la trace ne comporte que des maxima à des décalages multiples entiers de la période. Il est toutefois difficile de caractériser dans quelle mesure un maximum local ou un palier dans un cycle donnent lieu à des maxima indésirables dans l'autocorrélation, puisque cela est intimement relié à la forme globale du cycle. Les figures 3.3 et 3.4 illustrent cette situation par quelques exemples. Pour cette raison, nos recherches ne se concentrent que sur les fonctions basées sur des cycles simples.

Il est intéressant de noter que la trace de l'autocorrélation de signaux périodiques s'approche elle-même d'un signal périodique. Toutefois, cette quasi-périodicité s'estompe vers la fin de la trace de l'autocorrélation, alors que  $d$  tend vers  $l_S$ .

---

<sup>5</sup>Il est important de noter que notre équation pour  $A_S$  soustrait le signal  $S$  de sa moyenne. Les cycles étudiés ont donc tous une moyenne de 0.

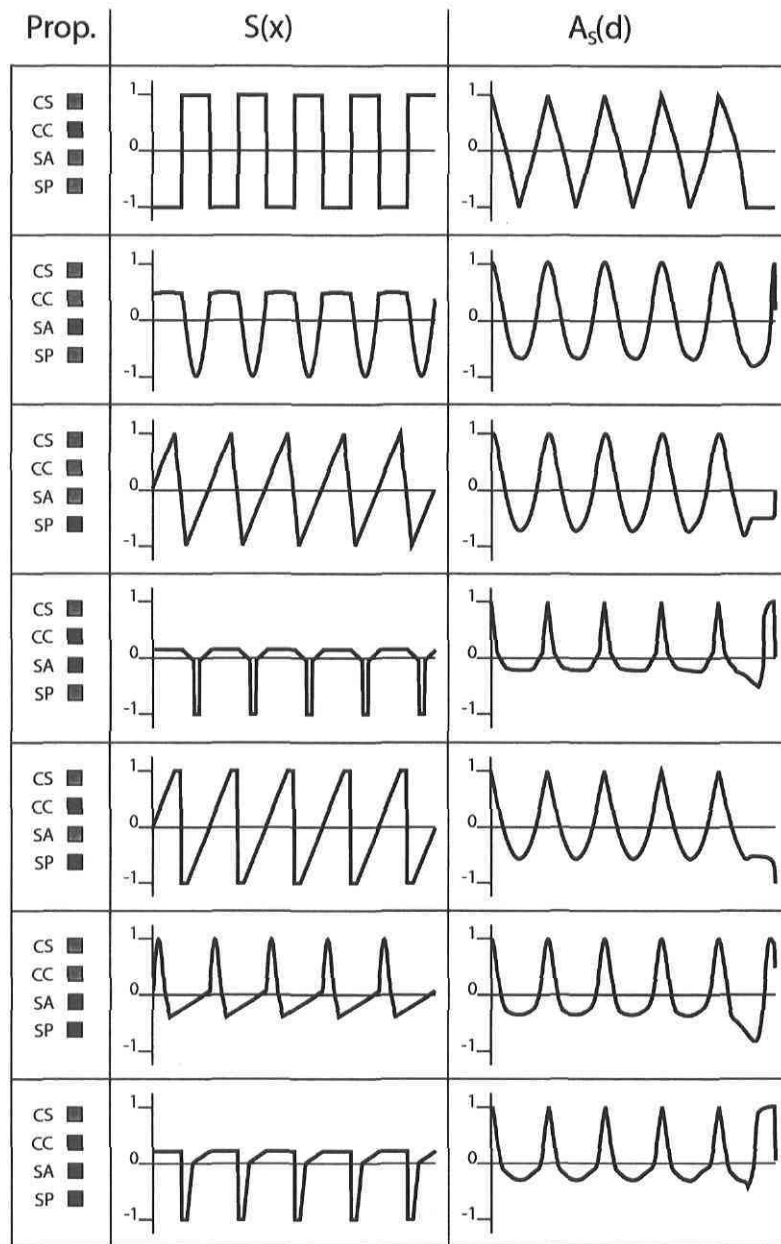


FIG. 3.2 – Signaux basés sur des cycles simples et la trace de leur autocorrélation. Chaque signal est précédé de ses propriétés : cycle simple (CS), continuellement croissant/décroissant (CC), symétrie d'amplitude (SA) et symétrie de phase (SP). Un carré vert indique que le signal possède cette propriété alors qu'un carré rouge indique l'inverse.

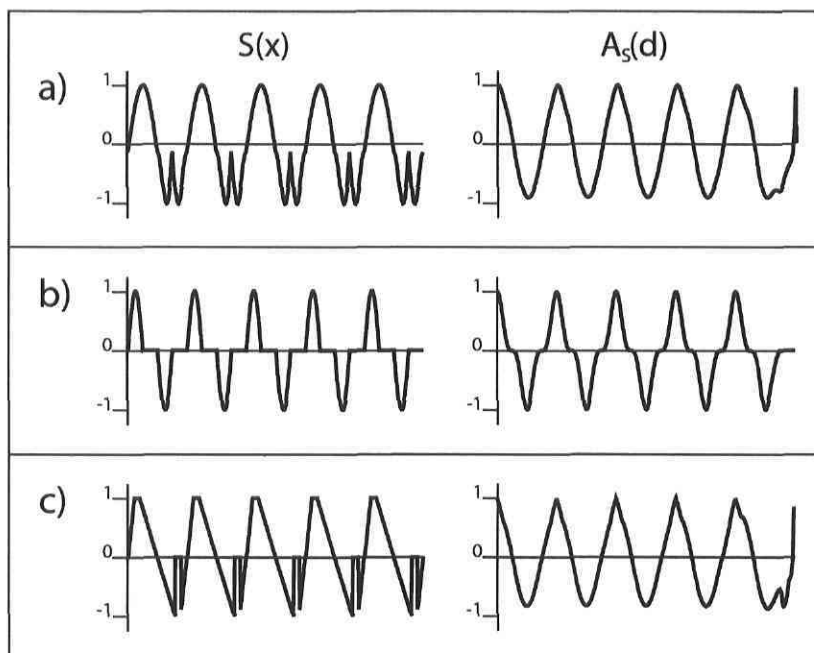


FIG. 3.3 – Exemples de signaux basés sur des cycles complexes dont l'autocorrélation ne comporte aucun maximum indésirable. a) Signal comportant plus d'un maximum par cycle. b) Signal comportant des paliers. c) Signal comportant plus d'un maximum par cycle et comportant des paliers.



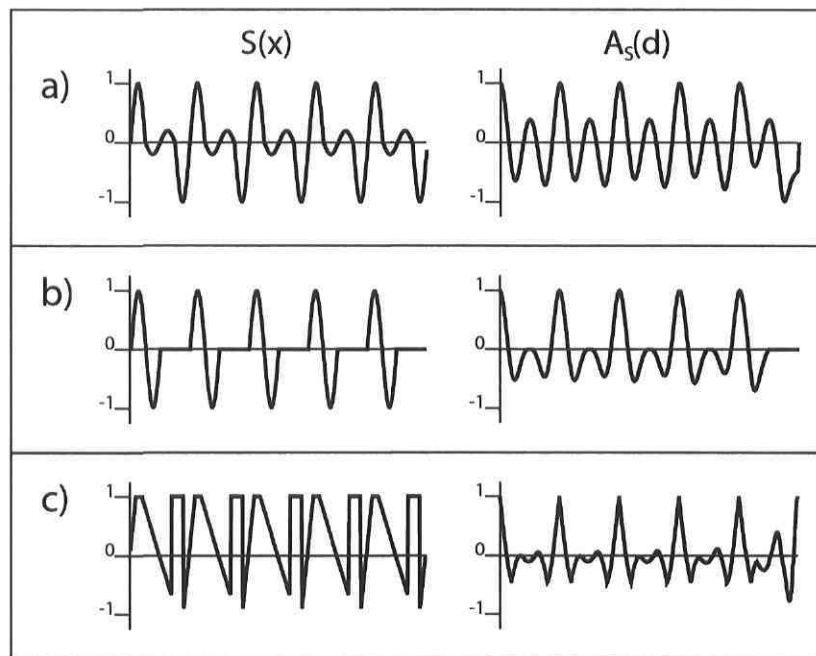


FIG. 3.4 – Exemples de signaux basés sur des cycles complexes dont l'autocorrélation comporte des maxima indésirables. a) Signal comportant plus d'un maximum par cycle. b) Signal comportant un palier. c) Signal comportant plus d'un maximum par cycle et comportant des paliers.

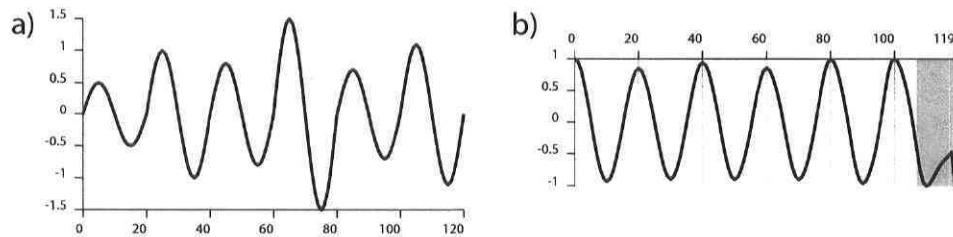


FIG. 3.5 – a) Un exemple de signal cyclique par modulation d'états. b) La trace de l'autocorrélation de ce signal. La zone ombragée recouvre le dernier dixième des valeurs d'autocorrélation. Ces valeurs peuvent être instables et sont donc ignorées.

## 3.2 Indice de périodicité

Appuyés par nos observations, nous avons donc posé l'hypothèse que l'autocorrélation d'un signal périodique de cycle simple présente des maxima d'une valeur de 1 pour tous les décalages multiples entiers de sa période et ne comporte aucun autre maximum. Il est alors normal de présumer qu'un signal cyclique présentera des traits similaires. Tel qu'illustré à la figure 3.5, dans le cas d'un signal cyclique par modulation d'états, les maxima sont toujours précisément aux décalages multiples entiers de la durée du cycle, mais ils n'ont pas forcément une valeur de 1. Cependant, d'une façon générale, plus la borne  $\delta$  tend vers 0, plus les maxima tendent vers 1. Ceci nous inspire à utiliser la valeur de ces maxima pour estimer le niveau de modulation d'états du signal cyclique. Des maxima approchant 1 indiquent une modulation faible alors que des maxima plus faibles indiquent une modulation plus prononcée.

Dans le cas d'un signal cyclique par modulation temporelle, les maxima n'ont pas forcément une valeur de 1 mais, plus important encore, ces maxima ne correspondront pas nécessairement aux décalages multiples entiers de la durée moyenne des cycles. La figure 3.6 illustre cette situation. D'une façon générale, plus la modulation temporelle est faible, plus les maxima tendent vers 1 et plus ils s'approchent des décalages multiples entiers de la durée moyenne des cycles. Ceci nous inspire à utiliser la valeur de ces maxima ainsi que le déphasage des maxima pour estimer le niveau de modulation temporelle du signal cyclique. Des maxima approchant 1 avec un déphasage faible indiquent une modulation faible alors que des maxima plus faibles avec un déphasage plus important indiquent une modulation plus prononcée.

Dans le cas général d'un signal cyclique, le déphasage des maxima est un indice de modulation temporelle alors que la valeur des maxima est un indice combinant à la fois la modulation temporelle et la modulation d'états. Notre analyse ne requiert

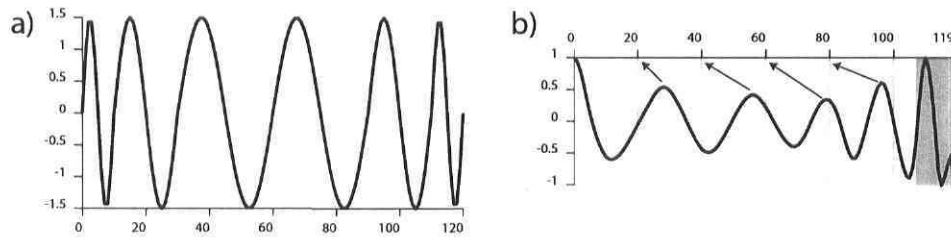


FIG. 3.6 – a) Un exemple de signal cyclique par modulation temporelle. b) La trace de l'autocorrélation de ce signal. Les flèches indiquent pour chaque maximum où il se trouverait pour un signal périodique.

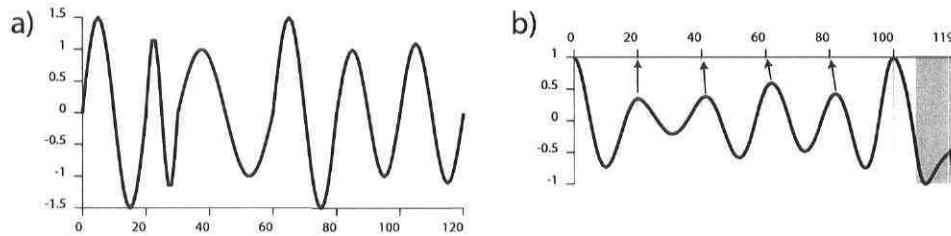


FIG. 3.7 – a) Un exemple de signal cyclique selon la définition générale. b) La trace de l'autocorrélation de ce signal.

pas nécessairement de pouvoir différencier les effets de la modulation temporelle de ceux de la modulation d'états, dans la mesure que les deux types de modulation soient considérées. La figure 3.7 illustre la situation.

### 3.2.1 Indice de périodicité

Utilisant les observations précédentes, notre idée est donc de comparer l'autocorrélation d'un signal cyclique ayant une durée moyenne de cycle  $c_S$  avec celle d'un signal périodique de période  $c_S$ . Cette analyse nous permettra de noter la périodicité du signal cyclique sur une échelle donnée. Plus précisément, nous nous attarderons à comparer les maxima de ces deux autocorrélations. L'indice prendra donc la forme d'une mesure de proximité entre les maxima attendus, soit ceux d'un signal périodique de période  $c_S$ , et les maxima réellement observés sur l'autocorrélation du signal analysé.

Soit  $M_S$ , l'ensemble ordonné des décalages auxquels nous retrouvons les maxima de l'autocorrélation  $A_S$  du signal  $S^6$ . La durée moyenne du cycle de  $S$  est estimée comme

<sup>6</sup>Pour contrer les effets du bruit possible dans le signal, nous ne considérons que les maxima supé-

suit :

$$c_S = \sum_{i=1}^{|M_S|-1} \frac{M_S(i+1) - M_S(i)}{|M_S| - 1} = \frac{M_S(|M_S|)}{|M_S| - 1} \quad (3.2)$$

Dans le même ordre d'idée,  $E_S$  est l'ensemble ordonné des décalages des maxima attendus et est défini en fonction de  $c_S$  comme suit :

$$E_S = (0, c_S, 2c_S, \dots, M_S(|M_S|)) \quad (3.3)$$

Par la définition de  $c_S$ , nous observons que le nombre de maxima observés est le même que le nombre de maxima attendus :

$$|E_S| = |M_S| \quad (3.4)$$

il est donc possible d'établir une correspondance biunivoque entre ces deux ensembles de maxima.

En utilisant ces variables, nous pouvons établir une première version de notre indice de périodicité. L'indice de périodicité  $\Psi_S$  est donc une mesure de proximité entre les paires de maxima correspondants. Pour chaque paire, l'indice de périodicité dépend de leur différence en décalage normalisée par la durée moyenne d'un cycle :

$$\Psi_S \propto 1 - \frac{|E_S(i) - M_S(i)|}{c_S} \quad (3.5)$$

et de leur différence en valeur :

$$\Psi_S \propto A_S(M_S(i)) \quad (3.6)$$

L'expression complète de l'indice de périodicité  $\Psi_S$  est donc :

$$\Psi_S = \frac{1}{|M_S| - 1} \sum_{i=2}^{|M_S|} \left(1 - \frac{|E_S(i) - M_S(i)|}{c_S}\right) \cdot A_S(M_S(i)) \quad (3.7)$$

Dans le cas d'un signal périodique, cet indice prend toujours la valeur de 1. Cette valeur diminue au fur et à mesure que ce signal est modulé. L'indice peut prendre une valeur négative pour un cas dégénéré<sup>7</sup>. Le premier maximum de  $A_S$  n'est pas considéré dans le calcul de  $\Psi_S$  étant donné qu'il correspond toujours à  $A_S(0)$ , qui est toujours parfaitement identique au maximum attendu correspondant.

rieurs aux valeurs avoisinantes dans un rayon donné.

<sup>7</sup>Un déphasage de maxima supérieur à  $c_S$  est un exemple de cas dégénéré.

L'indice  $\Psi_S$  peut toutefois souffrir de l'instabilité de l'autocorrélation lorsque  $d$  tend vers  $l_S$ . Tel que mentionné plus tôt, cette instabilité pourrait ajouter un maximum indésirable à une autocorrélation autrement presque idéale et ainsi réduire son indice de périodicité significativement. Nos expériences nous indiquent que cette région instable est généralement entièrement contenue dans le dernier 10% des valeurs de l'autocorrélation. Soient  $A'_S$ , le premier 90% des valeurs de  $A_S$ ,  $M'_S$ , l'ensemble ordonné des décalages des maxima de  $A'_S$ ,  $c'_S$ , la durée moyenne de cycle calculée sur  $A'_S$  et  $E'_S$ , l'ensemble ordonné des décalages des maxima attendus défini en fonction de  $c'_S$ . Nous pouvons définir l'indice de périodicité ajusté  $\Psi'_S$  comme suit :

$$\Psi'_S = \frac{1}{|M'_S| - 1} \sum_{i=2}^{|M'_S|} \left(1 - \frac{|E'_S(i) - M'_S(i)|}{c'_S}\right) \cdot A'_S(M'_S(i)) \quad (3.8)$$

### 3.2.2 Indice $l_S$ -normalisé

Dans notre analyse, nous aurons besoin de comparer plusieurs segments d'un signal sur la base de leur indice de périodicité. Lorsque la longueur d'un segment n'est pas un facteur, comme c'est le cas pour l'indice  $\Psi'_S$ , un segment de trois cycles a de fortes chances d'obtenir une valeur plus élevée qu'un segment de cinq ou dix cycles puisque ce dernier comportera probablement plus de différences entre les diverses itérations du cycle. Toutefois, afin d'atteindre notre objectif de segmentation complète et précise des portions cycliques, il est important de pouvoir favoriser un segment cyclique plus long par rapport à un segment cyclique plus court. Il faut toutefois faire attention à ne pas favoriser des segments non cycliques plus longs par rapport à des segments cycliques plus courts. Pour s'en assurer, il faut pouvoir quantifier les termes *cyclique* et *non cyclique*. Ceci sera fait par le biais du seuil bas  $\eta_B$  posé sur l'indice de périodicité. Seuls les segments obtenant un indice supérieur à  $\eta_B$  seront considérés cycliques.

En utilisant ce seuil  $\eta_B$ , il est possible d'élaborer un indice de périodicité normalisé par la longueur du signal  $S$  qui favorisera une longue durée pour les segments cyclique sans le faire pour les segments non cycliques. Cet indice  $\mathbf{Y}_{S[i,j]}$ , dit  $l_S$ -normalisé, s'exprime comme suit :

$$\mathbf{Y}_{S[i,j]} = \eta_B^{1 - \frac{j-i+1}{l_S}} \cdot \Psi'_{S[i,j]} \frac{j-i+1}{l_S} \quad (3.9)$$

Cet indice  $l_S$ -normalisé peut être résumé à l'aide des quelques propriétés suivantes :

- Lorsque la durée du segment  $[i, j]$  tend vers  $l_S$ ,  $\mathbf{Y}_{S[i,j]}$  tend vers  $\Psi'_{S[i,j]}$  ;
- lorsque la durée du segment  $[i, j]$  tend vers 0,  $\mathbf{Y}_{S[i,j]}$  tend vers  $\eta_B$  ;
- $(\Psi'_{S[i,j]} < \eta_B) \Leftrightarrow (\mathbf{Y}_{S[i,j]} < \eta_B)$  ;

- $(\Psi'_{S_{[i,j]}} = \eta_B) \Leftrightarrow (\mathbf{Y}_{S_{[i,j]}} = \eta_B)$  ;
- $(\Psi'_{S_{[i,j]}} > \eta_B) \Leftrightarrow (\mathbf{Y}_{S_{[i,j]}} > \eta_B)$ .

En quelques mots, en augmentant la durée d'un segment cyclique, nous augmentons la valeur de son indice  $l_S$ -normalisé alors qu'en augmentant la durée d'un segment non cyclique, nous diminuons la valeur de son indice  $l_S$ -normalisé.

Les indices  $\Psi_S$ ,  $\Psi'_S$  et  $\mathbf{Y}_S$  sont trois exemples d'implémentation de l'algorithme *IndiceP(S)* mentionné dans l'algorithme 3.1. Notons que ces trois indices ont une complexité de calcul de l'ordre de  $O(l_S^2)$  puisqu'ils demandent tous le calcul de  $A_S(d)$  pour tous les  $d \in [0, l_S]$ .<sup>8</sup>

Comme nous le verrons sous peu, l'indice  $l_S$ -normalisé sera utilisé pour identifier les segments prometteurs, puisque cet indice favorise la détection d'activités entières. Cependant, nous utiliserons l'indice  $\Psi'_S$  dans l'algorithme 3.1, puisqu'il offre une mesure de périodicité dépendant uniquement de la forme du signal, sans égard à sa durée. Cet aspect est important puisque nous nous intéressons alors à confronter la périodicité d'un segment à un seuil fixe.

### 3.2.3 Analyse d'un indice basé sur la période

Tel que nous l'avons mentionné plus tôt, l'indice de périodicité proposé par Polana et Nelson [9] ne s'applique pas très bien à notre objectif. Le problème est que nous cherchons à obtenir une segmentation précise des portions cycliques du signal. Nous espérons donc trouver des indices de périodicité plus élevés pour ces segments et les segments s'y apparentant beaucoup. Toutefois, l'indice décrit à l'équation (2.12) peut être relativement instable dans un voisinage restreint. Un segment idéal peut donc avoir un indice élevé, mais se retrouver tout prêt d'un segment ayant un pointage très faible ou nul. Ceci est dû au fait que ce pointage peut changer dramatiquement lorsque la fréquence fondamentale change. Un effet indésirable d'une telle fonction chaotique est qu'une recherche sous-optimale du segment idéal peut s'avérer significativement plus complexe. Par opposition, les indices de périodicité  $\Psi'_S$  et  $\mathbf{Y}_{S_{[i,j]}}$  sont beaucoup plus stables. La figure 3.8 compare la trace de  $\Psi'_S$  et de l'équation (2.12) pour tous les segments possibles d'un signal. En observant cet exemple, nous remarquons aussi que l'indice  $\Psi'_S$  est beaucoup plus sensible aux silences à l'intérieur d'un segment, alors que l'indice proposé par Polana accorde une valeur de périodicité très élevée à des segments englobant les deux parties cycliques à la fois, ce qui n'est pas souhaitable lorsque nous désirons procéder à une segmentation précise.

<sup>8</sup>Cette complexité peut être ramenée à  $O(l_S \cdot \log(l_S))$  en calculant  $A_S$  à l'aide de l'algorithme FFT.

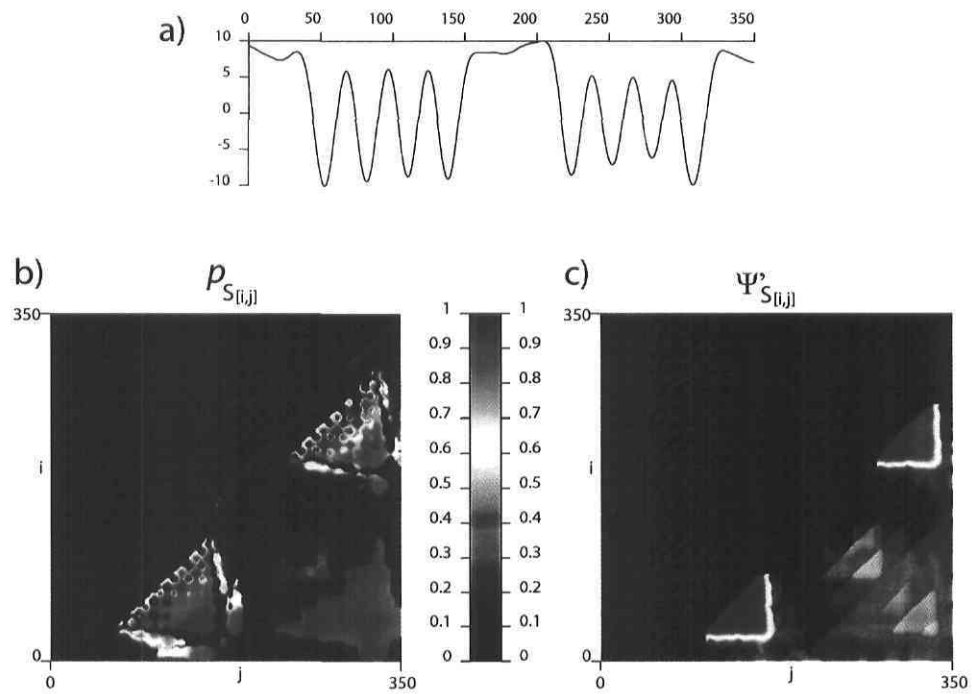


FIG. 3.8 – **a)** Exemple de signal à segmenter. **b)** Trace de l'équation (2.12) pour tous les segments  $[i, j]$  possibles sur le signal présenté en a). La correspondance entre les couleurs et les valeurs est présentée sur l'échelle à la droite de cette matrice. **c)** Trace de l'indice  $\Psi'_{S[i,j]}$  pour tous les segments  $[i, j]$  possibles sur le signal présenté en a). Les couleurs correspondent aux mêmes valeurs qu'en b).

### 3.3 Recherche du meilleur segment

Une opération vitale à notre algorithme glouton consiste à pouvoir identifier parmi tous les segments possibles celui étant le plus propice à englober précisément et entièrement une activité cyclique. Sans un algorithme remplissant cette tâche, la notion d'algorithme glouton serait perdue. Notons que pour un signal  $S$  de durée  $l_S$ , il existe  $\frac{(l_S - \beta)^2}{2}$  segments possibles, avec  $\beta$  la durée minimum d'une activité cyclique.

Considérant le grand nombre de solutions possibles à ce problème, nous en avons étudié sous trois approches distinctes : l'approche exhaustive, le sous-échantillonnage et l'optimisation numérique. Ces approches sont décrites plus en détail dans les sections suivantes.

#### 3.3.1 Approche exhaustive

L'approche exhaustive est sans doute la solution la plus simple et la plus efficace, mais elle est également la plus coûteuse au niveau du temps de calcul. Cette approche consiste à calculer l'indice de périodicité  $l_S$ -normalisé pour tous les segments de  $S$  possibles et à retourner le segment maximisant cet indice. L'algorithme 3.2 décrit cette approche en pseudo-code. Rappelons que  $\beta$  est un seuil sur la durée minimum d'une activité cyclique pouvant être détectée et que  $l_S$  représente la durée du signal  $S$ .

---

#### Algorithme 3.2 *MeilleurSegmentExhaustif(S)*

---

```

1:  $(I, J) \leftarrow (0, 0)$ 
2: pour  $i \leftarrow 1$  à  $l_S$  faire
3:   pour  $j \leftarrow i + \beta - 1$  à  $l_S$  faire
4:     si  $Y_{S_{[i,j]}} > Y_{S_{[I,J]}}$  alors
5:        $(I, J) \leftarrow (i, j)$ 
6:     fin si
7:   fin pour
8: fin pour
9: retourner  $(I, J)$ 

```

---

L'avantage indéniable de l'approche exhaustive est qu'elle retourne toujours le segment ayant l'indice le plus élevé. Donc, sous l'hypothèse que l'indice  $l_S$ -normalisé attribue bien des valeurs plus élevées aux segments les plus intéressants, aucun autre algorithme ne devrait pouvoir obtenir une segmentation plus précise que l'approche exhaustive. En pratique, l'indice  $l_S$ -normalisé n'attribue pas toujours exactement la valeur



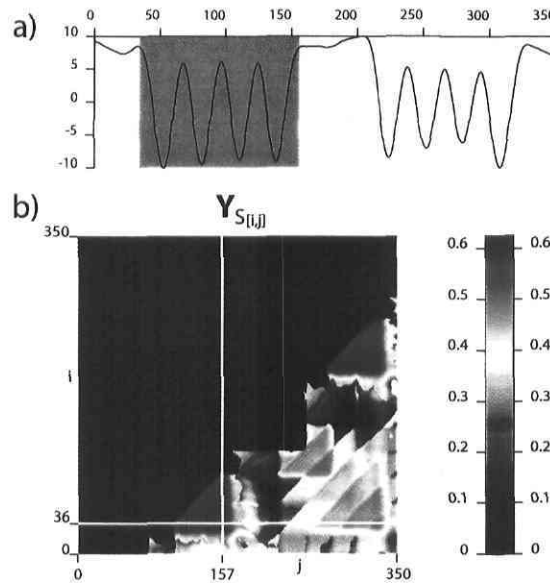


FIG. 3.9 – a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par une approche exhaustive sur ce signal. b) Matrice des valeurs calculées par l'approche exhaustive, correspond à la trace de l'indice  $l_S$ -normalisé pour tous les segments possibles. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches.

la plus élevée au segment le plus intéressant, il est donc possible qu'une autre approche soit plus précise que l'approche exhaustive pour un signal donné.

Cette exactitude s'obtient toutefois au coût de la performance. Puisqu'elle demande le calcul de l'indice de périodicité sur chaque segment possible, l'approche exhaustive a une complexité de calcul de l'ordre de  $O(l_S^2 \cdot (l_S - \beta)^2)$  que nous pouvons simplifier à  $O(l_S^4)$  lorsque  $\beta$  est petit devant  $l_S$ , ce qui est généralement le cas.

Cette approche est donc intéressante lorsque la qualité de la segmentation est beaucoup plus importante que la vitesse à laquelle elle est obtenue. Elle constitue également un bon algorithme de référence lorsque nous désirons évaluer la qualité des solutions d'autres approches par rapport à une autre segmentation automatique.

### 3.3.2 Sous-échantillonnage

Le sous-échantillonnage est une famille regroupant plusieurs solutions possibles à notre problème. L'essence de cette approche est de n'évaluer qu'un nombre restreint de

segments parmi tous les segments possibles et de pouvoir localiser le segment idéal ou un de ses proches voisins. Le fondement sur lequel se base notre utilisation de cette approche pour résoudre notre problème est qu'un segment d'indice élevé est généralement entouré par un voisinage de segments d'indices élevés.

### Sous-échantillonnage uniforme

Cette méthode de sous-échantillonnage bien connue consiste à évaluer une fonction à des points équidistants. En procédant par sous-échantillonnage uniforme sur l'indice  $l_S$ -normalisé avec un pas  $\Delta_U$ , nous n'évaluons donc que l'indice pour des segments  $S_{[i,j]}$  dont les indices  $i$  et  $j$  sont respectivement un multiple entier de  $\Delta_U$  et  $\beta$  additionné d'un multiple entier de  $\Delta_U$ . Nous obtenons alors l'algorithme 3.3.

---

#### Algorithme 3.3 *MeilleurSegmentUniforme(S)*

---

```

1:  $(I, J) \leftarrow (0, 0)$ 
2: pour  $i \leftarrow 1$  à  $\lfloor \frac{l_S}{\Delta_U} \rfloor$  faire
3:   pour  $j \leftarrow i + \lfloor \frac{\beta}{\Delta_U} \rfloor - 1$  à  $\lfloor \frac{l_S}{\Delta_U} \rfloor$  faire
4:     si  $Y_{S_{[i\Delta_U, j\Delta_U]}} > Y_{S_{[I, J]}}$  alors
5:        $(I, J) \leftarrow (i\Delta_U, j\Delta_U)$ 
6:     fin si
7:   fin pour
8: fin pour
9: retourner  $(I, J)$ 

```

---

Le sous-échantillonnage uniforme comporte un avantage indéniable sur l'approche exhaustive au niveau de la complexité du calcul puisqu'il ne demande qu'une fraction des calculs de l'approche exhaustive. En fait, l'algorithme 3.3 a une complexité de calcul de l'ordre de  $O(l_S^2 \cdot \frac{l_S}{\Delta_U})$  ce qui est un gain significatif par rapport à l'algorithme 3.2 lorsque  $\Delta_U$  est suffisamment grand.

L'envers de la médaille est la perte d'exactitude de la solution trouvée. En effet, si le segment idéal n'est pas parmi les points échantillonnés, il ne pourra être trouvé. Généralement, l'algorithme trouvera un segment voisin du segment idéal, mais, si le pas  $\Delta_U$  est trop grand, il est également possible que tous les points échantillonnés les plus proches du segment idéal aient une valeur faible. Ceci risque alors d'entraîner une détection manquée. De plus, bien que cet algorithme soit plus rapide que l'approche exhaustive, il peut demeurer coûteux en temps de calcul lorsque le pas d'échantillonnage est relativement faible.

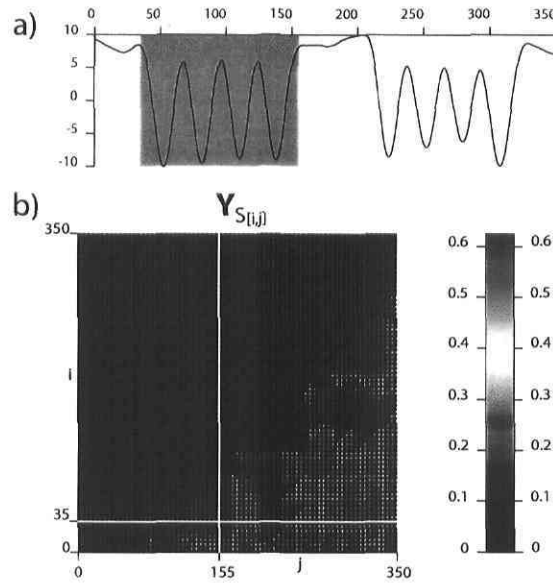


FIG. 3.10 – a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par sous-échantillonnage uniforme avec  $\Delta_U = 5$ . b) Matrice des valeurs calculées par sous-échantillonnage uniforme avec  $\Delta_U = 5$ . Les points ont été grossis pour en améliorer la lisibilité. La zone noire représente les valeurs inconnues. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches.

La figure 3.10 présente un exemple de sous-échantillonnage uniforme. Nous pouvons remarquer que le segment trouvé ici est très proche du segment trouvé par l'approche exhaustive, bien que le sous-échantillonnage uniforme n'évalue ici qu'environ 4% des segments évalués par l'approche exhaustive.

### Sous-échantillonnage sélectif

Nous pouvons chercher à augmenter l'exactitude obtenue par le sous-échantillonnage uniforme sans en augmenter significativement la complexité de calcul. L'idée de base est de procéder à un échantillonnage grossier des segments possibles et de raffiner l'échantillonnage dans les régions prometteuses de façon itérative. Pour ce faire, nous réduisons le pas  $\Delta_S$  de moitié à chaque itération et nous évaluons seulement les segments qui sont voisins de segments déjà évalués ayant une valeur supérieure à une borne fonction de la moyenne des indices calculés, de l'indice maximum calculé et d'une constante de sélection  $\gamma$ .

L'algorithme 3.4 peut sembler légèrement complexe au premier abord. Pour en sim-

**Algorithme 3.4** *MeilleurSegmentSelectif(S)*


---

```

1:  $M \leftarrow (-1)_{l_S \times l_S}$  {Une matrice initialisée à -1}
2:  $(I, J) \leftarrow (0, 0)$ 
3: {Première itération (non sélective)}
4:  $\Delta_S \leftarrow 2^D$ 
5: pour  $i \leftarrow 1$  à  $\lfloor \frac{l_S}{\Delta_S} \rfloor$  faire
6:   pour  $j \leftarrow i + \lfloor \frac{\beta}{\Delta_S} \rfloor - 1$  à  $\lfloor \frac{l_S}{\Delta_S} \rfloor$  faire
7:      $M_{i\Delta_S, j\Delta_S} \leftarrow \mathbf{Y}_{S_{[i\Delta_S, j\Delta_S]}}$ 
8:     si  $M_{i\Delta_S, j\Delta_S} > M_{I, J}$  alors
9:        $(I, J) \leftarrow (i\Delta_S, j\Delta_S)$ 
10:    fin si
11:  fin pour
12: fin pour
13: {Itérations successives (sélectives)}
14: tant que  $D > 0$  faire
15:    $D \leftarrow D - 1$ 
16:    $\Delta_S \leftarrow 2^D$ 
17:   pour  $i \leftarrow 1$  à  $\lfloor \frac{l_S}{\Delta_S} \rfloor$  faire
18:     pour  $j \leftarrow i + \lfloor \frac{\beta}{\Delta_S} \rfloor - 1$  à  $\lfloor \frac{l_S}{\Delta_S} \rfloor$  faire
19:       si  $M_{i\Delta_S, j\Delta_S} = -1$  alors
20:         si  $\gamma \in [0, 0.5]$  alors
21:           {Interpolation linéaire entre 0 et la moyenne  $\bar{M}$  des indices connus}
22:            $IndiceMin \leftarrow 2\bar{M}\gamma$ 
23:         sinon si  $\gamma \in ]0.5, 1]$  alors
24:           {Interpolation linéaire entre  $\bar{M}$  et l'indice maximum connu}
25:            $IndiceMin \leftarrow \bar{M} + (M_{I, J} - \bar{M})(\gamma - 0.5)$ 
26:         fin si
27:         si  $\max_{a \in [i-0.5]\Delta_S, [i+0.5]\Delta_S}, b \in [j-0.5]\Delta_S, [j+0.5]\Delta_S} (M_{a, b}) > IndiceMin$  alors
28:            $M_{i\Delta_S, j\Delta_S} \leftarrow \mathbf{Y}_{S_{[i\Delta_S, j\Delta_S]}}$ 
29:           si  $M_{i\Delta_S, j\Delta_S} > M_{I, J}$  alors
30:              $(I, J) \leftarrow (i\Delta_S, j\Delta_S)$ 
31:           fin si
32:         fin si
33:       fin si
34:     fin pour
35:   fin pour
36: fin tant que
37: retourner  $(I, J)$ 

```

---

plifier la compréhension, nous pouvons remarquer que les lignes numérotées 4 à 12 sont une première itération d'échantillonnage uniforme utilisant le pas d'échantillonnage maximal. Les lignes suivantes procèdent à l'échantillonnage sélectif jusqu'à l'atteinte d'un pas  $\Delta_S = 1$ . La condition de la ligne 19 ne sert qu'à assurer que l'indice d'un segment n'est pas calculé deux fois. Les lignes numérotées de 20 à 26 servent à déterminer quelle valeur d'indice correspond au seuil  $\gamma$  donné en fonction des indices des segments déjà calculés. Un seuil  $\gamma = 0$  correspond à un indice de 0, un seuil  $\gamma = 0.5$  correspond à l'indice moyen des segments analysés et un seuil  $\gamma = 1$  correspond à l'indice maximal des segments calculés. Pour des valeurs intermédiaires de  $\gamma$ , l'indice est déterminé par interpolation linéaire entre ces points. La condition à la ligne 27 vérifie qu'un segment se trouvant à l'intérieur d'un pas  $\Delta_S$  de distance a été analysé et possède un indice de périodicité supérieur à l'indice seuil déterminé plus tôt. Lorsque c'est le cas, l'indice de ce segment est calculé et, s'il obtient un indice supérieur au maximum actuel, il est conservé comme meilleur segment.

Puisque l'algorithme termine toujours avec un pas unitaire, tous les segments possibles peuvent être considérés. Cela rend possible une exactitude équivalente à celle de l'approche exhaustive. Le coût de ce gain d'exactitude se voit au niveau de la complexité de calcul. Elle est fonction du degré de pas initial  $D$  et de la constante de sélection  $\gamma$ . Une valeur de  $D$  plus élevée réduit le nombre de segments à évaluer, alors qu'une valeur de  $D$  plus proche de 0 vient augmenter le nombre de segments à évaluer. L'effet de  $\gamma$  sur la complexité dépend de la distribution des valeurs d'indice de périodicité, mais un  $\gamma$  élevé viendra toujours réduire la complexité alors qu'un  $\gamma$  faible viendra l'augmenter. Bien qu'elle soit difficile à exprimer sans connaître la distribution des indices de périodicité, la complexité de cet algorithme se retrouve entre celle de l'approche exhaustive et celle du sous-échantillonnage uniforme. Par contre, pour des choix de  $D$  et  $\gamma$  suffisamment élevés, elle s'approchera davantage de la complexité du sous-échantillonnage uniforme.

La figure 3.11 présente un exemple de sous-échantillonnage sélectif. Nous remarquons que dans cet exemple, le segment trouvé par sous-échantillonnage sélectif est identique à celui trouvé par l'approche exhaustive. Dans cet exemple, le sous-échantillonnage sélectif n'évalue qu'environ 10% des segments évalués par l'approche exhaustive.

### 3.3.3 Optimisation numérique

Les méthodes d'optimisation numérique sont des algorithmes cherchant à maximiser ou minimiser la valeur d'une fonction donnée. La littérature en intelligence artificielle regorge de tels algorithmes. La majorité de ces algorithmes s'appuient sur l'hypothèse

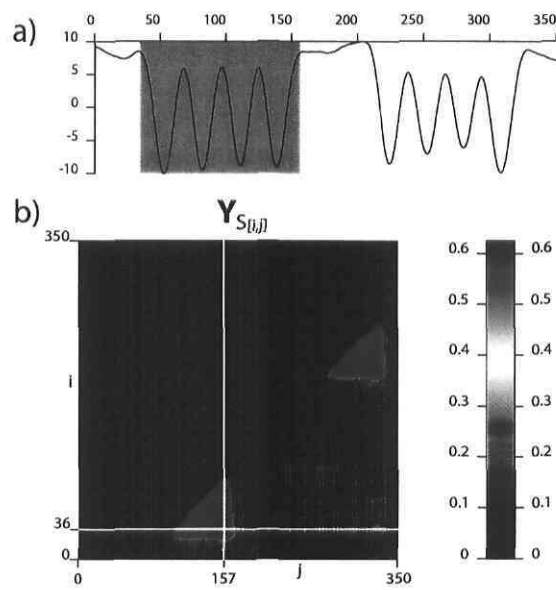


FIG. 3.11 – a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par sous-échantillonnage sélectif avec  $D = 4$  et  $\gamma = 0.75$ . b) Matrice des valeurs calculées par sous-échantillonnage sélectif. La zone noire représente les valeurs inconnues. Le segment obtenu par l'approche exhaustive est dénoté par l'intersection des lignes blanches.

que la fonction à optimiser est stable dans un voisinage proche. En d'autres termes, tous les points de cette fonction doivent avoir des valeurs s'approchant des valeurs aux points voisins. Ceci est important puisque les algorithmes d'optimisation numérique vont généralement estimer le relief local d'une fonction à partir de quelques points connus. Si le relief ne peut être estimé de la sorte, ces algorithmes sont voués à l'échec. Heureusement, l'indice  $\mathbf{Y}_{S_{[i,j]}}$  remplit généralement cette condition.

Selon la définition que nous en avons donnée, l'optimisation numérique est également une forme de sous-échantillonnage. Les algorithmes d'optimisation utilisent toutefois généralement significativement moins d'évaluations de la fonction pour arriver à leur fin. Ceci est dû au fait qu'ils utilisent des informations de plus haut niveau sur la fonction, en estimant localement son gradient par exemple.

Tel que nous l'avons mentionné, il existe un grand nombre d'algorithmes d'optimisation numérique. Notre objectif n'étant pas de les comparer entre eux autant que d'avoir une bonne idée de ce que l'optimisation numérique pourrait contribuer à notre approche, nous n'avons testé qu'un seul algorithme. Nous avons choisi d'utiliser l'algorithme de *Dynamic Hill-Climbing* (DHC) présenté par Yuret dans [16] étant donné sa capacité éprouvée à optimiser des fonctions fortement irrégulières et le nombre restreint d'évaluations qu'il demande.

En quelques mots, l'algorithme DHC s'adapte à la topologie locale de la fonction pour trouver un maximum local en un nombre restreint d'étapes. L'algorithme s'adapte à la topologie en estimant grossièrement le gradient de la fonction et en ajustant la distance entre les points évalués en fonction du nombre d'évaluations consécutives dans une même direction.

Du fait qu'il estime la topologie locale, l'algorithme DHC, comme plusieurs autres algorithmes d'optimisation numérique, s'applique mieux sur des fonctions de domaine infini. Plus précisément, l'algorithme aura de la difficulté à évaluer la topologie locale près des limites du domaine. Cela peut donc se traduire par une détection manquée d'un segment cyclique qui débute au début du signal ou qui se termine à la fin du signal. Pour corriger ce problème, nous reflétons la fonction  $\mathbf{Y}_{S_{[i,j]}}$  lorsque nous désirons l'évaluer au-delà de ses frontières. La figure 3.12 illustre ce processus. L'étoile à la fin de  $\mathbf{Y}_{S_{[i,j]}}^*$  indique que nous référons à cette nouvelle version de la fonction  $l_S$ -normalisée. Nous pouvons noter que seules les trois réflexions présentées à la figure 3.12 sont nécessaires. Puisque le contour du domaine ainsi formé est toujours de valeur nulle, soit la valeur la plus basse admissible par la fonction, l'algorithme DHC n'ira jamais explorer à l'extérieur de ce domaine.

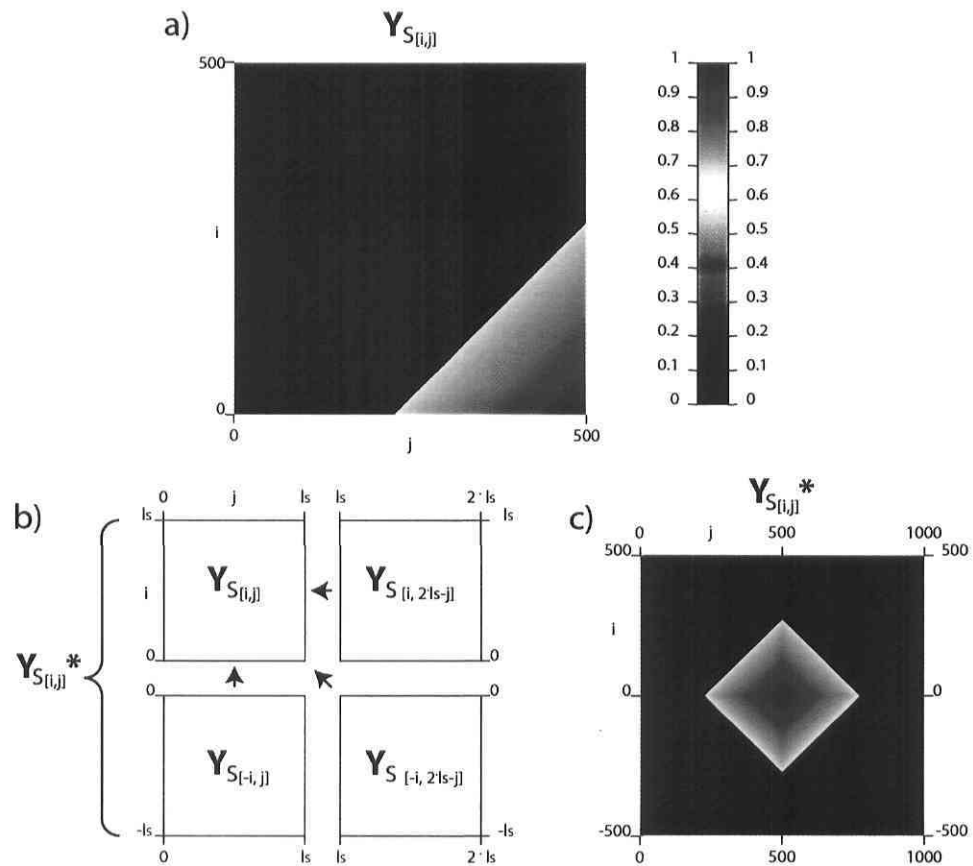


FIG. 3.12 – a) Indice  $l_S$ -normalisé calculé pour tous les segments d'un signal sinusoïdal. Nous remarquons que le maximum se retrouve aux limites de la fonction. b) Schéma illustrant comment la réflexion de la fonction est effectuée. Seules ces trois réflexions sont nécessaires au bon fonctionnement de notre algorithme. c) Le résultat des réflexions appliqué à la matrice présentée en a). Nous remarquons que le maximum ne se retrouve plus sur une frontière.



Tel que suggéré dans [16], notre implémentation choisit  $N$  points de départ aléatoires pour y lancer les recherches de maxima locaux. Nous utilisons également la méthode suggérée dans [16] pour positionner un nouveau point de départ sur une portion de la fonction qui n'a pas déjà été couverte par les recherches précédentes. Nous nommerons cette procédure *SegmentAleatoire*( $S$ ). Le maximum de plus grande valeur parmi les  $N$  maxima trouvés est ensuite retenu comme maximum global estimé de la fonction. L'algorithme 3.5 décrit cette procédure en pseudo-code.

---

**Algorithme 3.5** *MeilleurSegmentDHC*( $S$ )
 

---

```

1:  $(I, J) \leftarrow (0, 0)$ 
2: pour  $n \leftarrow 1$  à  $N$  faire
3:    $(i_0, j_0) \leftarrow \text{SegmentAleatoire}(S)$ 
4:    $(i_m, j_m) \leftarrow \text{DHC}(S, (i_0, j_0), \mathbf{Y}^*)$ 
5:   si  $\mathbf{Y}_{S_{[i_m, j_m]}}^* > \mathbf{Y}_{S_{[I, J]}}^*$  alors
6:      $(I, J) \leftarrow (i_m, j_m)$ 
7:   fin si
8: fin pour
9: si  $I < 0$  alors
10:   $I \leftarrow -I$ 
11: fin si
12: si  $J > |S|$  alors
13:   $J \leftarrow 2 \cdot |S| - J$ 
14: fin si
15: retourner  $(I, J)$ 

```

---

La complexité de l'algorithme 3.5 dépend d'au moins deux facteurs difficiles à modéliser formellement : la forme de la fonction  $\mathbf{Y}_{S_{[i, j]}}$  pour un signal  $S$  donné et la position aléatoire des points de départ de chaque recherche. Pour cette raison, il est très difficile de poser une borne théorique précise sur cette complexité. Toutefois, nos études empiriques nous indiquent clairement un temps d'exécution de beaucoup moindre par rapport à l'approche exhaustive.

Alors que l'approche exhaustive garantit de trouver le maximum global réel et que le sous-échantillonnage sélectif le fait aussi sous la condition que le pas initial ne soit pas trop grand, l'algorithme 3.5 ne peut offrir une telle garantie. Une fois de plus, ceci est dû au fait que le succès de son exécution dépend directement de la forme de la fonction  $\mathbf{Y}_{S_{[i, j]}}$  et des choix de points de départ des recherches.

La figure 3.13 illustre grossièrement une itération de l'algorithme DHC pour isoler un segment cyclique dans un signal. Il est important de noter que la trace de la fonction  $\mathbf{Y}_{S_{[i, j]}}$  n'est fournie qu'à titre indicatif, puisqu'à l'intérieur d'une itération, l'algorithme

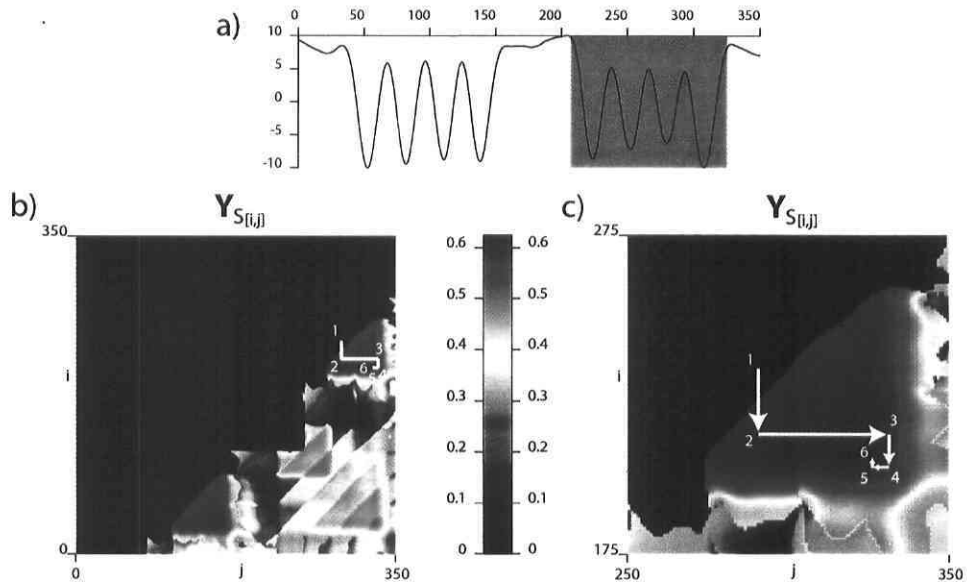


FIG. 3.13 – a) Exemple de signal à segmenter. La zone bleue représente le segment trouvé par une itération de l'algorithme DHC. b) Trace de la fonction  $Y_{S[i,j]}$  pour le signal donné en a). Le tracé blanc représente les étapes d'optimisation de l'algorithme DHC. c) Tracé partiel de la fonction, permettant de mieux voir les étapes effectuées par l'algorithme DHC.

DHC n'évalue la fonction que pour un nombre très petit de segments, typiquement bien en deçà d'une centaine.

### 3.4 Simplification du problème

Tel que nous l'avons expliqué au début de ce chapitre, le problème de la segmentation automatique des segments cycliques d'un signal est un défi de taille. L'approche *glouton* présentée plus tôt vient alléger le problème partiellement, en cherchant de façon itérative le meilleur segment restant. Cela simplifie le problème en ce sens qu'elle ne cherche qu'un segment à la fois, ce qui correspond à fixer uniquement deux variables, au lieu de s'attarder à trouver tous les segments simultanément, où le nombre de variables serait beaucoup plus grand.

Il existe toutefois d'autres façons de réduire la complexité de ce problème. Nous présentons ici celles que nous avons mises en place dans le cadre de nos recherches.

### 3.4.1 Hypothèses simplificatrices

Poser des hypothèses sur la forme des séquences à traiter vient restreindre le champ d'application de notre solution, mais elle peut également réduire la complexité du problème. Heureusement, il est possible de poser des hypothèses ne donnant que de faibles contraintes tout en améliorant significativement la performance de notre solution.

Tout d'abord, nos expérimentations nous ont montré qu'il était extrêmement difficile de détecter des activités cycliques ne comportant que deux cycles. Le problème n'étant pas de détecter les activités cycliques réelles, mais plutôt de minimiser le nombre de détections erronées. En posant une limite de trois cycles minimum, le nombre de détection erronée chute généralement bien en deçà d'un seuil acceptable tout en ne restreignant pas sévèrement les activités cycliques pouvant être détectées.

Toujours dans le même objectif, nous pouvons déterminer quelle sera la fréquence maximale d'une activité cyclique devant être détectée. À titre d'exemple, en analyse de mouvements humains, il serait surprenant d'avoir à détecter des mouvements au-delà de 5Hz. Cette hypothèse, qui est ajustée en fonction de l'application, ne restreindra généralement pas du tout le type d'activités cycliques pouvant être détecté. Elle sert à déterminer la durée minimum d'un cycle et surtout, sachant que nous exigeons un minimum de trois cycles, à déterminer la durée minimale d'une activité cyclique devant être détectée<sup>9</sup>. Cela nous permet donc d'ignorer tous les segments possibles de taille inférieurs à cette limite.

### 3.4.2 Soustraction des pauses

Une autre façon de simplifier le problème auquel nous faisons face se base sur l'hypothèse que les différentes activités cycliques perceptibles sur un signal ont une amplitude du même ordre de grandeur. Cette hypothèse est généralement vérifiée pour des activités cycliques humaines où les membres font généralement des mouvements de mêmes amplitudes. À titre d'exemple, un bras fera généralement de grands mouvements alors que les doigts en feront généralement de petits.

Sous cette hypothèse, il est possible d'identifier les parties du signal où aucun mouvement significatif n'a lieu. En soustrayant de telles pauses d'un signal  $S$ , il ne nous reste qu'à segmenter les portions restantes du signal. Cela peut réduire considérablement la complexité du problème puisque nous n'avons plus à traiter un signal de longueur  $l_S$

---

<sup>9</sup>Cette durée minimale est représentée dans nos algorithmes par le paramètre  $\beta$ .

mais seulement des portions de ce signal. Puisque le nombre de segments possibles augmente proportionnellement au carré de  $l_S$ , le fait de traiter plusieurs portions d'un signal au lieu de traiter le signal complet vient réduire significativement le nombre de segments possibles. Le gain réel est évidemment fonction du nombre de pauses contenues dans le signal. À titre d'exemple, le traitement d'un signal où chaque activité cyclique est immédiatement suivie d'une pause profitera énormément de cette approche.

L'algorithme 3.6 décrit la segmentation et la soustraction des pauses dans un signal  $S$ . En quelques mots, l'algorithme parcourt le signal  $S$  du début à la fin en y cherchant une suite de plus de  $\beta_P$  valeurs consécutives dont l'amplitude ne dépasse pas un faible pourcentage  $\epsilon$  multiplié par l'amplitude du signal  $S$ . Les segments ainsi trouvés sont identifiés comme des pauses dans le signal. L'algorithme retourne donc les segments du signal se retrouvant entre ces pauses, dans un ensemble de segments  $\mathbb{Y}$ .

---

**Algorithme 3.6** *SoustractionPauses*( $S$ )
 

---

```

1:  $\mathbb{X} \leftarrow \emptyset$ 
2:  $\mathbb{Y} \leftarrow \emptyset$ 
3: {Segmentation des pauses}
4:  $i \leftarrow 1$ 
5: pour  $j \leftarrow 2$  à  $l_S$  faire
6:   si  $\max(S_{[i,j]}) - \min(S_{[i,j]}) > \epsilon(\max(S_{[1,l_S]}) - \min(S_{[1,l_S]}))$  alors
7:     si  $j - i + 1 > \beta_P$  alors
8:        $\mathbb{X} \leftarrow \mathbb{X} \cup \{(i, j)\}$ 
9:     fin si
10:     $i \leftarrow j$ 
11:  fin si
12: fin pour
13: {Soustraction des pauses}
14:  $a \leftarrow 1$ 
15: tant que  $\mathbb{X} \neq \emptyset$  faire
16:    $(i, j) \leftarrow \mathbb{X}(1)$  {Le premier segment dans  $\mathbb{X}$ }
17:    $\mathbb{Y} \leftarrow \mathbb{Y} \cup \{(a, i - 1)\}$ 
18:    $a \leftarrow j + 1$ 
19: fin tant que
20:  $\mathbb{Y} \leftarrow \mathbb{Y} \cup \{(a, l_S)\}$ 
21: retourner  $\mathbb{Y}$ 

```

---

En ajoutant une étape préalable de soustraction de pauses à notre algorithme de segmentation *glouton*, nous obtenons l'algorithme 3.7. La seule différence entre ces deux algorithmes se trouve à la deuxième ligne, où l'ensemble  $\mathbb{X}$  est initialisé par les segments retournés par la soustraction de pauses sur  $S$  au lieu d'être initialisé à la totalité du

signal  $S$ .

---

**Algorithme 3.7** *SegmenterSignalGlouton'(S)*


---

```

1: SEG  $\leftarrow \emptyset$ 
2:  $\mathbb{X} \leftarrow \text{SoustractionPauses}(S)$ 
3: tant que  $\mathbb{X} \neq \emptyset$  faire
4:    $(I, J) \leftarrow$  élément arbitraire de  $\mathbb{X}$ 
5:   si  $J - I + 1 \geq \beta$  alors
6:      $(i, j) \leftarrow \text{MeilleurSegment}(S_{[I, J]})$ 
7:     si  $\text{IndiceP}(S_{[i, j]}) > \eta_E$  alors
8:       SEG  $\leftarrow \text{SEG} \cup \{(i, j)\}$ 
9:     fin si
10:    si  $i - 1 > I$  alors
11:       $\mathbb{X} \leftarrow \mathbb{X} \cup \{(I, i - 1)\}$ 
12:    fin si
13:    si  $j + 1 < J$  alors
14:       $\mathbb{X} \leftarrow \mathbb{X} \cup \{(j + 1, J)\}$ 
15:    fin si
16:  fin si
17:   $\mathbb{X} \leftarrow \mathbb{X} / \{(I, J)\}$ 
18: fin tant que
19: retourner SEG

```

---

# Chapitre 4

## Segmentation d'une séquence de squelettes

Maintenant que nous sommes dotés d'une méthode de segmentation des parties cycliques d'un signal, nous pouvons revenir au problème initial, soit celui de segmenter en activités cycliques une séquence de squelettes.

Avant de s'attaquer au problème, il convient de définir formellement les termes *topologie squelettique*, *squelette* et *séquence de squelettes*, qui sont d'une grande importance dans ce chapitre.

### Définition 4.0.1 : Topologie squelettique

Une topologie squelettique est un graphe simple  $G = (V, E)$  où  $V$  est un ensemble fini non vide de sommets et  $E$  est un ensemble d'arcs, soient des couples d'éléments distincts de  $V$ .

### Définition 4.0.2 : Pose

Une pose  $M$  est un ensemble contraint et fini de paires  $m = (v, p)$  où  $v$  est un sommet de graphe et  $p$  est une position bidimensionnelle ou tridimensionnelle. Les contraintes sur l'ensemble sont que toutes les positions  $p$  de l'ensemble doivent être de même dimensionnalité et que tous les sommets  $v$  de l'ensemble doivent être distincts.

### Définition 4.0.3 : Squelette

Un squelette est une paire contrainte  $s = (G, M)$  associant une topologie  $G$  à une pose  $M$ . La contrainte sur la paire est que chaque sommet  $v \in V$  dans la topologie  $G$  doit être positionné par la pose  $M$ .

**Définition 4.0.4 : Pose temporelle**

Une pose temporelle est une paire  $T = (M, t)$  associant une pose  $M$  à un temps  $t$ .

**Définition 4.0.5 : Séquence de squelettes**

Une séquence de squelettes est une paire  $S = (G, D)$  associant une topologie squelettique  $G$  et un ensemble contraint et fini de poses temporelles  $D$ . Les contraintes sur l'ensemble  $D = \{T_0, T_1, T_2, \dots, T_n\}$  sont que chaque temps  $t$  des poses temporelles  $T$  doivent être unique dans  $D$  et que chaque  $v \in V$  dans la topologie  $G$  doit être positionné par chaque pose  $M$  des poses temporelles  $T$ .

## 4.1 Segmentation anatomique

Puisque, dans le contexte de cet ouvrage, les squelettes sont utilisés pour représenter des sujets humains, il est possible de procéder à une segmentation anatomique rudimentaire du squelette. Ceci nous permettra de connaître quelles parties du corps<sup>1</sup> sont impliquées dans une activité cyclique et rendra également possible la détection de plusieurs activités cycliques simultanées. Cette approche nous permettra également de découpler les mouvements globaux des mouvements relatifs. Il s'agit là d'avantages découlant de notre approche de haut niveau.

Spécifions que, dans le contexte de nos recherches, nous définissons un mouvement global comme une translation ou une rotation affectant le torse d'un individu. De tels mouvements sont qualifiés de globaux puisqu'ils entraînent un déplacement sur toutes les parties du corps. À l'inverse, nous définissons un mouvement relatif comme étant un mouvement qui n'affecte que certaines parties du corps. De tels mouvements sont générés par des flexions et des extensions des articulations.

Afin de demeurer le plus générique possible dans notre approche et de ne pas poser de contraintes additionnelles sur la topologie du squelette, la segmentation anatomique effectuée est plutôt grossière. Cette étape n'est pas automatisée, mais elle ne doit être effectuée qu'une fois par topologie squelettique utilisée.

Tout d'abord, pour une topologie donnée, il faut identifier quels sont les sommets appartenant au torse. Les arcs liant deux sommets appartenant au torse appartiennent également au torse. Ceci nous permet de définir le torse simplement comme un sous-

---

<sup>1</sup>Dans le contexte de nos recherches, une partie du corps est soit un bras, une jambe, la tête ou le torse d'un individu.

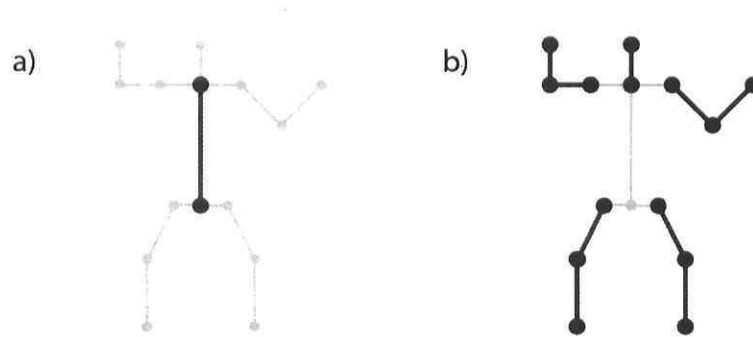


FIG. 4.1 – En a), les sommets et arcs tracés en bleu représentent le sous-graphe défini pour le torse. En b), les sommets et arcs tracés en bleu représentent les sous-graphes définis pour chaque membre de la topologie squelettique.

graphe de la topologie squelettique donnée. Notons que le torse doit former une composante connexe. L'information de mouvement de ces sommets et arcs sera utilisée pour caractériser le mouvement global du squelette. Ceci nous sera donc utile pour procéder à la segmentation d'activités cycliques globales, telles que sauter sur place ou se balancer de droite à gauche. En général, nous formerons le torse à partir des éléments de la topologie représentant la colonne vertébrale. La figure 4.1a illustre un exemple de torse pour une topologie squelettique donnée.

Une fois les sommets du torse identifiés, il faut identifier les sommets de la topologie appartenant à chaque membre<sup>2</sup>. Tout arc dans la topologie reliant deux sommets appartenant au même membre est également associé à ce membre. Les membres sont donc également des sous-graphes de la topologie squelettique. Chaque membre doit former une composante connexe en forme de chaîne : les deux extrémités ne sont liées qu'à un autre sommet alors que les autres sommets sont tous liés à exactement deux sommets. Nous devons également identifier laquelle des deux extrémités de chaque membre s'attache au torse. Notons qu'un sommet ne peut être associé à plus d'un membre, mais un sommet extrémité d'un membre peut aussi être associé au torse. Notons également que la topologie peut comporter autant de membres que nécessaire. La figure 4.1b illustre un exemple de composition de membres pour une topologie squelettique donnée.

## 4.2 Sélection des signaux

Afin de pouvoir profiter de notre approche de segmentation de signaux, il convient de pouvoir représenter adéquatement les mouvements du sujet par des signaux. Il va de

<sup>2</sup>Ici, le terme *membre* peut désigner un bras, une jambe ou même la tête d'un individu.



soi qu'aucun signal unidimensionnel ne peut représenter adéquatement tous les mouvements possibles par le sujet, nous optons donc pour l'utilisation d'un ensemble de signaux. Par représentation adéquate, nous entendons que toute activité cyclique effectuée dans la séquence de squelettes doit pouvoir être détectée comme une partie cyclique dans au moins un des signaux choisis pour représenter la séquence. Le choix de ces signaux découle de la segmentation anatomique que nous avons effectuée.

Pour représenter les mouvements globaux du squelette dans la séquence, nous utilisons l'information donnée par les noeuds du torse. Celle-ci nous permettra de segmenter les activités cycliques globales, telles que sauter sur place ou se balancer de droite à gauche. Pour caractériser ces mouvements, nous avons choisi de former des signaux à partir des positions absolues en abscisse et en ordonnée de chaque noeud du torse. Ces signaux servent à caractériser les mouvements de translation. Afin de caractériser les mouvements de rotation, nous formons également des signaux à partir des angles absolus de chaque arc dans le torse. La figure 4.2 illustre la formation de ces signaux.

Pour représenter les mouvements relatifs du squelette dans la séquence, nous utilisons l'information donnée par les noeuds de chaque membre indépendamment. Cette information servira à la segmentation des activités cycliques relatives du squelette. Évidemment, cela implique de ne considérer que le mouvement relatif de chaque noeud, et c'est de là que provient l'utilité d'identifier pour chaque membre quelle extrémité se fixe au torse. En considérant que ce noeud est fixe, tous les mouvements possibles par un membre peuvent être décrits en fonction de changement d'angles aux articulations. Il suffit alors de caractériser les flexions et les extensions à chaque articulation du membre. Afin de se protéger des effets du bruit possible dans les signaux, il est préférable d'utiliser plus d'un signal pour caractériser le mouvement à chaque articulation. Ainsi, pour chaque articulation du membre, soit chaque noeud du membre à l'exception de l'extrémité n'étant pas attachée au torse, un premier signal est formé à partir de la valeur de l'angle intérieur qu'il forme avec ses deux arcs. Deux autres signaux sont formés à partir des positions relatives en abscisse et en ordonnée du noeud suivant<sup>3</sup>, soit le noeud adjacent le plus distant du torse en terme de topologie. La figure 4.3 illustre la formation de ces signaux.

### 4.3 Segmentation d'un ensemble de signaux

Nous travaillons sous l'hypothèse que plusieurs activités cycliques peuvent se produire en même temps sur un même squelette, mais qu'une seule activité cyclique peut

---

<sup>3</sup>Les positions sont notées relativement à celle du noeud du membre fixé au torse.

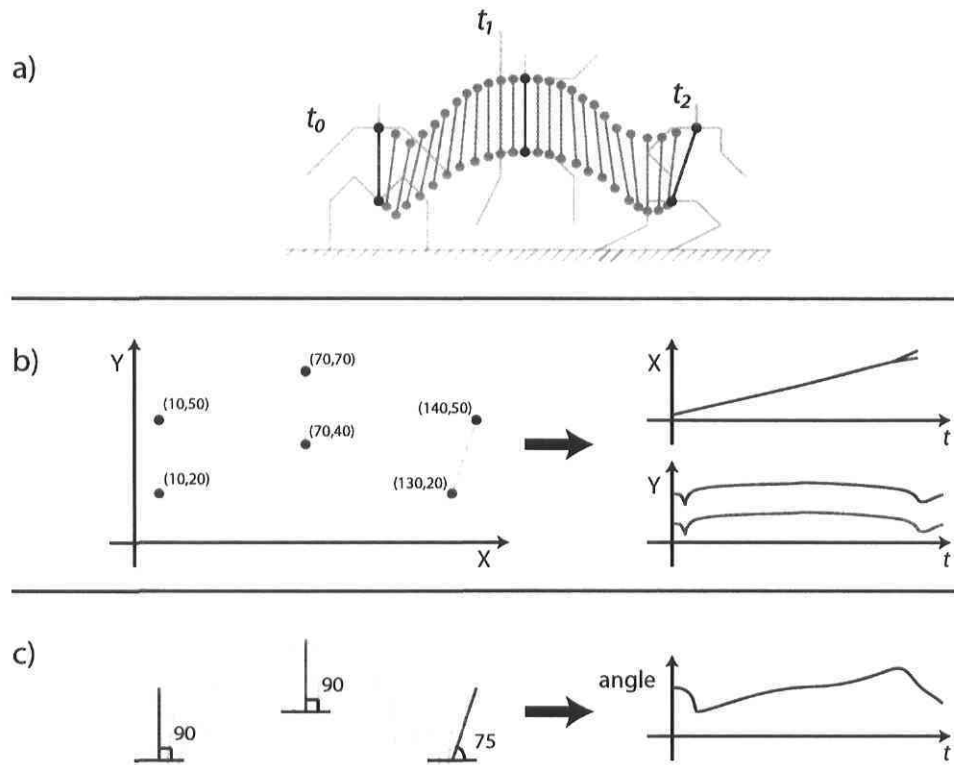


FIG. 4.2 – Illustration des signaux utilisés pour caractériser les mouvements globaux. a) Mouvement de saut vers la droite. b) Approximation des signaux formés à partir des positions en abscisse et en ordonnée pour chaque noeud du torse. c) Approximation du signal formé à partir de l'angle absolu de l'arc contenu dans le torse.

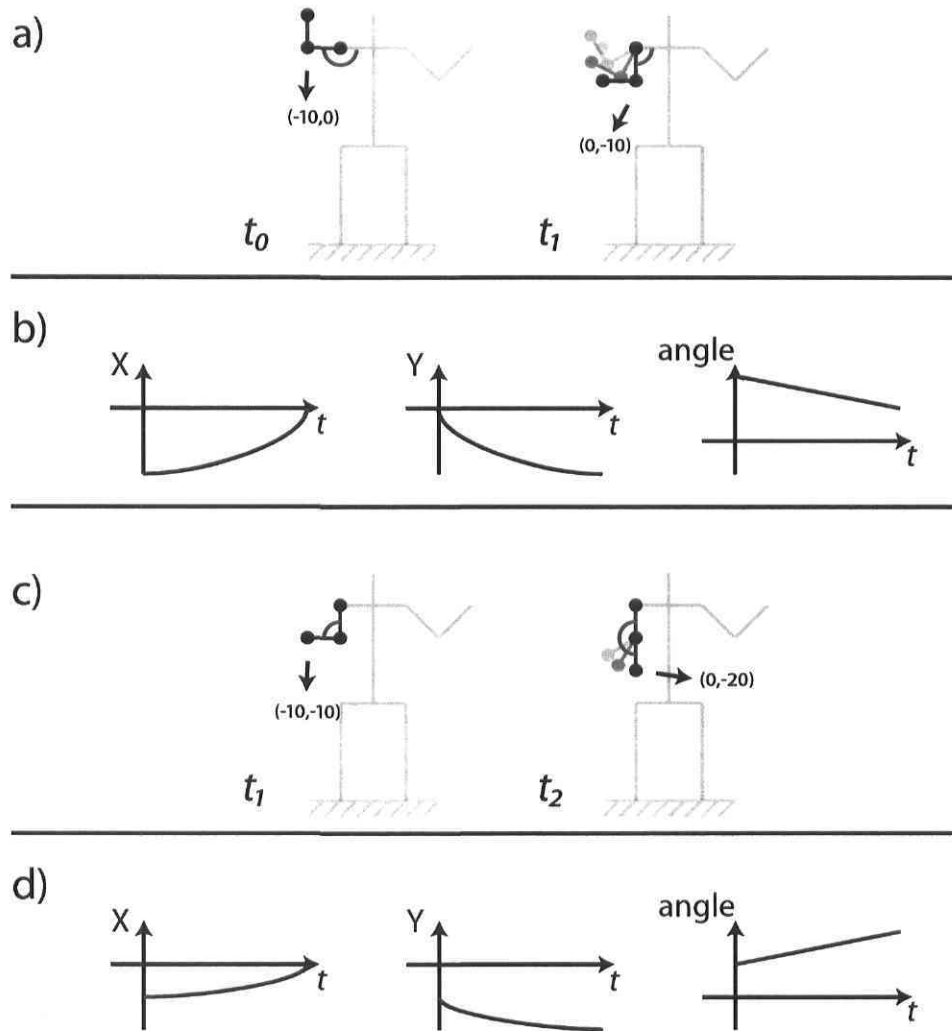


FIG. 4.3 – Illustration des signaux utilisés pour caractériser les flexions et les extensions d'une articulation. a) Flexion de l'épaule. b) Approximation des signaux extraits de l'articulation de l'épaule pour le mouvement a). c) Extension du coude. d) Approximation des signaux extraits de l'articulation du coude pour le mouvement c).

se produire à la fois pour une partie du corps donnée. Il est en effet possible d'imaginer quelqu'un qui marche et qui salue en même temps. Il est toutefois beaucoup moins probable d'effectuer un mouvement cyclique au niveau du coude et un autre mouvement cyclique, indépendant du premier, au niveau de l'épaule du même bras. Puisque nous travaillons dans un contexte de mouvements naturels, cette hypothèse ne pose pas de limites supplémentaires sur les activités pouvant être segmentées.

Suivant cette hypothèse, nous regroupons dans un ensemble tous les signaux appartenant à une même partie du corps. Cela nous accorde une certaine redondance puisque le mouvement cyclique d'un membre risque fort d'être perceptible sur plus d'un signal de l'ensemble correspondant. Comme nous le verrons, cette redondance permet d'améliorer la qualité des segmentations en diminuant le nombre de fausses détections.

Une fois l'ensemble de signaux formé, la segmentation consiste simplement à segmenter indépendamment chaque signal à l'aide des méthodes présentées au chapitre précédent. Il ne reste ensuite qu'à regrouper les segmentations individuelles pour en former une seule segmentation s'appliquant à cet ensemble de signaux.

## 4.4 Fusion des segmentations

L'étape de la fusion des segmentations est un autre point critique dans notre approche. À cette étape, nous disposons d'une segmentation pour chaque signal de chaque ensemble de signaux. Pour un même ensemble de signaux, nous allons considérer que tous les segments de toutes les segmentations associées à ces signaux forment l'ensemble des segments candidats. En effet, chacun de ces segments représente potentiellement une activité cyclique effectuée par une partie du squelette. Il faut donc tenter de trouver automatiquement quels segments parmi les candidats représentent le mieux ces activités.

Soit un ensemble de signaux dont l'ensemble des segments candidats comporte  $N$  candidats distincts. Le problème posé consiste donc à choisir un sous-ensemble de segments candidats distincts. La seule contrainte connue sur ce sous-ensemble est que son cardinal doit être compris entre 0 et  $N$ . Sous ces paramètres, il y a donc  $\sum_{i=0}^N \binom{N}{i}$  solutions possibles. Ce grand nombre nous a donc poussés à utiliser une fois de plus une approche *glouton* pour résoudre ce problème. Nous avons développé deux algorithmes différents à cette fin.

### 4.4.1 Algorithme de fusion simple

Cet algorithme procède en choisissant un segment candidat prometteur, puis en éliminant de l'ensemble des candidats les segments ayant un recoupement temporel avec le segment choisi. Il répète cette procédure tant qu'il reste des segments dans l'ensemble des candidats. À chaque itération, la sélection du meilleur segment parmi les candidats se fait à l'aide d'un indice de périodicité global, qui indique à quel point le segment se généralise à l'ensemble de signaux. En d'autres termes, cet indice indique dans quelle mesure le segment isole une partie cyclique dans chacun des signaux de l'ensemble. À chaque itération, l'algorithme choisira donc le segment candidat ayant l'indice de périodicité global le plus élevé. L'algorithme 4.1 décrit cette procédure en pseudo-code pour un ensemble de signaux  $\mathbb{S}$  et un ensemble ordonné de segments candidats  $\mathbb{C}$ .

---

#### Algorithme 4.1 *FusionSimple*( $\mathbb{S}, \mathbb{C}$ )

---

```

1:  $\text{SEG} \leftarrow \emptyset$ 
2: tant que  $\mathbb{C} \neq \emptyset$  faire
3:    $(I, J) \leftarrow \text{argmax}_{c_k \in \mathbb{C}} (\text{IndiceGlobal}(c_k, \mathbb{S}))$ 
4:    $\text{SEG} \leftarrow \text{SEG} \cup \{(I, J)\}$ 
5:   pour  $k \leftarrow |\mathbb{C}|$  à 1 faire
6:      $(i, j) \leftarrow c_k$ 
7:     si  $i \geq J \wedge j \leq I$  alors
8:        $\mathbb{C} \leftarrow \mathbb{C} / \{c_k\}$ 
9:     fin si
10:  fin pour
11: fin tant que
12: retourner  $\text{SEG}$ 

```

---

Cet algorithme a l'avantage d'être simple et rapide. Toutefois, la qualité de ses solutions dépend entièrement de l'indice de périodicité global. Si celui-ci permet de bien discerner les segments réellement intéressants des candidats moins intéressants, la segmentation obtenue sera généralement adéquate.

Pour calculer l'indice de périodicité global d'un segment sur un ensemble de signaux, nous faisons appel à l'indice de périodicité  $\Psi'_S$  défini au chapitre précédent. La procédure consiste à calculer la valeur de  $\Psi'_S$  pour ce segment sur chaque signal de l'ensemble. L'indice de périodicité global est alors calculé comme la somme des  $\Psi'_S$  supérieurs à un seuil bas  $\eta_B$ . Le seuil  $\eta_B$  est inférieur au seuil  $\eta_E$  utilisé pour la segmentation des signaux puisque nous considérons qu'il n'y aura généralement plus de fausses détections à filtrer. Ce seuil doit toutefois être conservé pour assurer que la somme de plusieurs indices faibles ne puisse donner un indice global intéressant. L'algorithme 4.2 décrit cette

procédure en pseudo-code. L'idée derrière cette approche est de favoriser un segment qui isole des parties cycliques sur plusieurs signaux de l'ensemble tout en proposant un certain compromis entre le nombre de tels signaux et la périodicité des parties isolées.

---

**Algorithme 4.2**  $IndiceGlobal((i, j), \mathbb{S} = \{S_1, S_2, \dots, S_n\})$

---

```

1: indice ← 0
2: pour  $k \leftarrow 1$  à  $n$  faire
3:   si  $\Psi'_{S_{k[i,j]}} \geq \eta_B$  alors
4:      $indice \leftarrow indice + \Psi'_{S_{k[i,j]}}$ 
5:   fin si
6: fin pour
7: retourner indice

```

---

#### 4.4.2 Algorithme de fusion par différence

L'algorithme de fusion simple comporte toutefois une faille majeure : il élimine invariablement tout segment candidat chevauchant les segments choisis, même si le chevauchement est négligeable. Dans le cas de séquences comportant plusieurs activités cycliques temporellement adjacentes, ceci peut se traduire par une détection manquée. À titre d'exemple, supposons qu'une séquence comporte une activité cyclique du temps  $t_0$  au temps  $t_1$  et une deuxième activité cyclique du temps  $t_1 + 1$  au temps  $t_2$ . Supposons que lors de la segmentation, nous avons obtenu un premier segment  $[t_0, t_1]$  isolant parfaitement la première activité et un second segment  $[t_1, t_2]$  précédant le début de la seconde activité d'une seule trame. Peu importe quel segment est choisi en premier, l'algorithme 4.1 n'admettra jamais les deux segments candidats dans la segmentation finale, ce qui implique forcément une détection manquée pour une des deux activités<sup>4</sup>.

Afin de corriger cette faille, nous proposons un algorithme de fusion alternatif. Celui-ci s'apparente beaucoup à l'algorithme 4.1, mais en diffère lorsque vient le temps d'éliminer les candidats ayant un chevauchement temporel avec le segment choisi. Dans ce nouvel algorithme, si un candidat connaît un chevauchement temporel avec le segment choisi, seule la partie en chevauchement du candidat est éliminée. La partie restante ou les parties restantes sont conservées dans l'ensemble des candidats si le segment qu'elle forme obtient un indice de périodicité  $\Psi'_S$  supérieur au seuil élevé  $\eta_E$  sur au moins un signal de l'ensemble. Autrement, le reste du candidat est également éliminé. Ceci nous permet justement d'éviter la situation présentée ci-dessus tout en s'assurant, avec

---

<sup>4</sup>Notons que cette situation est propre au cas où des activités cycliques sont temporellement adjacentes.

le seuil  $\eta_E$ , de ne pas introduire de nouveaux candidats pouvant mener à une fausse détection. L'algorithme 4.3 présente cette procédure sous forme de pseudo-code.

Lorsqu'observé rapidement, l'algorithme 4.3 peut sembler beaucoup plus complexe que l'algorithme 4.1. En fait, les lignes numérotées de 9 à 19 contiennent le pseudo-code servant à identifier si le segment candidat comporte une partie précédant le début du segment retenu (ligne 9), à valider si le segment formé entre la trame de début du segment candidat et la trame de début du segment retenu obtient un indice de périodicité assez élevé sur au moins un des signaux de l'ensemble (lignes 10 à 15) et à l'ajouter à l'ensemble des segments candidats si c'est le cas (lignes 16 à 18). Les lignes numérotées 20 à 30 contiennent un pseudo-code identique, mais pour traiter la partie d'un segment candidat se trouvant après la trame de fin du segment retenu.

Cet algorithme ne comporte essentiellement pas de désavantages par rapport à l'algorithme de fusion simple. Il sera donc généralement préféré à ce dernier. Il peut toutefois arriver qu'un segment candidat introduisant une fausse détection soit éliminé par l'algorithme de fusion simple, alors que la fusion par différence en conservera la majeure partie. Dans cette situation, l'algorithme de fusion simple pourrait sembler préférable à la fusion par différence, en ce sens que la segmentation finale est plus adéquate, mais ce comportement ne se produit que très rarement et tient davantage de la chance que d'un véritable avantage inhérent à l'algorithme de fusion simple. Le seul véritable avantage de l'algorithme de fusion simple et qu'il est légèrement plus rapide d'exécution que l'algorithme de fusion par différence. Toutefois, ce gain de vitesse est généralement négligeable lorsque nous considérons le temps de calcul nécessaire à la segmentation de tous les signaux de l'ensemble.

## 4.5 Analyse au niveau de l'anatomie entière

Une fois la segmentation obtenue pour chaque ensemble de signaux, nous pouvons procéder à une analyse de plus haut niveau des activités. Puisque notre approche se veut générique, nous ne possédons pas de modèles d'activités qui puissent nous permettre d'identifier les activités cycliques à ce stade. Nous pouvons toutefois faire une analyse des segmentations temporelles pour en faciliter l'interprétation.

À ce stade, nous nous intéressons surtout à savoir si les activités segmentées indépendamment sur chaque partie du corps sont en fait des activités distinctes ou encore s'ils font partie d'une même activité cyclique impliquant plusieurs parties du corps. Sans modèles d'activités, il est impossible d'affirmer avec certitude si deux activités

---

**Algorithme 4.3** *FusionDifference*( $\mathbb{S} = \{S_1, S_2, \dots, S_n\}, \mathbb{C}$ )
 

---

```

1:  $\text{SEG} \leftarrow \emptyset$ 
2: tant que  $\mathbb{C} \neq \emptyset$  faire
3:    $(I, J) \leftarrow \text{argmax}_{c_k \in \mathbb{C}} (\text{IndiceGlobal}(c_k, \mathbb{S}))$ 
4:    $\text{SEG} \leftarrow \text{SEG} \cup \{(I, J)\}$ 
5:   pour  $k \leftarrow |\mathbb{C}|$  à 1 faire
6:      $(i, j) \leftarrow c_k$ 
7:     si  $i \geq J \wedge j \leq I$  alors
8:        $\mathbb{C} \leftarrow \mathbb{C} / \{c_k\}$ 
9:       si  $i < I$  alors
10:         $flag \leftarrow 0$ 
11:        pour  $l \leftarrow 1$  à  $n$  faire
12:          si  $\Psi'_{S_{[i:I-1]}} \geq \eta_E$  alors
13:             $flag \leftarrow 1$ 
14:          fin si
15:        fin pour
16:        si  $flag = 1$  alors
17:           $\mathbb{C} \leftarrow \mathbb{C} \cup \{(i : I - 1)\}$ 
18:        fin si
19:      fin si
20:      si  $j > J$  alors
21:         $flag \leftarrow 0$ 
22:        pour  $l \leftarrow 1$  à  $n$  faire
23:          si  $\Psi'_{S_{[J+1:j]}} \geq \eta_E$  alors
24:             $flag \leftarrow 1$ 
25:          fin si
26:        fin pour
27:        si  $flag = 1$  alors
28:           $\mathbb{C} \leftarrow \mathbb{C} \cup \{(J + 1 : j)\}$ 
29:        fin si
30:      fin si
31:    fin si
32:  fin pour
33: fin tant que
34: retourner  $\text{SEG}$ 

```

---



se produisant en même temps sur deux parties du corps sont deux activités distinctes ou si elles composent une seule activité. Nous pouvons toutefois utiliser l'information fréquentielle des signaux pour regrouper les activités de fréquences similaires, celles-ci étant plus propices à former une seule activité cyclique.

La fréquence d'une activité cyclique est définie comme la fréquence du segment  $[i, j]$  sur le signal  $S$  de l'ensemble de signaux ayant l'indice de périodicité  $\Psi'_{S[i,j]}$  le plus élevé. Pour déterminer si deux activités cycliques sont de fréquences appariées, nous observons si une fréquence est un multiple ou un facteur entier de l'autre pour une certaine tolérance d'erreur  $\iota$ . Pour deux fréquences  $fr_A$  et  $fr_B$ , l'algorithme 4.4 retourne 0 si elles ne sont pas appariées et 1 si elles le sont. La fonction  $round(a)$  arrondi le nombre réel  $a$  à l'entier le plus près.

---

**Algorithme 4.4** *FrequenciesAppariees*( $fr_A, fr_B$ )
 

---

```

1: si  $fr_A < fr_B$  alors
2:   si  $|\frac{fr_B}{fr_A} - round(\frac{fr_B}{fr_A})| < \iota$  alors
3:     retourner 1
4:   fin si
5:   retourner 0
6: fin si
7: si  $|\frac{fr_A}{fr_B} - round(\frac{fr_A}{fr_B})| < \iota$  alors
8:   retourner 1
9: fin si
10: retourner 0

```

---

## 4.6 Alternative à la segmentation anatomique

Lorsque la segmentation s'avère plus difficile, à cause de bruit dans les signaux par exemple, ou lorsqu'une plus grande confiance dans la qualité de la solution est désirée, il est possible d'ignorer l'étape de la segmentation anatomique. Tous les signaux sont alors considérés comme appartenant à un seul ensemble de signaux.

Un premier effet de cette variation de l'approche est qu'il ne sera désormais plus possible de détecter plusieurs activités cycliques se produisant simultanément sur différentes parties du corps. Lorsque nous désirons seulement connaître à quels moments le squelette effectue des activités cycliques, cet effet est négligeable.

Un deuxième effet, positif cette fois, est le niveau de redondance beaucoup plus élevé dans cette variation de l'approche. Ainsi, une activité cyclique effectuée par le

squelette risque de se transposer sur un nombre plus grand de signaux. Ceci implique que l'algorithme de segmentation de signaux sera plus propice à segmenter toutes les activités cycliques. De plus, l'algorithme de fusion risque de faire face à un plus grand nombre de candidats pour chaque activité cyclique, ce qui implique que les chances sont bonnes d'obtenir une segmentation plus précise de chaque activité.

# Chapitre 5

## Étude empirique

Dans ce chapitre, nous présentons les expériences effectuées pour apprécier la rapidité d'exécution et l'exactitude des divers algorithmes proposés dans les chapitres précédents. En quelque sorte, ce chapitre présente la validation pratique de la théorie présentée dans cet ouvrage.

Les expériences y sont décrites de façon détaillée. Les topologies squelettiques utilisées sont d'abord présentées, suivies d'une description sommaire des séquences et des métriques d'évaluation utilisées. Le chapitre enchaîne finalement avec nombre de comparaisons entre les divers algorithmes présentés dans cet ouvrage.

### 5.1 Topologies squelettiques utilisées

Notons d'abord que les topologies utilisées sont fortement liées aux méthodes utilisées pour extraire automatiquement une séquence de ces squelettes à partir d'une séquence vidéo. Nous avons utilisé deux approches différentes pour extraire de telles séquences et, par conséquent, nous avons traité deux topologies squelettiques différentes.

La première méthode d'extraction de séquences de squelettes se base sur les travaux de Vignola [14]. La topologie qui en est issue est présentée par la figure 5.1.

Cette méthode fut développée pour l'extraction d'un squelette pour une seule image. Afin d'en obtenir une séquence de squelettes, nous avons simplement appliqué la méthode consécutivement sur toutes les trames de la séquence vidéo. Puisque cette mé-

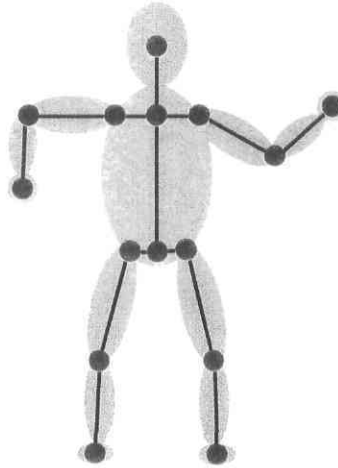


FIG. 5.1 – Topologie squelettique proposée dans [14]. Une silhouette grise est placée en arrière-plan en référence.

thode n'utilise pas l'information temporelle de la séquence, telle que le squelette extrait pour la trame précédente, les séquences de squelettes ainsi obtenues comportent typiquement une quantité significative de bruit. Cela se traduit par une sorte de tremblement du squelette tout au long de la séquence. Bien que cela puisse nuire au bon fonctionnement de nos algorithmes, nous avons décidé de conserver les séquences de squelettes ainsi formées afin de confronter nos algorithmes à des séquences imparfaites.

Un autre aspect particulier de cette approche est qu'elle se base sur l'hypothèse que la personne sur l'image se trouve toujours face à la caméra, que chaque membre se trouve dans son quadrant respectif et que la tête est toujours le point le plus élevé de la silhouette. Afin de respecter cette hypothèse, les activités cycliques dans les séquences traitées ont été conformées à ces spécifications. Bien que ceci nous ait limités dans le choix de nos activités, surtout en réduisant le nombre d'activités naturelles possibles, cette approche nous permet tout de même d'obtenir rapidement un grand nombre de séquences de squelettes sans équipements autres qu'une caméra vidéo et un ordinateur. De plus, la topologie utilisée par cette méthode présente un niveau de détail suffisamment élevé pour mettre en valeur notre approche de segmentation anatomique.

La seconde méthode d'extraction de squelettes est issue des travaux de suivi de membres présentés par Jean et coll. dans [6]. La topologie ainsi dérivée est présentée par la figure 5.2.

Cette méthode offre des avantages importants par rapport à la précédente. Puisqu'elle utilise l'information de mouvement dans la séquence, la séquence de squelettes

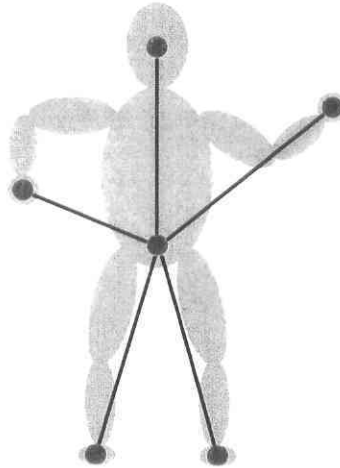


FIG. 5.2 – Topologie squelettique issue de l’approche de suivi des membres proposée dans [6]. Une silhouette grise est placée en arrière-plan en référence.

obtenue ne comporte généralement qu’une quantité de bruit négligeable. Ceci nous permet d’apprécier les capacités des algorithmes proposés sur des séquences s’approchant de séquences idéales.

Puisqu’elle se base sur le suivi des mains, des pieds et de la tête, cette méthode permet d’obtenir des squelettes représentant des mouvements naturels, tels que la marche, où les membres se croisent ou changent de quadrant. Le type d’activité n’y est donc pas limité. Cet aspect est très important considérant que la majorité des recherches en analyse de mouvements s’attardent surtout à des mouvements humains naturels.

Un dernier aspect de cette méthode est la simplicité de la topologie utilisée. Puisque l’approche fut développée d’abord pour effectuer le suivi des extrémités des membres, elle n’inclue pas explicitement les informations de mouvement des articulations. Ceci vient poser une certaine difficulté pour nos algorithmes, puisque les mouvements perceptibles sur la séquence de squelettes ne correspondent pas forcément à ceux qui sont perceptibles dans la séquence vidéo d’origine. À titre d’exemple, un croisement de membre ne sera jamais représenté par un squelette issu de cette méthode. Toutefois, dans les séquences saisies, toutes les activités cycliques effectuées dans les séquences d’origine se traduisent par des mouvements cycliques dans la séquence de squelettes, ce qui remplit l’unique condition nécessaire à la segmentation temporelle de ces activités. La simplicité de la topologie squelettique implique également un nombre réduit de signaux pour représenter le mouvement, ce qui implique une redondance moins forte et, donc, un risque plus élevé de détections manquées.

## 5.2 Séquences de squelettes utilisées

Chacun de nos tests empiriques est effectué sur un bon nombre de séquences différentes. De cette façon, il est plus facile de juger si des observations semblent bien indiquer un comportement général de nos algorithmes ou s'il ne s'agit que d'un comportement marginal sur quelques séquences. Étant donné le temps nécessaire à la saisie et au traitement de ces séquences, ainsi que l'espace de stockage que cela implique, nous avons dû nous en tenir à près d'une vingtaine de séquences différentes.

Les tableaux 5.1 et 5.2 listent les séquences vidéo saisies pour nos tests en indiquant leur durée respective, le nombre d'activités cycliques qu'elles comportent et une courte description. Le tableau 5.1 liste les séquences respectant les contraintes de mouvements posées par l'algorithme d'ajustement de squelette présenté dans [14] alors que le tableau 5.2 liste les séquences comportant des mouvements naturels tels que la marche. Ces séquences ont toutes été saisies devant un décor fixe, à une vitesse de 30 trames par seconde et à une résolution de 640x480.

Le nom de chaque séquence vidéo est composé d'une lettre majuscule suivie d'un chiffre ou d'une description abrégée. Alors que le chiffre ou la description abrégée ne sert qu'à assurer l'unicité de l'identifiant de la séquence, la lettre est soit *A*, *B* ou *C* et réfère aux groupements mentionnés dans la section 1.2. Pour récapituler rapidement, indiquons simplement que les séquences de type *A* comportent des pauses entre chacune des activités cycliques contenues. Les séquences de type *B*, quant à elles, comportent au moins deux d'activités cycliques temporellement adjacentes. Finalement, les séquences de type *C* comportent au moins deux activités cycliques temporellement adjacentes partageant un même mouvement cyclique. La figure 5.3 illustre chaque type de séquence à l'aide d'un exemple.

Le nom de chaque séquence de squelettes, quant à lui, est formé du nom de la séquence vidéo dont elle est issue précédée du préfixe *Vign* ou *Jean*. Le préfixe *Vign* indique que la séquence de squelettes a été formée à partir de l'algorithme proposé par Vignola dans [14] alors que le préfixe *Jean* indique qu'elle a été formée à partir de la méthode proposée par Jean et coll. dans [6]. Notons que les séquences vidéo du tableau 5.2 ne sont utilisées qu'avec cette dernière méthode puisqu'elles contiennent des activités cycliques naturelles ne pouvant être adéquatement représentées à l'aide de la méthode proposée par Vignola.

Il est aussi important de noter qu'il est difficile de garantir *a priori* qu'une séquence vidéo respecte la condition de cycle simple mentionnée au chapitre 1. Ceci est dû au

Séquence	Trames	Activités	Description
A1	370	2	Battements de bras, jumping jacks
A2	775	4	Battements de bras, battements une jambe et un bras, battements d'un bras, jumping jacks
A3	400	3	Battements de bras, battements de bras, battements de bras
A4	675	3	Battements jambe gauche, battements jambe droite, battements de bras
A5	450	2	Sauts sur place, sauts sur place
B1	1170	5	Battements une jambe et un bras, battements bras droit, battements de bras, battements une jambe et un bras, saluts sumo
B2	350	2	Pas chassés, battements de bras
B3	620	4	Battements de bras, sauts sur place, battements de bras, battements jambe droite
B4	555	3	Jumping jacks, sauts sur place, pas chassés
B5	625	4	Jumping jacks, battements jambe droite, battements de bras, sauts sur place
C1	380	2	Battements de bras, jumping jacks
C2	380	5	Battements bras droit, battements de bras, battements bras droit, pas chassés, jumping jacks
C3	450	3	Pas chassés, battements de bras, battements des deux bras et jambe droite

TAB. 5.1 – Sommaire des séquences vidéo saisies - mouvements contraints

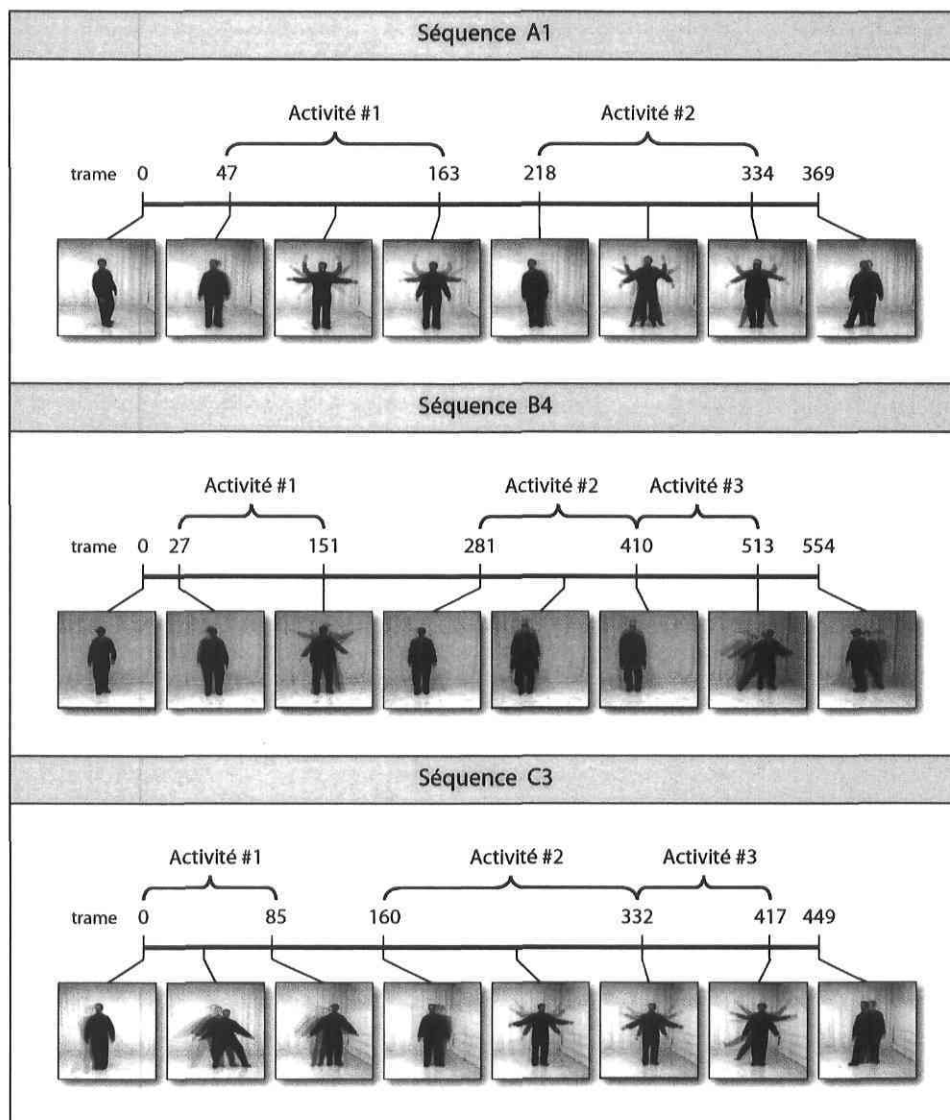


FIG. 5.3 – Exemples de séquences vidéo pour chaque type de séquence. Pour chaque trame clé montrée, quelques trames précédentes ont été superposées avec un niveau de transparence pour aider à percevoir le mouvement effectué.



Séquence	Trames	Activités	Description
ATwoDiag	460	2	Marche en diagonale par rapport au plan image de la caméra, virage à 180°, marche de retour
ATwoSide	360	2	Marche vers la droite, virage à 180°, marche vers la gauche
ASitting	510	2	Marche vers la droite, repos, marche vers la gauche
BFourDir	810	4	Marches successives dans quatre directions : droite - arrière - gauche - avant
BLimpWalk	260	2	Sautillement immédiatement suivi d'une marche normale

TAB. 5.2 – Sommaire des séquences vidéo saisies - mouvements naturels

fait que, par notre approche, les contraintes ne s'appliquent pas directement aux mouvements, mais plutôt aux signaux représentant ce mouvement. Puisque les méthodes d'extraction de squelettes utilisées ne garantissent pas une correspondance anatomique adéquate avec la pose de l'individu dans la séquence vidéo d'origine, il est possible qu'un mouvement cyclique simple soit représenté par des signaux cycliques non simples. Ce type d'erreurs est toutefois rarement observé dans nos séquences de test.

### 5.3 Métriques d'évaluation

Tel qu'expliqué dans la section 1.2, les métriques d'évaluation que nous avons retenues sont la précision, le rappel et l'indice de similarité  $I_S$ . Ces métriques sont calculées à partir de la segmentation formée et d'une segmentation de référence. Rappelons que les deux premières métriques ne s'attardent qu'aux détections d'activités cycliques, à savoir si le contenu d'un segment formé est cyclique, alors que la troisième s'attarde à l'exactitude des segments formés, à savoir si les segments de la segmentation de référence sont reproduits fidèlement dans la segmentation formée.

Ces métriques sont présentées plus en détail dans la section 1.2.

Param.	Algorithmes	Description
$\beta$	3.2; 3.3; 3.4; 3.7	Longueur minimale d'une activité cyclique pouvant être détectée (en nombre de trames). Utilisé pour éviter d'évaluer inutilement des segments trop courts.
$\beta_P$	3.6	Longueur minimale d'une pause à soustraire (en nombre de trames).
$\eta_E$	3.7; 4.3	Seuil haut sur la périodicité d'un segment de signal. Utilisé pour filtrer les fausses détections.
$\eta_B$	$\mathbf{Y}_S$ ; 4.2	Seuil bas sur la périodicité d'un segment de signal. Utilisé pour déterminer si un segment est considéré cyclique ou non.
$\Delta_U$	3.3	Pas d'échantillonnage uniforme.
$D$	3.4	Degré du pas d'échantillonnage initial. Le pas initial se calcul $\Delta_S = 2^D$ .
$\gamma$	3.4	Constante de sélection. À chaque itération, détermine quels segments doivent être évalués.
$N$	3.5	Nombre d'itérations de la boucle d'optimisation locale à effectuer.
$\epsilon$	3.6	Faible pourcentage de l'amplitude d'un signal. Utilisé pour déterminer si un segment de signal peut être considéré comme une pause.

TAB. 5.3 – Sommaire des paramètres utilisés par les algorithmes présentés

## 5.4 Choix des paramètres

Avant de procéder à l'étude de la performance de nos algorithmes, il convient de trouver les meilleures valeurs de paramètres possibles pour nos algorithmes. Nous sommes en effet d'avis que la solution que nous présentons sera plus pratique si elle ne demande l'ajustement d'aucun paramètre.

Nous chercherons donc à illustrer qu'il est possible d'isoler des valeurs de paramètres fixes produisant de bons résultats sur l'ensemble de nos séquences de test. Ce faisant, nous allons également trouver de telles valeurs pour tous les paramètres utilisés par nos algorithmes. Le tableau 5.3 présente la liste des paramètres mentionnés dans cet ouvrage. Pour chaque paramètre, il indique quels algorithmes y font appel et en donne une brève description.

Tel que nous l'avons mentionné plus tôt, nous ne considérons pas les activités cy-

cliques de deux cycles ou moins. En effet, nos algorithmes étant prévus pour traiter des signaux comportant du bruit et des cycles modulés, il est fréquent qu'une partie de signal soit considérée comme deux cycles suffisamment ressemblants alors qu'elle ne correspond qu'à du bruit ou à un mouvement non cyclique. En ne considérant que les activités cycliques de trois cycles ou plus, la probabilité de ce genre de formation aléatoire devient négligeable.

De plus, puisque nous travaillons à la segmentation de mouvements humains, nous pouvons poser l'hypothèse réaliste qu'aucune activité cyclique n'atteindra une fréquence supérieure à 5Hz.

Ces deux observations nous permettent de déterminer la durée minimum des activités à segmenter. Dans le cas limite, l'activité cyclique comportera exactement 3 cycles effectués à une fréquence de 5Hz, ce qui pose une borne inférieure de 0.6 seconde sur la durée d'une activité cyclique. Sur des séquences acquises à 30 trames par seconde, cela se traduit par une limite inférieure de 18 trames.

Cela nous permet donc de fixer théoriquement la valeur du paramètre  $\beta$ . Les autres paramètres devront toutefois être fixés de façon empirique. Étant donné le nombre de paramètres non fixés à cette étape, il serait lourd de chercher à tous les optimiser simultanément. Nous posons donc l'hypothèse que ces paramètres sont indépendants et procédons en fixant les paramètres d'un seul algorithme à la fois. Cette hypothèse n'est probablement pas exacte, mais nous permettra tout de même d'obtenir de bons résultats.

Nous devons donc attribuer une valeur temporaire à tous les paramètres n'étant pas fixés à cette étape. Ces valeurs ont été choisies puisqu'elles fournissaient des résultats satisfaisants lors de tests préliminaires. Le tableau 5.4 liste les valeurs attribuées aux différents paramètres à cette étape. Nous sommes conscients que le choix de ces valeurs temporaires est subjectif, mais puisque ces valeurs seront fixées empiriquement à l'aide d'un nombre restreint de séquences, le choix des valeurs finales des paramètres sera inévitablement empreint de subjectivité.

### 5.4.1 Paramètres de soustraction des pauses

Les premiers paramètres que nous désirons fixer sont ceux utilisés par la soustraction de pauses. Rappelons-nous que cet algorithme a pour unique objectif de réduire le temps de calcul nécessaire à la segmentation automatique d'une séquence de squelettes. Il ne

Param.	Valeur
$\beta$	18
$\beta_P$	<i>15</i>
$\eta_E$	<i>80%</i>
$\eta_B$	<i>50%</i>
$\Delta_U$	4
$D$	3
$\gamma$	<i>95%</i>
$N$	<i>100</i>
$\epsilon$	<i>3%</i>

TAB. 5.4 – Liste des paramètres avec leur valeur respective. Les valeurs temporaires sont indiquées en italique.

devrait donc pas contribuer à améliorer la qualité de la segmentation obtenue<sup>1</sup>, mais peut par contre y nuire. En effet, une valeur trop faible pour  $\beta_P$  risque de détecter des pauses en plein milieu de mouvements cycliques et ainsi empêcher la segmentation de ces mouvements en les scindant en plusieurs parties. De la même façon, une valeur de  $\epsilon$  trop grande pourrait considérer des mouvements d'ampleur significative comme faisant partie de silences. La combinaison de paramètres recherchée ici est donc celle qui permettra de réduire le plus possible le temps de calcul sans toutefois nuire à la qualité des segmentations obtenues.

Afin de trouver les valeurs idéales<sup>2</sup> pour  $\beta_P$  et  $\epsilon$ , nous avons confronté différentes valeurs de ces paramètres à une évaluation empirique. Cette évaluation consistait à appliquer notre approche à l'ensemble de nos séquences tests en utilisant l'approche exhaustive (algorithme 3.2) pour la segmentation de signaux. Cet algorithme a été préféré aux autres étant donné qu'il offre théoriquement toujours la meilleure segmentation possible par notre approche. Pour chaque combinaison de valeurs de paramètres, l'évaluation a été faite sur toutes les séquences à l'aide de l'algorithme de fusion simple (algorithme 4.1) et une seconde fois à l'aide de l'algorithme de fusion par différence (algorithme 4.3).

Chaque combinaison de valeurs ( $\beta_P = x, \epsilon = y$ ) a été évaluée deux fois sur chacune des séquences de squelettes de test, soit une fois par algorithme de fusion utilisé. Notons que nous avons un total de 31 séquences de test : les 13 séquences vidéo de la table 5.1 traitées avec l'approche proposée par Vignola ; ces 13 mêmes séquences traitées avec

<sup>1</sup>En pratique, il peut arriver que la soustraction d'une pause prévienne une fausse détection en soustrayant une partie de signal qui semble cyclique, mais qui est de trop faible amplitude

<sup>2</sup>Les valeurs idéales sont ici les valeurs offrant les meilleurs résultats moyens sur l'ensemble de nos séquences test.

$\beta_P \setminus \epsilon$	0%	1%	2%	3%	5%	7%	10%	15%
<b>4</b>	0.71	0.60	0.42	0.30	0.16	0.00	0.00	0.00
<b>6</b>	0.71	0.70	0.61	0.53	0.32	0.23	0.06	0.01
<b>8</b>	0.71	0.69	0.68	0.60	0.53	0.45	0.28	0.16
<b>10</b>	0.71	0.72	0.71	0.68	0.61	0.55	0.44	0.28
<b>15</b>	0.71	0.70	0.70	0.72	0.67	0.66	0.59	0.55
<b>20</b>	0.71	0.69	0.70	0.72	0.69	0.68	0.67	0.61

TAB. 5.5 – Indice  $I_S$  moyen obtenu pour l'ensemble des tests pour différentes valeurs des paramètres  $\beta_P$  et  $\epsilon$ .

l'approche dérivée de la proposition de Jean et coll. ; et, finalement, les cinq séquences de la table 5.2 traitées avec cette même approche. Tel que discuté plus tôt, la topologie squelettique issue de la méthode proposée dans [6] n'est pas représentative des mouvements effectués. Comme nous le verrons plus tard, cela se traduit par une faible performance de notre approche sur les séquences de squelettes issus de cette méthode lorsque nous utilisons la segmentation anatomique. Pour les tests présentés dans cette section, nous avons donc omis la segmentation anatomique. Chaque combinaison de valeurs ( $\beta_P = x, \epsilon = y$ ) a donc été évaluée 62 fois dans des conditions différentes.

Le tableau 5.5 présente l'indice de similarité  $I_S$  moyen obtenu lors des tests pour chaque paire de valeurs de paramètres. Le tableau 5.6 présente le temps total pris pour toutes ces évaluations<sup>3</sup>. À titre de référence, la somme des durées des séquences de squelettes traitées est de 560 secondes, soit 9 minutes et 20 secondes. En multipliant par deux pour considérer les deux algorithmes de fusion utilisés, nous obtenons un total de 18 minutes et 40 secondes de séquences traitées. Certains des temps présentés dans le tableau 5.6 dépassent de beaucoup cette somme, mais cela est attendu, étant donné la forte complexité de calcul de l'approche de segmentation exhaustive. Ce tableau nous indique tout de même dans quelle mesure l'ajustement des paramètres peut réduire la complexité de calcul de notre approche.

En analysant les résultats présentés par les tableaux 5.5 et 5.6, nous observons que la combinaison ( $\beta_P = 10, \epsilon = 2\%$ ) semble offrir le meilleur compromis. En effet, parmi les combinaisons n'entraînant pas une perte de qualité significative de la solution<sup>4</sup>, il s'agit de la combinaison de valeurs offrant le temps de calcul cumulatif le plus faible. Nous pouvons aussi remarquer que le tableau 5.5 présente des valeurs supérieures à celle obtenue sans aucune segmentation de pauses. Ceci est attribuable au fait que la segmentation de pauses est susceptible de soustraire des segments de signaux immédiatement

<sup>3</sup>Toutes les expérimentations présentées dans ce chapitre ont été effectuées sur un processeur AMD Athlon 64bits 3200+ cadencé à 2.0GHz avec 512Mo de mémoire vive.

<sup>4</sup>Nous considérons qu'une perte de moins de 2% n'est pas significative.

$\beta_P \setminus \epsilon$	0%	1%	2%	3%	5%	7%	10%	15%
4	27h50m	0h15m	0h04m	0h01m	0m14s	0m02s	0m02s	0h01s
6	27h50m	0h51m	0h15m	0h08m	0h03m	0m42s	0m06s	0m02s
8	27h50m	2h09m	0h37m	0h19m	0h07m	0h03m	0h01m	0m06s
10	27h50m	3h18m	1h38m	0h46m	0h19m	0h08m	0h04m	0h01m
15	27h50m	8h44m	3h23m	2h18m	1h15m	0h35m	0h19m	0h09m
20	27h50m	13h52m	7h51m	3h40m	2h09m	1h23m	0h46m	0h22m

TAB. 5.6 – Temps cumulatif pris pour l’exécution de l’ensemble des tests pour différentes valeurs des paramètres  $\beta_P$  et  $\epsilon$ .

adjacents à une activité cyclique. Lorsque c’est le cas, l’algorithme de segmentation de signaux est plus propice à obtenir une segmentation précise, puisqu’il ne peut inclure dans un segment les trames appartenant à la pause qui vient d’être soustraite.

Si nous comparons avec les résultats obtenus sans la soustraction de pauses, qui sont représentés dans les tableaux par la colonne  $\epsilon = 0$ , nous remarquons que le calcul est effectué plus de 17 fois plus rapidement tout en conservant une qualité comparable. Cette observation vient donc valider l’idée de la soustraction des pauses pour simplifier la complexité de notre approche.

## 5.4.2 Seuils sur la périodicité

Les prochains paramètres que nous désirons fixer sont les seuils bas et élevé sur la périodicité. Puisque le rôle du seuil bas n’implique pas de filtrer les fausses détections, nous commençons par fixer la valeur du seuil élevé, de façon à minimiser le nombre de ces fausses détections, pour ensuite fixer le seuil bas.

Les tests effectués pour différentes valeurs de  $\eta_E$  sont identiques à ceux effectués plus tôt, mais en utilisant les valeurs identifiées précédemment par les paramètres  $\beta_P$  et  $\epsilon$ . Le tableau 5.7 présente l’indice de similarité  $I_S$  moyen obtenu lors des tests. Puisque l’objectif de ce seuil concerne uniquement la qualité de la solution, il n’est pas pertinent d’observer les temps de calcul relatifs à chaque valeur de  $\eta_E$ . Mentionnons d’ailleurs que la variation de la valeur de  $\eta_E$  n’affecte pas significativement le temps de calcul nécessaire aux tests présentés.

En étudiant la table 5.7, nous observons qu’une valeur de  $\eta_E = 85\%$  semble offrir des segmentations généralement de plus grandes qualités que les autres valeurs testées. Nous fixons donc le seuil de périodicité élevé à cette valeur. En fait, n’importe quelle

$\eta_E$	Indice $I_S$ moyen
100%	0.003
95%	0.541
90%	0.677
85%	0.718
80%	0.713
75%	0.696
70%	0.676
65%	0.672
60%	0.652
55%	0.650
50%	0.650

TAB. 5.7 – Indice  $I_S$  moyen obtenu pour l'ensemble des tests pour différentes valeurs du paramètre  $\eta_E$ .

valeur comprise entre 85% et 75% serait également acceptable. Les indices plus faibles pour les valeurs de  $\eta_E > 85\%$  s'expliquent par les détections manquées des activités cycliques ayant un indice de périodicité plus faible. Les indices plus faibles pour les valeurs de  $\eta_E < 75\%$  sont attribuables aux fausses détections allouées par un seuil trop bas. La marge entre l'indice de périodicité d'un segment cyclique et celui d'un segment non cyclique peut sembler très maigre, mais nous remarquons également que ces fausses détections n'augmentent que très tranquillement au fur et à mesure que la valeur de  $\eta_E$  descend. Ceci nous indique que même avec un seuil  $\eta_E = 50\%$ , le nombre de fausses détections demeure faible.

En ayant fixé le seuil élevé, nous pouvons alors tenter de fixer le seuil bas. Les résultats présentés à la table 5.8 sont issus de tests identiques à ceux effectués ci-dessus, mais en utilisant la nouvelle valeur du paramètre  $\eta_E$ . Ces résultats nous portent à choisir le seuil bas  $\eta_B = 50\%$  par rapport aux autres valeurs testées. Toutefois, toute valeur comprise entre 15% et 60% aurait également été acceptable.

### 5.4.3 Paramètres des algorithmes de segmentation

Les derniers paramètres restants à fixer sont les paramètres propres à chaque algorithme de segmentation. Puisqu'ils sont tous indépendants les uns des autres, l'ordre dans lequel nous allons les fixer n'importe pas. Notons que les tests dont les résultats sont présentés dans cette section sont similaires aux tests précédents, à la différence que la segmentation n'est pas effectuée par l'approche exhaustive, mais plutôt par l'al-

$\eta_B$	Indice $I_S$ moyen
85%	0.610
80%	0.619
75%	0.642
70%	0.649
65%	0.663
60%	0.703
55%	0.705
50%	0.718
45%	0.717
40%	0.717
35%	0.716
30%	0.712
25%	0.703
20%	0.711
15%	0.702
10%	0.692
5%	0.589
0%	0.415

TAB. 5.8 – Indice  $I_S$  moyen obtenu pour l'ensemble des tests pour différentes valeurs du paramètre  $\eta_B$ .



$\Delta_U$	Indice $I_S$ moyen	Temps cumulatif
1	0.718	1h25m
2	0.718	23m13s
3	0.699	10m17s
4	0.699	5m36s
5	0.710	3m40s
6	0.702	2m30s
8	0.699	1m28s
10	0.708	0m52s
15	0.650	0m16s
20	0.650	0m06s
30	0.639	0m04s

TAB. 5.9 – Indice  $I_S$  moyen et temps cumulatif pour l'ensemble des tests pour différentes valeurs du paramètre  $\Delta_U$ .

gorithme de segmentation utilisant le paramètre à fixer.

Commençons par fixer le pas d'échantillonnage dans l'algorithme d'échantillonnage uniforme (algorithme 3.3). L'ajustement de ce paramètre  $\Delta_U$  vise à diminuer le plus possible la complexité de calcul de la méthode tout en n'altérant pas significativement la qualité des solutions trouvées par rapport à celles trouvées par l'approche exhaustive. Nous nous attardons donc à la fois à la qualité des résultats ainsi qu'à la performance d'exécution. Le tableau 5.9 présente donc l'indice de similarité  $I_S$  moyen obtenu lors des tests, de même que le temps cumulatif nécessaire à l'exécution de ces tests.

Notons que, dans le cas de l'échantillonnage uniforme, un pas unitaire correspond exactement à l'approche de segmentation exhaustive. Les valeurs données pour  $\Delta_U = 1$  dans le tableau 5.9 peuvent donc être utilisées en référence. Par comparaison, parmi les valeurs étudiées, nous observons que toutes les valeurs entre 2 et 10 semblent offrir une qualité de segmentations comparables. Cela peut paraître surprenant *a priori* puisqu'un pas d'échantillonnage plus faible devrait permettre une exactitude plus grande. Toutefois, cette observation s'explique par le fait que l'indice de périodicité ne traduit pas exactement l'évaluation qu'un humain peut avoir de la périodicité d'une séquence. La trame de début ou de fin d'un segment, telle qu'identifiée dans notre segmentation de référence, peut donc se trouver à quelques trames de distance de celle identifiée par une valeur maximum de notre indice de périodicité. Ceci attribue donc une part d'erreur inévitable à l'approche exhaustive alors que l'échantillonnage, en fonction de la longueur de son pas, risque à l'occasion de tomber précisément sur les trames localisées par la segmentation de référence.

$D \setminus \gamma$	100%	95%	90%	85%	80%	70%	60%
<b>0</b>	0.718	0.718	0.718	0.718	0.718	0.718	0.718
<b>1</b>	0.718	0.718	0.718	0.718	0.718	0.718	0.718
<b>2</b>	0.718	0.718	0.718	0.718	0.718	0.718	0.718
<b>3</b>	0.718	0.718	0.718	0.718	0.718	0.718	0.718
<b>4</b>	0.718	0.718	0.718	0.718	0.718	0.718	0.718
<b>5</b>	0.712	0.718	0.718	0.718	0.718	0.718	0.718
<b>6</b>	0.706	0.718	0.718	0.718	0.718	0.718	0.718
<b>8</b>	0.699	0.699	0.699	0.699	0.699	0.699	0.699
<b>10</b>	0.707	0.707	0.707	0.707	0.707	0.707	0.707
<b>12</b>	0.707	0.707	0.707	0.707	0.707	0.707	0.707
<b>15</b>	0.707	0.707	0.707	0.707	0.707	0.707	0.707

TAB. 5.10 – Indice  $I_S$  moyen obtenu pour l'ensemble des tests pour différentes valeurs des paramètres  $D$  et  $\gamma$ .

Puisque l'objectif de cette approche est de minimiser le temps de calcul, nous fixons  $\Delta_U = 10$ . Cette valeur semble en effet suffisamment grande pour réduire très significativement le temps de calcul sans toutefois engendrer une perte significative d'exactitude dans la solution comparativement à l'approche exhaustive.

Les prochains paramètres auxquels nous nous attardons sont ceux utilisés par l'échantillonnage sélectif, soient le degré du pas initial  $D$  et la constante de sélection  $\gamma$ . Le tableau 5.10 présente l'indice de similarité moyen issu des tests effectués pour les différentes combinaisons de valeurs de paramètres, alors que le tableau 5.11 en présente les temps cumulatifs.

Avec l'échantillonnage sélectif, un degré  $D = 0$  correspond exactement à l'approche de segmentation exhaustive. En observant le tableau 5.10, on remarque qu'une valeur de  $\gamma < 95\%$  n'apporte aucun gain mesurable au niveau de la qualité des solutions par rapport à  $\gamma = 95\%$ , alors qu'une valeur de  $\gamma$  égale à 100% offre des résultats légèrement moins intéressants pour certaines valeurs de  $D$ . Bien que la différence de qualité entre les résultats obtenus pour ces deux valeurs de  $\gamma$  ne soit pas significative, nous opterons tout de même pour une valeur de 95%, qui offre une certaine sécurité théorique. En effet, un  $\gamma$  de 100% implique qu'à chaque itération, seuls les quelques points avoisinants le segment ayant obtenu le plus haut indice de périodicité sont évalués. Ceci entraîne donc une recherche très localisée dans l'espace de tous les segments possibles, ce qui peut être plus sensible aux maxima locaux. Nous fixerons donc le paramètre  $\gamma$  à la valeur de 95% dans nos expérimentations futures.

$D \setminus \gamma$	100%	95%	90%	85%	80%	70%	60%
0	1h25m	1h25m	1h25m	1h25m	1h25m	1h25m	1h25m
1	35m25s	36m54s	38m26s	40m09s	42m08s	46m10s	50m19s
2	18m03s	19m30s	21m04s	22m34s	24m16s	28m16s	32m16s
3	11m13s	12m24s	13m33s	15m00s	16m20s	19m36s	23m16s
4	8m11s	9m06s	10m08s	11m24s	12m39s	15m33s	18m51s
5	6m47s	7m43s	8m38s	9m39s	10m54s	13m34s	16m46s
6	6m02s	7m00s	7m52s	8m58s	10m06s	12m44s	15m47s
8	4m41s	5m26s	6m22s	7m10s	7m56s	10m14s	12m50s
10	4m34s	5m03s	5m51s	6m35s	7m18s	9m10s	11m29s
12	4m24s	5m03s	5m51s	6m35s	7m18s	9m11s	11m26s
15	4m24s	5m04s	5m50s	6m36s	7m17s	9m10s	11m26s

TAB. 5.11 – Temps cumulatif pris pour l'exécution de l'ensemble des tests pour différentes valeurs des paramètres  $D$  et  $\gamma$ .

Quant au degré d'échantillonnage, la valeur  $D = 6$  semble la plus appropriée. En effet, ce degré d'échantillonnage obtient des solutions quasi identiques à l'approche exhaustive pour un temps de calcul beaucoup moindre. Bien que des valeurs de  $D$  plus grandes offrent des solutions de qualité comparables, on observe tout de même une tendance vers la perte de qualité pour un gain de temps de calcul moins important. Pour ces raisons, la combinaison ( $D = 6, \gamma = 95\%$ ) semble la plus prometteuse.

À ce stade, il ne reste plus qu'un paramètre à fixer : le nombre  $N$  d'itérations d'optimisation locale dans l'algorithme 3.5. Puisque cet algorithme a également pour objectif de minimiser le temps de calcul sans significativement diminuer la qualité des solutions, nous nous attardons à la fois à l'indice de similarité moyen des tests effectués et au temps cumulatif nécessaire à l'exécution de ces tests. Le tableau 5.12 présente ces deux métriques pour les différentes valeurs de  $N$  testées.

En observant les résultats présentés dans le tableau 5.12, nous pouvons remarquer qu'une valeur de  $N = 40$  semble offrir le meilleur temps d'exécution sans montrer une perte significative de qualité des segmentations par rapport à l'approche exhaustive. Bien qu'il puisse sembler risqué de choisir un nombre d'itération constant alors que l'espace de recherche grandit en fonction du carré de  $l_S$ , nos résultats empiriques portent à croire qu'il n'en est rien. Ce comportement est sans doute attribuable au fait que l'espace de recherche ne comporte que peu de maxima locaux, ce qui permet à chaque itération de l'algorithme DHC de couvrir une grande partie de l'espace de recherche. De plus, le tableau 5.12 indique des temps d'exécution significativement plus importants pour obtenir un indice moyen comparable lorsque nous utilisons une valeur de  $N$

$N$	Indice $I_S$ moyen	Temps cumulatif
<b>10</b>	0.667	0m25s
<b>20</b>	0.690	0m53s
<b>30</b>	0.686	1m16s
<b>40</b>	0.706	1m38s
<b>50</b>	0.705	2m04s
<b>75</b>	0.705	3m02s
<b>100</b>	0.706	3m57s
<b>150</b>	0.706	5m56s
<b>200</b>	0.705	7m56s
<b>300</b>	0.707	12m43s
<b>500</b>	0.710	22m56s
$l_S / 4$	0.696	2m40s
$l_S / 2$	0.700	5m11s
$l_S$	0.703	10m11s
$(l_S)^2 / 32$	0.705	10m01s
$(l_S)^2 / 16$	0.706	20m05s

TAB. 5.12 – Indice  $I_S$  moyen et temps cumulatif pour l'ensemble des tests pour différentes valeurs du paramètre  $N$ .

fonction de  $l_S$  ou de  $(l_S)^2$ .

Les expérimentations présentées ci-dessus nous ont donc permis de fixer nos paramètres à des valeurs qui offrent de bons résultats sur l'ensemble de nos séquences de test. Le tableau 5.13 présente les valeurs fixées pour chaque paramètre. Ces valeurs sont celles utilisées dans toutes les expérimentations présentées ultérieurement.

## 5.5 Étude empirique des algorithmes

En ayant fixé empiriquement les valeurs des paramètres de chaque algorithme, nous pouvons avoir une certaine assurance que chaque algorithme pourra s'exécuter dans des conditions favorables. La suite logique de nos expérimentations consiste donc à confronter les divers algorithmes concurrents pour observer quels sont leurs avantages et inconvénients respectifs.

Nous avons deux groupes d'algorithmes concurrents : les algorithmes de fusion de segmentations et les algorithmes de recherche de meilleur segment. Dans chacun des cas,

Param.	Valeur
$\beta$	18
$\beta_P$	10
$\eta_E$	85%
$\eta_B$	45%
$\Delta_U$	10
$D$	6
$\gamma$	95%
$N$	40
$\epsilon$	2%

TAB. 5.13 – Liste des paramètres avec leur valeur fixes respectives.

nous allons premièrement procéder à des tests utilisant la segmentation anatomique et, ensuite, à des tests omettant la segmentation anatomique.

Notons que, dans le cas de la segmentation anatomique, les algorithmes ne seront pas testés sur les séquences de squelettes basés sur la méthode de suivi présentée par Jean et coll. dans [6]. Nous omettons volontairement ces séquences étant donné que les mouvements qu’elles présentent ne sont pas anatomiquement corrects par rapport aux mouvements de la séquence vidéo originale. Ceci est simplement dû la nature de l’approche, qui consiste à effectuer le suivi automatique des mains, des pieds, de la tête et du centre de masse. Nos expériences préliminaires utilisaient le centre de masse pour faire office de torse, mais nous avons vite remarqué que le centre de masse est, naturellement, déplacé par le mouvement des membres. Il devient donc impossible d’isoler les mouvements globaux du squelette de ses mouvements relatifs, ce qui est une visée importante de la segmentation anatomique. La segmentation temporelle d’activités cycliques à l’aide de cette topologie squelettique est toutefois tout indiquée, puisque tout mouvement cyclique naturel dans la séquence vidéo sera généralement traduit par un mouvement cyclique dans la séquence de squelettes. Donc, bien qu’elle ne soit pas anatomiquement représentative au niveau des mouvements, cette topologie squelettique conserve généralement la qualité cyclique ou non cyclique des mouvements. Les exceptions sont des mouvements où les articulations, telles que les genoux ou les coudes, sont en mouvements sans que les extrémités des membres ne soient en mouvement. Toutefois, peu de mouvements naturels répondent à ce critère.

Nos choix de membres et de torse pour la topologie de squelette présentée par Vignola dans [14] sont illustrés par la figure 5.4.

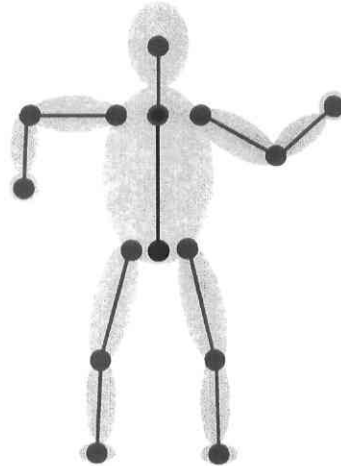


FIG. 5.4 – Segmentation anatomique de la topologie squelettique proposée dans [14]. Les éléments en rouge représentent les membres alors que les éléments en bleu composent le torse. La base du cou fait à la fois partie du torse et du *membre* de la tête.

### 5.5.1 Algorithmes de fusion

Dans le cadre de cet ouvrage, nous avons abordé deux algorithmes de fusion de segmentations distincts : la fusion simple (algorithme 4.1) et la fusion par différence (algorithme 4.3). En fait, ces deux algorithmes partagent une approche très similaire, mais le second se trouve à être une amélioration théorique du premier. Les tests présentés ici nous permettront donc d’apprécier la validité empirique de cette amélioration.

Pour comparer ces algorithmes, nous les avons exécutés sur l’ensemble de nos séquences de squelettes, d’abord en utilisant la segmentation anatomique et, ensuite, en l’omettant. Dans tous les cas, les signaux ont été segmentés à l’aide de l’approche *glouton* de segmentation de signal (algorithme 3.7) utilisant l’approche exhaustive pour la recherche du meilleur segment (algorithme 3.2).

Le tableau 5.14 présente l’indice de similarité, la précision et le rappel<sup>5</sup> obtenu par nos deux algorithmes de fusion sur chacune des séquences de squelettes permettant la segmentation anatomique. Le tableau 5.15 présente des résultats analogues pour une segmentation temporelle n’utilisant pas la segmentation anatomique. Puisqu’aucun des deux algorithmes de fusion n’est significativement plus rapide que l’autre, nous n’encombrerons pas cette analyse des tables de temps obtenues pour chaque test.

<sup>5</sup>Les résultats de précision et de rappel présentés dans ce chapitre sont calculés avec un seuil  $\tau = 75\%$ . Voir la section 1.2 pour de plus amples détails.

Séquence	Simple $I_S$	Simple Rappel	Simple Précision	Diff. $I_S$	Diff. Rappel	Diff. Précision
<b>VignA1</b>	58%	100%	100%	58%	100%	100%
<b>VignA2</b>	64%	100%	100%	64%	100%	100%
<b>VignA3</b>	75%	100%	75%	75%	100%	75%
<b>VignA4</b>	60%	100%	67%	60%	100%	67%
<b>VignA5</b>	36%	100%	67%	36%	100%	67%
<b>VignB1</b>	62%	100%	75%	62%	100%	75%
<b>VignB2</b>	38%	50%	100%	38%	50%	100%
<b>VignB3</b>	62%	100%	60%	62%	100%	60%
<b>VignB4</b>	59%	100%	56%	59%	100%	56%
<b>VignB5</b>	54%	100%	67%	54%	100%	67%
<b>VignC1</b>	57%	100%	100%	57%	100%	100%
<b>VignC2</b>	48%	60%	71%	48%	60%	71%
<b>VignC3</b>	61%	67%	67%	61%	67%	67%

TAB. 5.14 – Avec segmentation anatomique - Indice  $I_S$ , précision et rappel obtenu par l’algorithme de fusion simple (Simple) et l’algorithme de fusion par différence (Diff.) appliqué à chaque séquence de test.

En observant le tableau 5.14, nous remarquons premièrement que les indices de similarités obtenus sont généralement faibles et ne génèrent donc pas des segmentations adéquates. Ceci s’explique en partie par la qualité des séquences de squelettes, qui comportent un bruit important dans le mouvement. Toutefois, ces indices faibles s’expliquent surtout par la subjectivité de la segmentation humaine qui sert de référence. Le jugement humain semble effectivement beaucoup plus subjectif lorsque le temps vient de *décider* quels sont les membres qui participent à une certaine activité cyclique. À titre d’exemple, lorsque le sujet balance ses bras dans la séquence *VignA3*, le reste du corps n’est pas parfaitement fixe et le déplacement des bras induit un mouvement faible mais parfois cyclique au niveau des jambes. En utilisant un jugement humain, il est clair que le battement de bras n’inclut pas les jambes. Par contre, puisque notre approche n’a aucune connaissance sur le mouvement effectué, elle inclut ces mouvements cycliques des jambes dans sa segmentation finale qui serait autrement presque parfaite. La figure 5.5 illustre la segmentation temporelle effectuée pour le bras gauche sur la séquence *VignA3*<sup>6</sup>. Nous pouvons y remarquer que la segmentation automatique est très similaire à la segmentation de référence. La figure 5.6 illustre la périodicité de certaines portions de signaux représentant le mouvement des jambes dans cette même séquence. Ceci nous permet de bien voir que l’étape de segmentation des signaux n’est pas à blâmer. L’erreur est plutôt attribuable au fait que notre approche ne possède aucune connaissance sur

<sup>6</sup>La segmentation obtenue pour l’autre bras est pratiquement identique.

le mouvement effectué, alors qu'un observateur humain est guidé par ses connaissances lorsqu'il décide de quels membres doivent être inclus dans la segmentation.

Pour la majorité des séquences, les indices faibles s'expliquent par une telle différence entre les membres inclus dans l'activité cyclique par la segmentation de référence et ceux inclus par la segmentation automatique. Lorsque certains membres sont ajoutés à une activité cyclique par la segmentation automatique, nous pouvons remarquer un taux de précision plus faible, alors que lorsque certains membres manquent pour une activité cyclique, nous remarquons un taux de rappel plus faible.

Une autre part d'erreur, surtout dans les quelques cas où le taux de rappel et le taux de précision sont de 100%, s'explique par le fait que dans la segmentation de référence, pour une même activité cyclique, l'activité débute et se termine en même temps pour tous les membres qui participent à cette activité. Puisqu'elle traite chaque partie du corps indépendamment, la segmentation automatique ne peut toutefois pas procéder de cette façon. Il est alors difficile d'obtenir une segmentation précise pour tous les membres.

Toujours en observant le tableau 5.14, nous ne pouvons observer aucune différence entre les segmentations offertes par les deux algorithmes de fusion. Les deux algorithmes présentent des segmentations parfaitement identiques dans toutes les séquences. Ceci s'explique par le fait que la segmentation anatomique ne produit que rarement des chevauchements entre segments trouvés pour un même membre. Omettre la segmentation anatomique est toutefois plus propice à générer de tels chevauchements puisque les segments trouvés sur l'ensemble des parties du corps sont comparés.

Les résultats observés au tableau 5.15 montrent premièrement une performance significativement meilleure de notre approche lorsque la segmentation anatomique est omise. Nous notons que les deux algorithmes présentent des segmentations identiques pour toutes les séquences de type *A*, ce qui est normal étant donné que l'avantage théorique de la fusion par différence ne s'applique qu'à des situations où des activités cycliques sont temporellement adjacentes. Sur les séquences de type *B* et *C*, nous pouvons remarquer une différence entre les deux approches sur plusieurs séquences. Nous notons que bien que les deux approches obtiennent une segmentation presque identique sur la majorité des séquences, certaines séquences sont mieux segmentées à l'aide de la fusion par différence alors que d'autres sont mieux segmentées à l'aide de la fusion simple.

Tel qu'expliqué dans la section 3.3, les cas où l'algorithme de fusion par différence est préférable sont ceux où plusieurs activités cycliques temporellement adjacentes sont



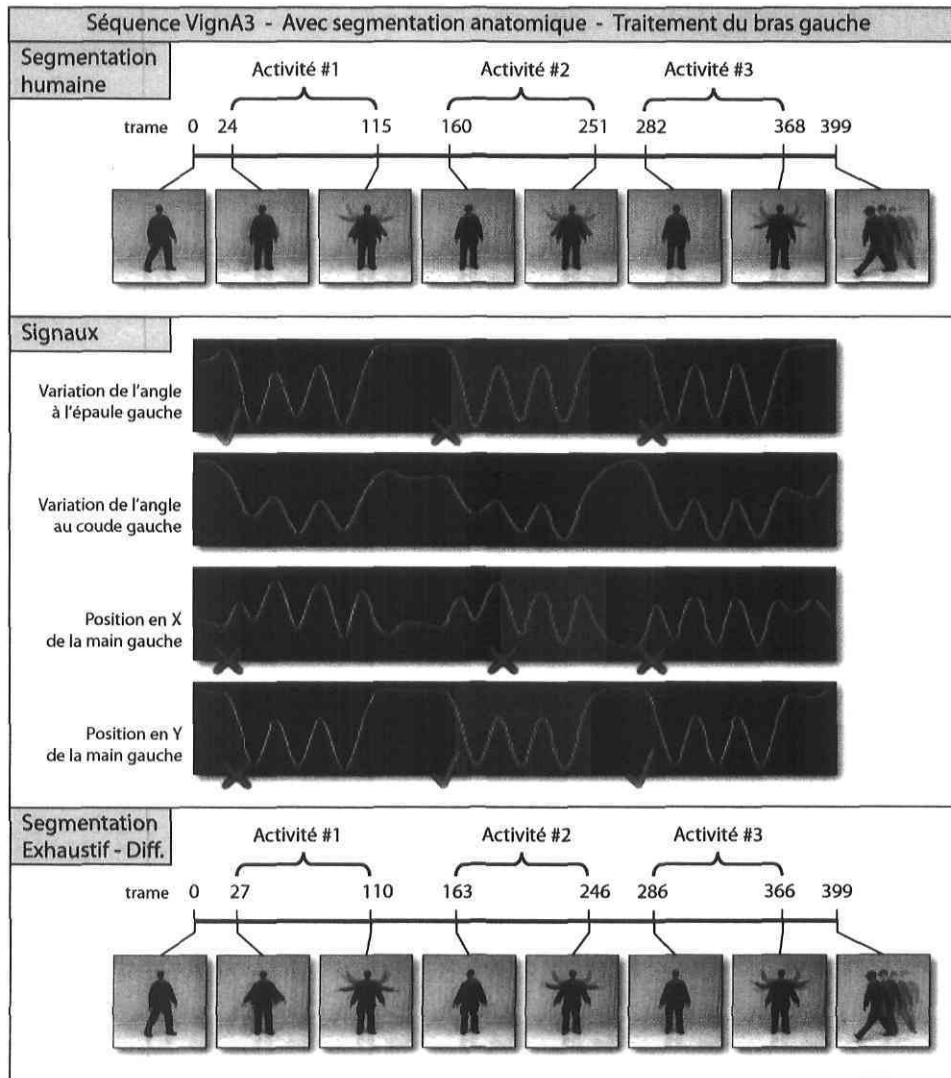


FIG. 5.5 – Segmentation de référence, segmentation des signaux et segmentation finale pour les mouvements du bras gauche dans la séquence *VignA3*. Le bleu et le rouge sont utilisés en alternance comme couleur de fond pour représenter les segments trouvés lors de la segmentation des signaux. Les crochets verts indiquent les segments retenus lors de la fusion alors que les X rouges indiquent les segments éliminés.

Séquence	Simple $I_S$	Simple Rappel	Simple Précision	Diff. $I_S$	Diff. Rappel	Diff. Précision
VignA1	71%	100%	100%	71%	100%	100%
VignA2	95%	100%	100%	95%	100%	100%
VignA3	92%	100%	100%	92%	100%	100%
VignA4	83%	100%	100%	83%	100%	100%
VignA5	65%	100%	100%	65%	100%	100%
JeanA1	84%	100%	100%	84%	100%	100%
JeanA2	74%	100%	100%	74%	100%	100%
JeanA3	90%	100%	100%	90%	100%	100%
JeanA4	72%	100%	100%	72%	100%	100%
JeanA5	38%	50%	100%	38%	50%	100%
JeanASitting	95%	100%	100%	95%	100%	100%
JeanATwoDiag	79%	100%	100%	79%	100%	100%
JeanATwoSide	80%	100%	100%	80%	100%	100%
VignB1	76%	100%	100%	69%	100%	100%
VignB2	48%	50%	100%	48%	50%	100%
VignB3	78%	100%	100%	78%	100%	100%
VignB4	87%	100%	100%	87%	100%	100%
VignB5	70%	100%	100%	86%	100%	100%
JeanB1	66%	100%	100%	66%	100%	100%
JeanB2	50%	100%	67%	50%	100%	67%
JeanB3	78%	100%	100%	78%	100%	100%
JeanB4	75%	100%	100%	73%	67%	67%
JeanB5	83%	100%	100%	79%	100%	100%
JeanBFourDir	41%	50%	100%	41%	50%	100%
JeanBLimpWalk	81%	100%	100%	81%	100%	100%
VignC1	86%	100%	100%	86%	100%	100%
VignC2	74%	80%	100%	74%	80%	100%
VignC3	54%	67%	100%	85%	100%	100%
JeanC1	83%	100%	100%	83%	100%	100%
JeanC2	72%	60%	75%	72%	60%	75%
JeanC3	69%	100%	100%	53%	100%	75%

TAB. 5.15 – Sans segmentation anatomique - Indice  $I_S$ , précision et rappel obtenu par l'algorithme de fusion simple (Simple) et l'algorithme de fusion par différence (Diff.) appliqué à chaque séquence de test.

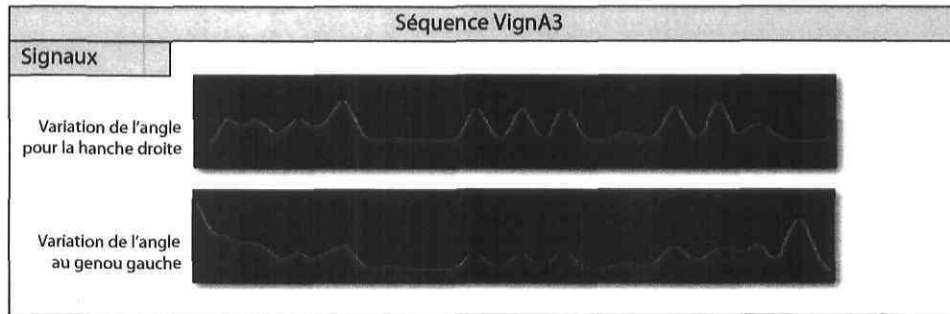


FIG. 5.6 – Segments cycliques trouvés par la segmentation automatique sur des signaux représentant le mouvement des jambes dans la séquence *VignA3*.

détectées avec un léger chevauchement temporel. L'algorithme de fusion simple engendre alors plus de détections manquées. Cette situation est pratiquement inévitable, même dans les meilleures conditions, et c'est pourquoi nous avons développé l'algorithme de fusion par différence. Les figures 5.7 et 5.8 illustrent bien cet avantage de l'algorithme de fusion par différence en comparant la segmentation finale obtenue par fusion simple et celle obtenue par segmentation par différence. Nous pouvons y remarquer que certains segments trouvés à l'étape de la segmentation des signaux regroupent plus d'une activité cyclique. Sans l'approche de fusion par différence, l'information sur la première de ces deux activités est simplement perdue. Cette situation s'observe lorsque le rappel de l'approche de fusion simple est plus faible que celui de la fusion par différence.

La situation inverse, où l'algorithme de fusion simple obtient une segmentation plus intéressante, est également possible. Elle peut se présenter lorsque de fausses détections sont obtenues à l'étape de la segmentation des signaux ou lorsque des segments cycliques plus longs sont divisés en plusieurs parties par la segmentation des signaux. Lorsque ces segments indésirables sont en chevauchement temporel avec un segment cyclique correct, la fusion simple risque fort de s'en défaire, alors que l'approche de fusion par différence risque de les conserver dans la segmentation finale, expliquant un indice de similarité plus faible. Cette situation s'observe lorsque la précision de l'approche de fusion par différence est plus faible que celle de la fusion simple. Les figures 5.9 et 5.10 illustrent cet avantage que la fusion simple peut connaître. Dans cet exemple, la seconde activité cyclique n'est segmentée que partiellement par le premier segment retenu. Alors que la fusion simple s'en tient à ce segment, la fusion par différence vient recouvrir le reste de l'activité à l'aide d'un autre segment généré par l'algorithme de différence. Puisque l'indice de similarité n'admet qu'un seul segment par activité cyclique dans la segmentation de référence, ceci pénalise fortement l'approche de fusion par différence. Il est toutefois possible, dans certains contextes, que ce comportement de l'algorithme de fusion par différence soit souhaitable. Nous nous en tiendrons toutefois

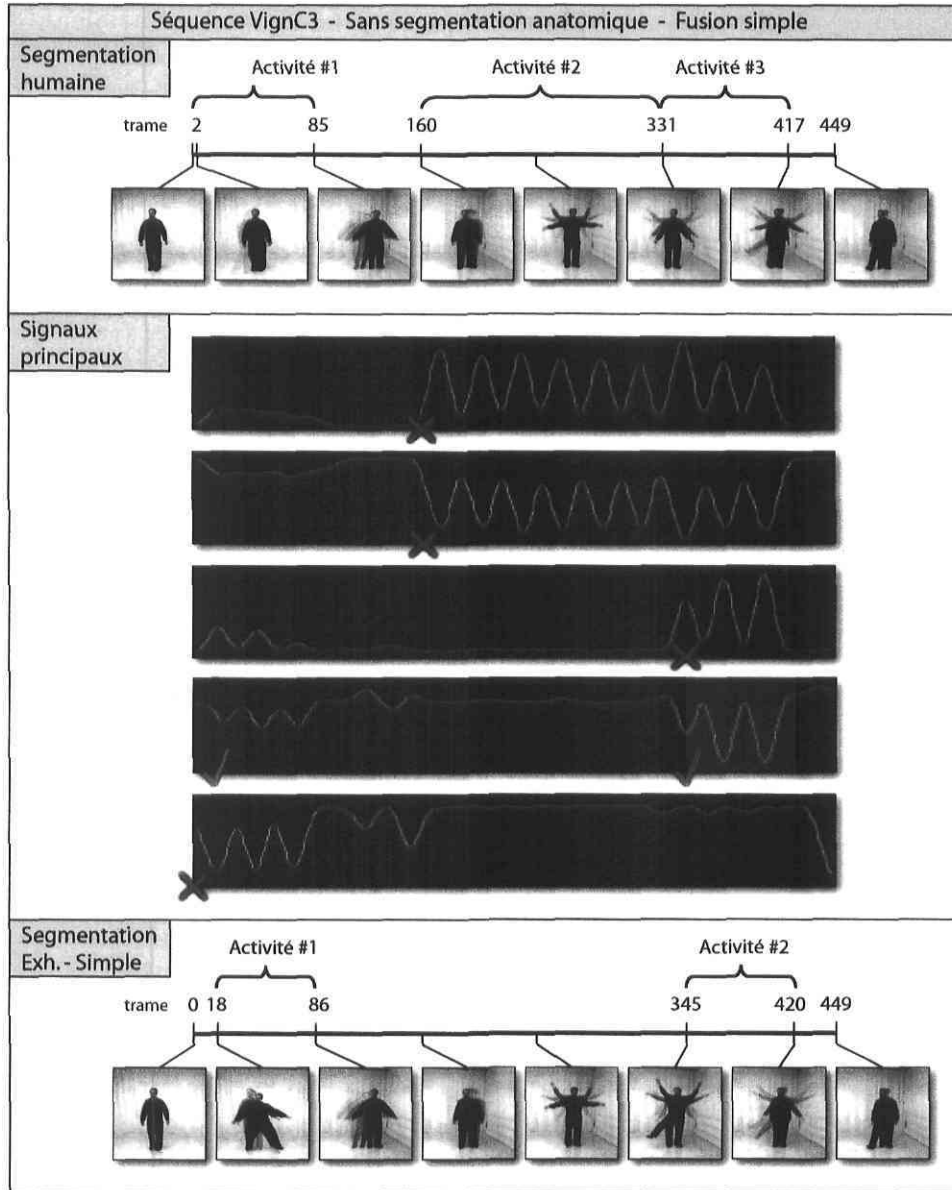


FIG. 5.7 – Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence *VignC3* en utilisant l’algorithme de fusion simple. Les crochets verts indiquent les segments retenus lors de la fusion alors que les *X* rouges indiquent les segments éliminés.

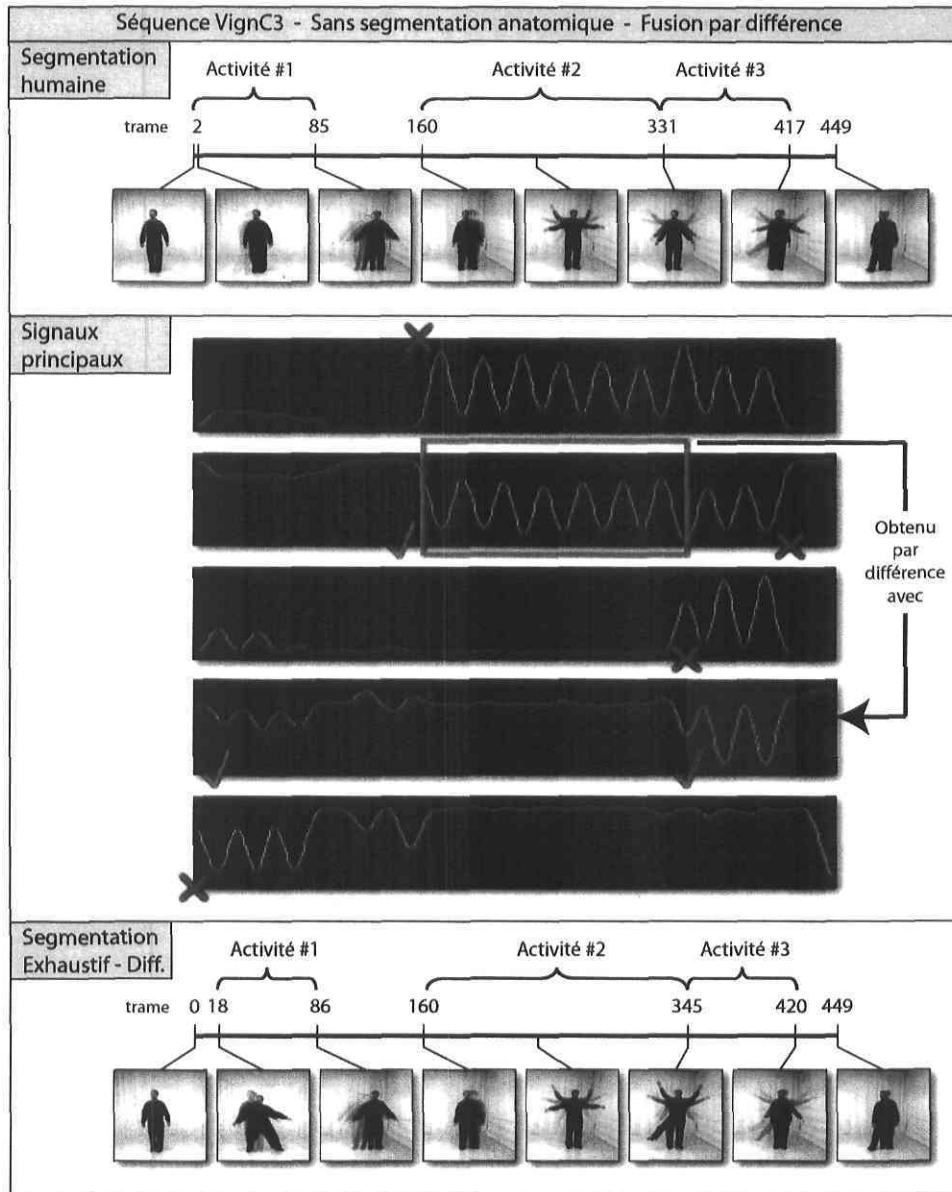


FIG. 5.8 – Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence *VignC3* en utilisant l’algorithme de fusion par différence. L’encadré vert représente un segment retenu qui a été obtenu par l’algorithme de différence.

uniquement aux objectifs que nous avons fixés au préalable, soit d’avoir la meilleure correspondance possible entre la segmentation de référence et la segmentation obtenue automatiquement.

Les autres séquences, où les algorithmes de fusion obtiennent un indice de similarité différent mais obtiennent un rappel et une précision similaire, s’expliquent simplement par le choix de différents segments. À titre d’exemple, si dans la figure 5.7 un segment isolant assez bien la deuxième activité cyclique avait été trouvé sur un signal, la fusion simple aurait permis de détecter les trois activités. Le résultat aurait alors pu être meilleur ou pire que celui obtenu par la fusion par différence, en fonction des segments retenus.

Puisque l’avantage de la fusion simple sur la fusion par différence tient davantage d’erreurs à l’étape de la segmentation des signaux que d’une astuce délibérée de cette approche, nous opterons généralement pour utiliser la fusion par différence, dont l’avantage est défendu théoriquement.

### 5.5.2 Algorithmes de recherche du meilleur segment

La section 3.3 du présent document présente quatre méthodes différentes pour rechercher le segment le plus cyclique d’un signal :

- l’approche exhaustive (algorithme 3.2),
- l’échantillonnage uniforme (algorithme 3.3),
- l’échantillonnage sélectif (algorithme 3.4)
- et l’optimisation numérique (algorithme 3.5).

Contrairement aux algorithmes de fusion, les algorithmes de recherche du meilleur segment ont un impact majeur sur le temps de calcul nécessaire à la segmentation temporelle d’une séquence en plus d’avoir un impact de premier plan sur la qualité de la solution trouvée. Il est donc pertinent d’analyser ces différents algorithmes à la fois sous l’aspect de la performance et sous l’aspect de la qualité des segmentations obtenues.

En utilisant la segmentation anatomique, chaque algorithme a été testé sur l’ensemble des séquences de test. L’algorithme de fusion par différence a été utilisé. Le tableau 5.16 présente l’indice de similarité obtenu pour les solutions générées par chaque algorithme sur chaque séquence de test. Les tableaux 5.17 et 5.18 présentent respectivement le rappel et la précision obtenus sur chaque séquence. Le tableau 5.23 quant à lui présente le temps nécessaire à la segmentation temporelle de chaque séquence traitée par chaque algorithme. Étant donné sa nature non déterministe, les résultats

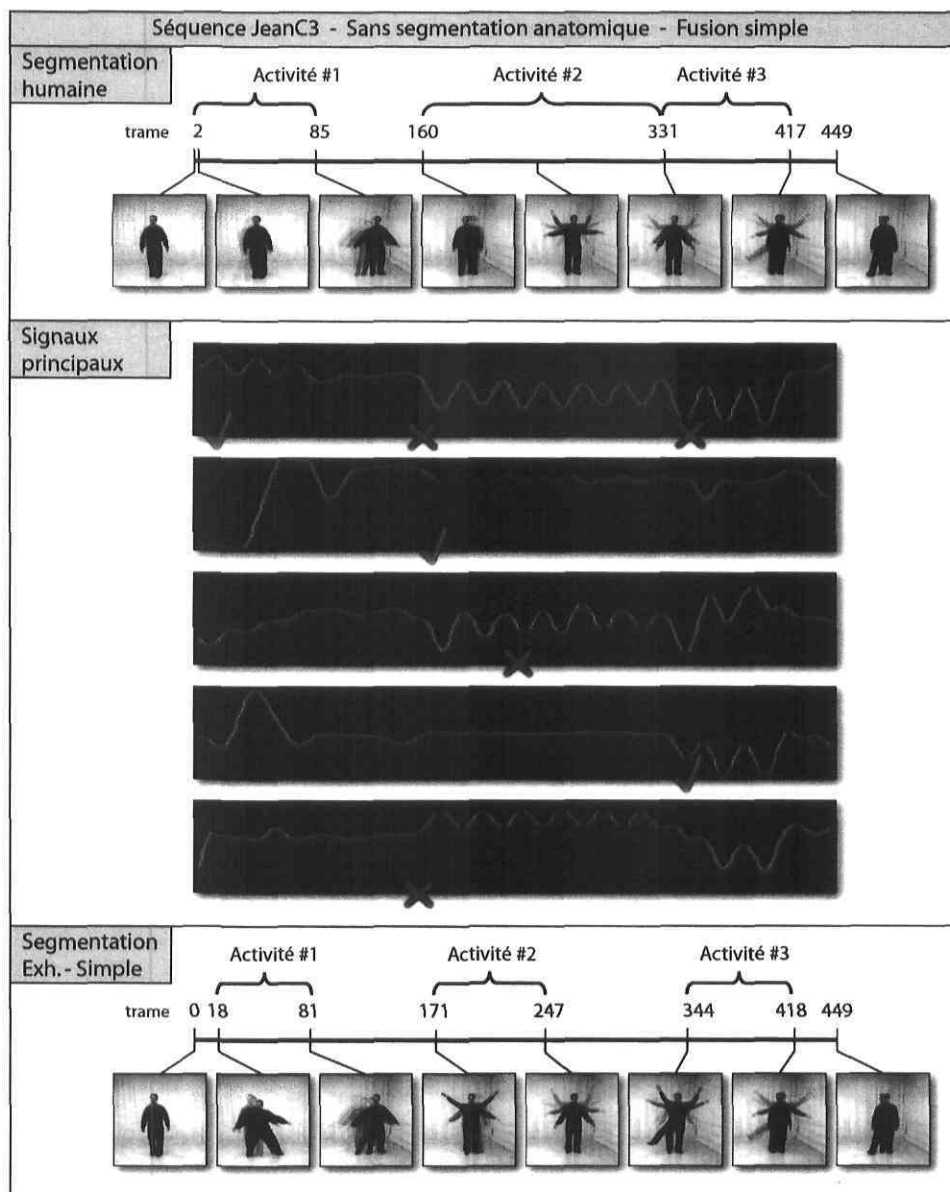


FIG. 5.9 – Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence *JeanC3* en utilisant l’algorithme de fusion simple.

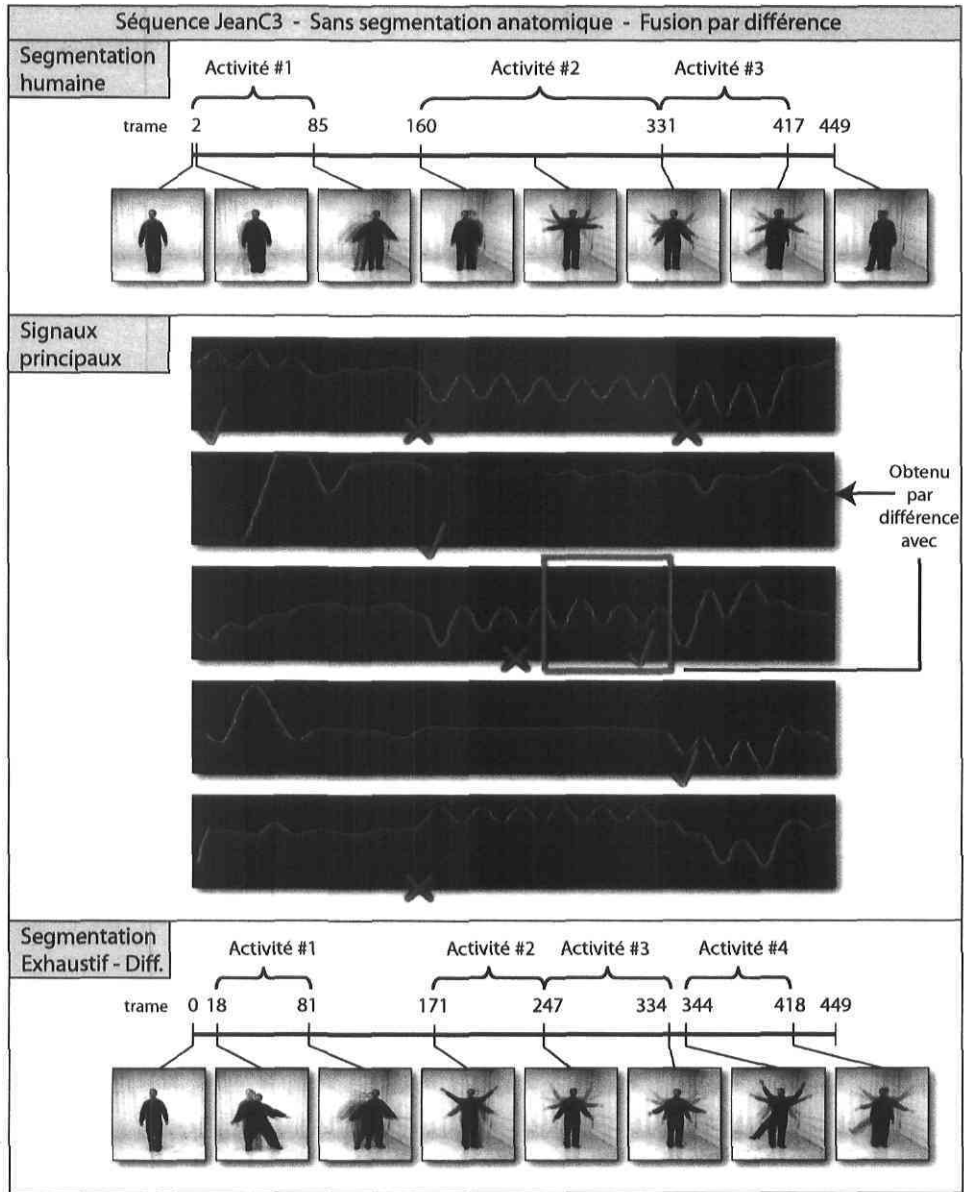


FIG. 5.10 – Segmentation de référence, segmentation des signaux les plus importants et segmentation finale pour la séquence *JeanC3* en utilisant l’algorithme de fusion par différence. L’encadré vert représente un segment retenu qui a été obtenu par l’algorithme de différence.



Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	58%	60%	58%	59%
<b>VignA2</b>	64%	58%	64%	60%
<b>VignA3</b>	75%	70%	75%	79%
<b>VignA4</b>	60%	93%	73%	69%
<b>VignA5</b>	36%	43%	36%	36%
<b>VignB1</b>	62%	61%	62%	67%
<b>VignB2</b>	38%	18%	38%	38%
<b>VignB3</b>	62%	61%	62%	62%
<b>VignB4</b>	59%	65%	59%	49%
<b>VignB5</b>	54%	53%	54%	55%
<b>VignC1</b>	57%	63%	57%	58%
<b>VignC2</b>	48%	55%	41%	46%
<b>VignC3</b>	61%	51%	61%	48%

TAB. 5.16 – Avec segmentation anatomique - Indice  $I_S$  obtenu par chacun des algorithmes appliqué à chaque séquence de test.

présentés pour l’approche d’optimisation numérique sont des moyennes obtenues sur 100 itérations de l’algorithme.

Les tableaux 5.20, 5.21, 5.22 et 5.23 sont les tableaux analogues pour la segmentation temporelle omettant la segmentation anatomique.

Une fois de plus, en observant les résultats présentés par les tableaux 5.16, 5.17 et 5.18, nous observons que les segmentations obtenues ne sont généralement pas de bonne qualité. Les explications données dans la section 5.5.1 sont également valides ici.

À partir de ces tableaux, il est difficile de bien apprécier si le type de séquence, soit A, B ou C, influence la qualité de la solution étant donné que la difficulté provient davantage des failles de l’approche de génération de la séquence de squelettes plutôt que du contenu en activités cycliques de la séquence vidéo concernée.

Plusieurs observations peuvent être faites à l’aide des tableaux 5.20, 5.21 et 5.22. Une première telle observation est que les taux de rappel et de précision sont très élevés sur une majorité de séquences. Ceci nous indique que, lorsque l’on omet la segmentation anatomique, nos algorithmes ne font que peu de détections manquées et ne font que peu de fausses détections. Nous pouvons donc avoir une bonne confiance en la qualité des segmentations obtenues quant à leurs détections. Soulignons que cela n’indique toutefois pas que les activités cycliques sont segmentées avec exactitude. Par contre, un grand nombre de segmentations obtiennent un indice  $I_S$  supérieur à 70%, ce qui est

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
VignA1	100%	100%	100%	100%
VignA2	100%	100%	100%	100%
VignA3	100%	100%	100%	100%
VignA4	100%	100%	100%	100%
VignA5	100%	100%	100%	100%
VignB1	100%	80%	100%	100%
VignB2	50%	50%	50%	50%
VignB3	100%	100%	100%	100%
VignB4	100%	100%	100%	67%
VignB5	100%	100%	100%	100%
VignC1	100%	100%	100%	100%
VignC2	60%	80%	60%	60%
VignC3	67%	67%	67%	60%

TAB. 5.17 – Avec segmentation anatomique - Rappel obtenu par chacun des algorithmes appliqué à chaque séquence de test.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
VignA1	100%	100%	100%	100%
VignA2	100%	100%	100%	100%
VignA3	75%	75%	75%	100%
VignA4	67%	100%	80%	67%
VignA5	67%	60%	67%	67%
VignB1	75%	80%	75%	82%
VignB2	100%	100%	100%	100%
VignB3	60%	60%	60%	60%
VignB4	56%	71%	63%	54%
VignB5	67%	67%	67%	67%
VignC1	100%	86%	71%	74%
VignC2	71%	75%	67%	68%
VignC3	67%	60%	67%	57%

TAB. 5.18 – Avec segmentation anatomique - Précision obtenue par chacun des algorithmes appliqué à chaque séquence de test.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	5.4s	0.2s	0.7s	0.4s
<b>VignA2</b>	3.4s	0.2s	1.1s	1.0s
<b>VignA3</b>	2.5s	0.1s	0.5s	0.3s
<b>VignA4</b>	4.6s	0.2s	1.5s	0.9s
<b>VignA5</b>	2.0s	0.1s	0.6s	0.5s
<b>VignB1</b>	37.6s	0.7s	4.6s	2.0s
<b>VignB2</b>	9.5s	0.2s	1.8s	0.8s
<b>VignB3</b>	2.8s	0.2s	0.9s	0.8s
<b>VignB4</b>	6.3s	0.2s	0.9s	0.8s
<b>VignB5</b>	9.0s	0.2s	1.4s	0.9s
<b>VignC1</b>	8.4s	0.2s	2.0s	0.8s
<b>VignC2</b>	8.6s	0.2s	1.4s	0.6s
<b>VignC3</b>	5.0s	0.2s	0.9s	0.9s

TAB. 5.19 – Avec segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test.

très acceptable, sans toutefois frôler la perfection. En considérant les valeurs élevées de taux de rappel et de précision, ces valeurs d'indice s'expliquent dans la majorité des cas par des segments légèrement tronquées ou légèrement allongées.

Une seconde observation est que, une fois de plus, il est difficile de voir si les séquences de type C sont plus exigeantes pour nos algorithmes que des séquences de type B ou A. En effet, nous n'obtenons pas des résultats significativement pires ou meilleurs pour un type de séquence donné. Il semble donc que la proximité mutuelle des activités cycliques ne soit pas la difficulté principale rencontrée par nos algorithmes.

Nous pouvons également faire une troisième observation sur les méthodes d'extraction de squelettes. Parmi les résultats présentés, nous observons qu'aucune des deux méthodes utilisées ne semble générer des séquences de squelettes plus faciles à segmenter. Dans certains cas, l'approche proposée par Vignola [14] donne des séquences de squelettes pour lesquelles les segmentations sont significativement meilleures que sur les séquences de squelettes générées par l'approche dérivée de travaux de Jean et coll. [6] pour une même séquence vidéo. Dans d'autres cas, l'inverse se produit. Ceci nous amène à la conclusion, peu surprenante, qu'aucune de ces deux méthodes n'est bien adaptée à la tâche à laquelle nous les confrontons.

L'ensemble des tests effectués sur les quatre algorithmes, avec et sans segmentation anatomique, nous permet de les comparer. Sur le plan de la qualité des solutions, les quatre algorithmes semblent offrir des solutions comparables. Ils présentent en effet des

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	71%	76%	71%	73%
<b>VignA2</b>	95%	91%	95%	94%
<b>VignA3</b>	92%	87%	92%	92%
<b>VignA4</b>	83%	93%	96%	77%
<b>VignA5</b>	65%	62%	65%	64%
<b>JeanA1</b>	84%	91%	84%	88%
<b>JeanA2</b>	74%	74%	74%	68%
<b>JeanA3</b>	90%	91%	90%	89%
<b>JeanA4</b>	72%	74%	72%	72%
<b>JeanA5</b>	38%	35%	38%	37%
<b>JeanASitting</b>	95%	95%	95%	73%
<b>JeanATwoDiag</b>	79%	78%	79%	79%
<b>JeanATwoSide</b>	80%	78%	80%	79%
<b>VignB1</b>	69%	55%	69%	69%
<b>VignB2</b>	48%	44%	48%	47%
<b>VignB3</b>	78%	84%	78%	80%
<b>VignB4</b>	87%	83%	87%	78%
<b>VignB5</b>	86%	85%	86%	88%
<b>JeanB1</b>	66%	60%	66%	66%
<b>JeanB2</b>	50%	50%	50%	50%
<b>JeanB3</b>	78%	82%	78%	80%
<b>JeanB4</b>	73%	74%	73%	72%
<b>JeanB5</b>	79%	74%	86%	79%
<b>JeanBFourDir</b>	41%	41%	41%	41%
<b>JeanBLimpWalk</b>	81%	85%	81%	75%
<b>VignC1</b>	86%	83%	86%	77%
<b>VignC2</b>	74%	74%	58%	75%
<b>VignC3</b>	85%	90%	85%	82%
<b>JeanC1</b>	83%	67%	83%	81%
<b>JeanC2</b>	72%	73%	70%	71%
<b>JeanC3</b>	53%	27%	53%	44%

TAB. 5.20 – Sans segmentation anatomique - Indice  $I_S$  obtenu par chacun des algorithmes appliqué à chaque séquence de test.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
VignA1	100%	100%	100%	100%
VignA2	100%	100%	100%	100%
VignA3	100%	100%	100%	100%
VignA4	100%	100%	100%	100%
VignA5	100%	50%	100%	100%
JeanA1	100%	100%	100%	100%
JeanA2	100%	100%	100%	100%
JeanA3	100%	100%	100%	100%
JeanA4	100%	100%	100%	100%
JeanA5	100%	50%	50%	50%
JeanASitting	100%	100%	100%	50%
JeanATwoDiag	100%	100%	100%	100%
JeanATwoSide	100%	100%	100%	100%
VignB1	100%	80%	100%	100%
VignB2	50%	50%	50%	50%
VignB3	100%	100%	100%	100%
VignB4	100%	100%	100%	93%
VignB5	100%	100%	100%	100%
JeanB1	100%	80%	100%	100%
JeanB2	100%	100%	100%	100%
JeanB3	100%	100%	100%	100%
JeanB4	67%	100%	67%	87%
JeanB5	100%	75%	100%	100%
JeanBFourDir	50%	25%	50%	50%
JeanBLimpWalk	100%	100%	100%	70%
VignC1	100%	100%	100%	100%
VignC2	80%	80%	60%	80%
VignC3	100%	100%	100%	87%
JeanC1	100%	100%	100%	100%
JeanC2	60%	60%	40%	56%
JeanC3	100%	67%	100%	67%

TAB. 5.21 – Sans segmentation anatomique - Rappel obtenu par chacun des algorithmes appliqué à chaque séquence de test.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	100%	100%	100%	100%
<b>VignA2</b>	100%	100%	100%	100%
<b>VignA3</b>	100%	100%	100%	100%
<b>VignA4</b>	100%	100%	100%	100%
<b>VignA5</b>	100%	50%	100%	100%
<b>JeanA1</b>	100%	100%	100%	100%
<b>JeanA2</b>	100%	100%	100%	100%
<b>JeanA3</b>	100%	100%	100%	100%
<b>JeanA4</b>	100%	100%	100%	100%
<b>JeanA5</b>	100%	100%	100%	100%
<b>JeanASitting</b>	100%	100%	100%	60%
<b>JeanATwoDiag</b>	100%	100%	100%	100%
<b>JeanATwoSide</b>	100%	100%	100%	100%
<b>VignB1</b>	100%	100%	100%	100%
<b>VignB2</b>	100%	100%	100%	100%
<b>VignB3</b>	100%	100%	100%	100%
<b>VignB4</b>	100%	100%	100%	93%
<b>VignB5</b>	100%	100%	100%	100%
<b>JeanB1</b>	100%	100%	100%	100%
<b>JeanB2</b>	67%	67%	67%	67%
<b>JeanB3</b>	100%	100%	100%	100%
<b>JeanB4</b>	67%	100%	67%	87%
<b>JeanB5</b>	100%	75%	100%	100%
<b>JeanBFourDir</b>	100%	50%	100%	100%
<b>JeanBLimpWalk</b>	100%	100%	100%	80%
<b>VignC1</b>	100%	100%	100%	93%
<b>VignC2</b>	100%	100%	100%	100%
<b>VignC3</b>	100%	100%	100%	90%
<b>JeanC1</b>	100%	100%	100%	100%
<b>JeanC2</b>	75%	75%	50%	70%
<b>JeanC3</b>	75%	67%	75%	50%

TAB. 5.22 – Sans segmentation anatomique - Précision obtenue par chacun des algorithmes appliqué à chaque séquence de test.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	5.3s	0.1s	0.7s	0.4s
<b>VignA2</b>	4.0s	0.2s	1.1s	0.9s
<b>VignA3</b>	2.5s	0.1s	0.4s	0.3s
<b>VignA4</b>	4.8s	0.2s	1.4s	0.8s
<b>VignA5</b>	2.5s	0.1s	0.7s	0.6s
<b>JeanA1</b>	33.3s	0.5s	6.0s	1.5s
<b>JeanA2</b>	39.4s	0.8s	5.2s	3.1s
<b>JeanA3</b>	18.4s	0.3s	2.4s	0.9s
<b>JeanA4</b>	14m21s	9.9s	44.0s	4.6s
<b>JeanA5</b>	9.9s	0.2s	2.4s	1.2s
<b>JeanASitting</b>	1m52s	1.5s	9.0s	1.9s
<b>JeanATwoDiag</b>	39.7s	0.6s	8.6s	1.9s
<b>JeanATwoSide</b>	1m05s	1.0s	8.0s	2.0s
<b>VignB1</b>	40.1s	0.7s	5.3s	1.8s
<b>VignB2</b>	9.3s	0.2s	1.8s	0.8s
<b>VignB3</b>	3.1s	0.2s	1.0s	0.7s
<b>VignB4</b>	10.0s	0.2s	1.1s	0.8s
<b>VignB5</b>	9.2s	0.3s	1.4s	0.7s
<b>JeanB1</b>	10m16s	7.4s	30.9s	6.1s
<b>JeanB2</b>	1m40s	1.4s	9.3s	3.1s
<b>JeanB3</b>	1m05s	1.1s	12.8s	3.7s
<b>JeanB4</b>	1m53s	1.5s	14.1s	2.4s
<b>JeanB5</b>	2m58s	2.3s	14.7s	4.0s
<b>JeanBFourDir</b>	1m39s	1.5s	5.7s	2.5s
<b>JeanBLimpWalk</b>	18.3s	0.3s	2.2s	0.8s
<b>VignC1</b>	8.5s	0.2s	2.0s	0.8s
<b>VignC2</b>	8.7s	0.2s	1.4s	0.6s
<b>VignC3</b>	5.1s	0.2s	0.9s	0.8s
<b>JeanC1</b>	1m03s	0.9s	6.5s	2.4s
<b>JeanC2</b>	1m09s	1.0s	7.1s	1.9s
<b>JeanC3</b>	1m55s	1.6s	9.7s	2.9s

TAB. 5.23 – Sans segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test.

valeurs d'indice  $I_S$ , de rappel et de précision très similaires sur une grande majorité de séquences. La qualité des segmentations ne constitue donc pas une base sur laquelle nous pouvons discriminer ces algorithmes.

Sur le plan de la performance, nous pouvons toutefois remarquer que l'échantillonnage uniforme est de loin l'approche la plus rapide, offrant un gain de vitesse appréciable par rapport à toutes les autres approches, et surtout par rapport à l'approche exhaustive. L'échantillonnage sélectif et l'optimisation numérique font également preuve d'une performance digne de mention, offrant toujours un temps de calcul bien en deçà de la durée de la séquence traitée, un aspect important si nous désirons élaborer une méthode temps réel à partir de ces travaux.

Dans l'ensemble, puisqu'elle offre des solutions de qualité comparable à l'approche exhaustive tout en étant de loin la méthode la plus rapide, nous croyons que l'échantillonnage uniforme peut être préféré aux autres algorithmes.

### Segmentation idéale numérique

Parmi les approches de segmentation de signaux présentées, l'approche exhaustive cherche à obtenir la segmentation idéale alors que les trois approches sous-optimales cherchent à s'approcher le plus possible de la solution obtenue par l'approche exhaustive tout en réduisant considérablement le temps de calcul.

Il existe toutefois une différence importante entre les segmentations obtenues par l'approche exhaustive et les segmentations de référence effectuées par un humain. Nous dirons de la première qu'elle est une segmentation idéale numérique, puisque basée sur un modèle numérique, alors que la deuxième, obtenue par un opérateur humain, sera qualifiée de segmentation idéale subjective. Tel qu'observé dans les tableaux précédents, il est difficile de bien comparer la qualité des segmentations obtenues par les approches sous-optimales. En effet, pour un signal donné, la segmentation obtenue par une des approches sous-optimales vise à approcher la segmentation idéale numérique, mais elle est évaluée par rapport la segmentation idéale subjective. Une segmentation peut donc, dans certaines mesures, s'éloigner de la segmentation idéale numérique tout en s'approchant de la segmentation idéale subjective. Notons que ceci n'est pas un comportement souhaité pour les approches de segmentation sous-optimales que nous avons présentées.

Une difficulté qui demeure toujours à surmonter est d'élaborer un modèle numérique permettant d'obtenir une segmentation idéale numérique très rapprochée de la segmentation idéale subjective correspondante. Notre modèle actuel, basé sur l'indice



Séquence	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	91%	100%	92%
<b>VignA2</b>	94%	100%	98%
<b>VignA3</b>	93%	100%	99%
<b>VignA4</b>	86%	87%	96%
<b>VignA5</b>	90%	90%	90%
<b>VignB1</b>	70%	100%	99%
<b>VignB2</b>	93%	100%	100%
<b>VignB3</b>	90%	100%	94%
<b>VignB4</b>	91%	100%	90%
<b>VignB5</b>	94%	100%	95%
<b>VignC1</b>	84%	85%	76%
<b>VignC2</b>	94%	72%	94%
<b>VignC3</b>	91%	100%	84%

TAB. 5.24 – Sans segmentation anatomique - Indice  $I_S$  obtenu par chacun des algorithmes appliqué à un sous-ensemble des séquences de test en utilisant l’approche exhaustive comme segmentation de référence.

$l_S$ -normalisé  $Y_S$ , ne remplit pas complètement cette fonction.

Ceci étant dit, il est toutefois possible d’utiliser les résultats de l’approche exhaustive comme segmentations de références pour mieux comparer les approches de segmentation sous-optimales. Le tableau 5.24 présente les indices de similarité obtenus par les trois algorithmes de segmentation sous-optimaux lorsque la segmentation idéale numérique est utilisée comme segmentation de référence. Afin d’en alléger la présentation, nous ne présentons qu’un sous-ensemble des tests effectués. Les résultats sont représentatifs de ceux obtenus sur l’ensemble des séquences. Le tableau 5.25 rappelle les temps nécessaires à chaque algorithme pour procéder à la segmentation de la séquence <sup>7</sup>.

En observant les tableaux 5.24 et 5.25, nous pouvons noter que l’échantillonnage sélectif semble être la méthode la plus prometteuse des trois puisqu’elle offre souvent des segmentations parfaitement identiques à l’approche exhaustive pour une fraction du temps de calcul. Les approches d’optimisation numérique et d’échantillonnage uniforme offrent généralement des segmentations très proches de celles obtenues par l’approche exhaustive. L’échantillonnage uniforme obtient de loin le temps de calcul le plus faible alors que l’optimisation numérique semble offrir des résultats légèrement meilleurs. Il demeure toutefois difficile de déterminer laquelle de ces deux méthodes doit être préférée à l’autre. Il est par contre important de se rappeler que l’approche d’optimisation

<sup>7</sup>Les temps de segmentation de l’approche exhaustive sont fournis en référence.

Séquence	Exhaustif	Uniforme	Sélectif	Optimisation
<b>VignA1</b>	5.3s	0.1s	0.7s	0.4s
<b>VignA2</b>	4.0s	0.2s	1.1s	0.9s
<b>VignA3</b>	2.5s	0.1s	0.4s	0.3s
<b>VignA4</b>	4.8s	0.2s	1.4s	0.8s
<b>VignA5</b>	2.5s	0.1s	0.7s	0.6s
<b>VignB1</b>	40.1s	0.7s	5.3s	1.8s
<b>VignB2</b>	9.3s	0.2s	1.8s	0.8s
<b>VignB3</b>	3.1s	0.2s	1.0s	0.7s
<b>VignB4</b>	10.0s	0.2s	1.1s	0.8s
<b>VignB5</b>	9.2s	0.3s	1.4s	0.7s
<b>VignC1</b>	8.5s	0.2s	2.0s	0.8s
<b>VignC2</b>	8.7s	0.2s	1.4s	0.6s
<b>VignC3</b>	5.1s	0.2s	0.9s	0.8s

TAB. 5.25 – Sans segmentation anatomique - Temps nécessaires à chaque algorithme pour la segmentation temporelle de chaque séquence de test.

numérique n'est pas une approche déterministe.

Ces résultats nous informent de quels algorithmes seraient à favoriser dans une situation où les segmentations idéales numériques s'approchent suffisamment des segmentations idéales subjectives. Dans le contexte actuel, tel que mentionné plus tôt, tous les algorithmes offrent des segmentations de qualité comparables étant donné le biais produit par notre modèle numérique imparfait. L'échantillonnage uniforme est donc préféré à l'échantillonnage sélectif.

# Chapitre 6

## Conclusion

Dans cet ouvrage, nous avons présenté une approche générale pour la segmentation temporelle et anatomique des mouvements cycliques présents dans une séquence de squelettes. Cette approche, basée sur le traitement de signaux, consiste à extraire de la séquence un ensemble de signaux caractérisant bien les mouvements effectués, à segmenter temporellement les parties cycliques de chaque signal et, finalement, à fusionner ces segmentations pour obtenir une segmentation temporelle unique pour chaque partie du corps.

Ce faisant, nous avons également présenté différents algorithmes utilisés pour la segmentation de signaux et la fusion d'un ensemble de segmentations. Ayant présenté plus d'un algorithme pouvant remplir chaque tâche, nous les avons testés empiriquement pour faire ressortir les avantages et inconvénients de chacun.

Au niveau de la segmentation des signaux, un algorithme glouton est utilisé, faisant appel à la soustraction de pauses afin de réduire la complexité du problème, lorsque possible. Les variantes se trouvent ici dans l'approche utilisée pour rechercher le segment le plus cyclique dans un signal donné. Nous avons proposé et étudié quatre approches différentes pour résoudre ce problème à partir d'un indice de périodicité que nous avons défini. Nous avons d'abord défini l'approche exhaustive, qui évalue tous les segments possibles et retourne celui ayant le meilleur indice de périodicité. Pour pallier à la lenteur propre à cette approche, nous avons proposé les approches d'échantillonnage uniforme et d'échantillonnage sélectif. Alors que la première approche procède à la façon de l'approche exhaustive, mais sur un sous-ensemble de segments équidistants, la seconde raffine son échantillonnage de façon itérative en se concentrant sur les régions où les segments sont prometteurs. Finalement, nous avons proposé une approche d'optimisation numérique ayant une complexité de calcul de beaucoup moindre à l'approche

exhaustive, mais dont les résultats ne sont pas déterministes.

- En théorie, chacune de ces approches présente des avantages et des désavantages :
- l'approche exhaustive devrait générer une segmentation idéale, mais au coût d'une exécution relativement lente ;
  - l'échantillonnage sélectif devrait pouvoir s'approcher beaucoup de cette segmentation idéale, à un coup de calcul significativement moindre ;
  - l'échantillonnage uniforme et l'optimisation numérique devraient offrir une segmentation plus approximative, mais avec une complexité de calcul fortement réduite.

En pratique, nous avons toutefois observé que les maxima de l'indice de périodicité  $Y_S$  ne correspondent pas exactement aux segments de référence fournis par un humain. Ils s'en approchent, mais semblent plutôt les estimer. Ceci fait donc de l'approche exhaustive une méthode approximative. Puisque les trois autres approches sont des formes d'approximation de l'approche exhaustive, les segments qu'ils obtiennent se trouvent généralement assez près des segments de référence. Toutefois, comparativement aux segments trouvés par l'approche exhaustive, ceux trouvés par les autres approches se trouvent parfois plus loin des segments de références, mais peuvent également s'en trouver plus près. Donc, pour une séquence donnée, il est possible que l'optimisation numérique ou une approche d'échantillonnage produise une segmentation plus précise que celle produite par l'approche exhaustive. Ceci a pour effet que les quatre méthodes offrent des résultats de qualité comparable lorsque nous considérons l'ensemble des séquences. Tel que nous l'avons illustré en utilisant l'approche exhaustive comme segmentation de référence, si l'indice de périodicité était mieux ajusté par rapport au jugement humain, les avantages et désavantages théoriques mentionnés seraient alors perceptibles en pratique et l'approche d'échantillonnage sélectif serait alors généralement préférable aux autres approches. Toutefois, dans les conditions rencontrées lors de nos expérimentations, l'échantillonnage uniforme remporte les honneurs puisque cet algorithme produit des segmentations de même qualité que les autres en une fraction du temps de calcul.

Au niveau de la fusion des segmentations, nous avons présenté deux algorithmes différents : la fusion simple et la fusion par différence. Ces deux algorithmes sont basés sur un indice de périodicité global, indiquant dans quelle mesure un segment donné isole des parties cycliques de chaque signal d'un ensemble de signaux. L'idée générale de ces deux algorithmes est de construire un sous-ensemble de tous les segments compris dans l'ensemble des segmentations à fusionner. Ce sous-ensemble doit être tel qu'aucun segment ne chevauche temporellement un autre segment du sous-ensemble. La fusion simple procède en ajoutant un à un au sous-ensemble les segments en ordre décroissant d'indice de périodicité global, en éliminant au fur et à mesure les segments candidats

ayant un chevauchement temporel avec le dernier segment ajouté au sous-ensemble. La fusion par différence procède d'une façon similaire, mais en n'éliminant que la portion d'un segment qui est en chevauchement temporel. Le reste du segment, la différence, peut ensuite être considéré comme un segment candidat s'il est suffisamment cyclique.

De ces deux approches, nous avons défendu théoriquement que la fusion par différence est généralement préférable, en particulier lorsque la séquence à segmenter comporte des activités cycliques distinctes temporellement adjacentes. Dans ces situations, la fusion simple est propice à manquer des détections, et ce, à cause de segments ne se chevauchant que sur petit nombre de trames lors de la transition entre deux activités cycliques. Nous avons aussi noté l'avantage *involontaire* de l'approche de fusion simple qui est plus propice à se défaire d'une fausse détection obtenue à l'étape de la segmentation des signaux, lui attribuant l'avantage sur une de nos séquences de test. Nous avons toutefois préféré l'approche de fusion par différence étant donné son fondement théorique, alors que l'avantage de la fusion simple tient plutôt d'erreurs effectuées à l'étape de la segmentation des signaux.

Dans le cadre de cet ouvrage, nous avons également souligné l'intérêt de la segmentation anatomique afin de mieux pouvoir décrire les mouvements cycliques ainsi que pour permettre l'occurrence simultanée de mouvements cycliques distincts sur différentes parties du corps. Les faibles résultats obtenus et l'analyse que nous en avons faite nous portent à croire qu'il devient beaucoup plus difficile d'imiter une segmentation de référence humaine dans une situation de segmentation anatomique étant donné qu'elle semble empreinte d'une plus grande part de subjectivité. En contrepartie, nous avons illustré les avantages liés à l'omission d'une telle segmentation anatomique lorsque la topologie squelettique ne produit pas des mouvements anatomiquement analogues à ceux effectués dans la séquence vidéo originale ou lorsque les signaux comportent beaucoup de bruit et que notre approche nécessite une plus grande redondance.

## 6.1 Développements et travaux futurs

Dans l'ensemble, cet ouvrage propose une solution viable au problème, rarement abordé, de la segmentation temporelle d'activités cycliques. Au meilleur de nos connaissances, elle est également la seule solution de haut niveau proposée par la littérature pour ce problème. Étant une recherche exploratoire, notre recherche peut être considérée comme une réussite en ce sens qu'elle démontre la possibilité de succès d'une telle entreprise. Toutefois, comme toute recherche exploratoire, cette approche n'est pas sans faille.

Tout d'abord, au point de vue de l'implémentation de la segmentation des signaux, l'algorithme de l'approche exhaustive, qui conserve au moins un intérêt théorique, est plutôt lourd d'exécution. Plusieurs méthodes pourraient être étudiées pour améliorer cet aspect. Tout d'abord, étant donné que le calcul de l'indice de périodicité est fait de façon indépendante pour chaque segment analysé, il serait simple de transformer cette étape en processus parallèle. Il suffirait de diviser l'espace des segments possibles entre les processeurs disponibles. Il serait aussi possible d'envisager un algorithme dynamique faisant usage de la grande similarité dans le calcul de l'autocorrélation de deux segments voisins pour éviter de faire un même calcul nombre de fois.

Également, une étude plus approfondie du problème de la fusion des segmentations pourrait fortement profiter à notre approche. Bien qu'elle semble être de bon recours *a priori*, une méthode de fusion basée sur une approche *glouton* ne semble pas être la meilleure solution au problème de la fusion. En effet, si l'étape de segmentation des signaux laisse passer une fausse détection, les algorithmes de fusion présentés dans cet ouvrage sont presque assurés de laisser cette fausse détection dans la segmentation finale, sauf si elle est en chevauchement temporel avec une activité cyclique réelle. Il est certain que le problème des fausses détections devrait idéalement être contrôlé entièrement au niveau de la segmentation des signaux, mais puisque nous sommes appelés à travailler avec des signaux imparfaits, l'occurrence d'erreurs à cette étape est presque inévitable. Un algorithme de fusion plus informé pourrait ajouter un niveau de sécurité à cette étape.

Une partie importante des problèmes rencontrés avec nos algorithmes pourrait aussi être résolue par l'usage de séquences de squelettes plus *coopératives*. Tel que nous l'avons signalé lors de l'analyse de nos expérimentations, aucune des deux méthodes d'extraction de séquences de squelettes utilisées n'est adéquate pour notre tâche. En effet, parmi les séquences de squelettes utilisées, plusieurs sont très difficiles à traiter : soit elles comportent une quantité trop importante de bruit, soit le mouvement du squelette ne représente pas le mouvement d'origine. Certaines séquences combinent même un niveau important de bruit avec des mouvements non représentatifs. Malgré ces conditions adverses, nos algorithmes obtiennent tout de même des résultats intéressants sur plusieurs de ces séquences. Toutefois, il serait intéressant d'étudier la performance et la fiabilité de ces algorithmes sur des séquences montrant des mouvements représentatifs et ne comportant que peu de bruit. De telles séquences pourraient être obtenues par un système de capture de mouvements, mais cela viendrait poser de sévères contraintes sur les conditions d'utilisation de notre approche. Des recherches en vision numérique sur le sujet de l'estimation de la pose d'un objet articulé à partir d'images ou des séquences vidéo pourraient donc être fortement bénéfiques à notre approche.

De plus, il est intéressant de poser l'hypothèse que certains pointages moins intéressants soient simplement dus à la méthode d'évaluation des résultats. En effet, l'indice de similarité  $I_S$  est généralement plus sévère que le serait un humain lors de l'évaluation de la qualité d'une segmentation. À titre d'exemple, la segmentation automatique présentée à la figure 5.8 est pratiquement identique à la segmentation de référence et elle obtient un indice de similarité de 85%. Toutefois, selon les standards d'évaluation auxquels nous sommes habitués, un pointage de 85% indique une erreur significative. Il serait alors intéressant de pouvoir établir une mesure d'évaluation des segmentations qui soit plus représentative de l'appréciation qu'un humain aurait de la solution donnée.

Il serait aussi intéressant d'étudier comment notre méthode peut se généraliser à d'autres objets articulés. Théoriquement, notre méthode permet l'étude de tout mouvement articulé, peu importe l'objet qui en est la source. Toutefois, étant donné les algorithmes d'extraction de squelettes auxquels nous avons accès, nous n'avons pu tester d'autres objets articulés qu'un humain. Il en va de même pour un squelette en trois dimensions : bien que rien ne l'empêche dans notre approche, nous n'avons pas eu les ressources nécessaires pour le tester empiriquement.

Finalement, une recherche pourrait aussi étudier la possibilité d'appliquer cette méthode dans un système temps réel. Puisqu'elle analyse un signal dans sa totalité, notre approche ne s'adapte pas naturellement à ce type de tâche. Il est toutefois possible d'envisager des modifications pour en permettre une utilisation plus interactive. À titre d'exemple, il serait possible de former les signaux et d'en segmenter les pauses en temps réel, puisque cet algorithme parcourt toujours le signal en une seule passe, du début à la fin. Lorsqu'une pause serait détectée, un segment pourrait être formé avec la partie de signal précédant cette pause et être soumis à une segmentation temporelle d'activités cycliques. Pour qu'une telle approche soit réellement pratique, il faudrait d'abord pouvoir extraire la séquence de squelettes en temps réel.

À sa conclusion, cette recherche semble en susciter un nombre d'autres. Toutefois, elle aura au moins établi qu'il est possible d'utiliser une approche de haut niveau pour effectuer rapidement une segmentation temporelle d'activités cycliques d'une qualité acceptable. Elle aura également fourni des bases solides pour améliorer ce type d'approches.

# Bibliographie

- [1] Jezekiel Ben-Arie, Zhiqian Wang, Purvin Pandit, and Shyamsundar Rajaram. Human activity recognition using multidimensional indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8) :1091–1104, 2002.
- [2] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(3) :257–267, 2001.
- [3] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8) :781–796, 2000.
- [4] A. Divakaran, K.A. Peker, R. Radharkishnan, Z. Xiong, and R. Cabasson. *Video Mining*, chapter Video Summarization Using MPEG-7 Motion Activity and Audio Descriptors. Kluwer Academic Publishers, 2003.
- [5] Ismail Haritaoglu, Davis Harwood, and Larry S. David. W4 : Real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8) :809–830, 2000.
- [6] Frédéric Jean, Robert Bergevin, and Alexandra Branzan-Albu. Body tracking in human walk from monocular video sequences. In *Second Canadian Conference on Computer and Robot Vision (CRV)*, pages 144–151, Victoria, BC, May 9-11 2005.
- [7] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang. Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recogn. Lett.*, 18(5) :433–443, 1997.
- [8] Junghye Min and Rangachar Kasturi. Activity recognition based on multiple motion trajectories. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 199–202, Washington, DC, USA, 2004. IEEE Computer Society.
- [9] Ramprasad Polana and Randal Nelson. Detecting activities. In *DARPA93*, pages 569–574, 1993.



- [10] Ramprasad Polana and Randal Nelson. Low level recognition of human motion. In *IEEE Computer Science Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, Austin, TX, USA, 1994.
- [11] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1988.
- [12] Steven M. Seitz and Charles R. Dyer. View-invariant analysis of cyclic motion. *Int. J. Comput. Vision*, 25(3) :231–251, 1997.
- [13] Aravind Sundaresan, Amit RoyChowdhury, and Rama Chellappa. A hidden markov model based framework for recognition of humans from gait sequences. In *2003 International Conference on Image Processing. Proceedings.*, 2003.
- [14] Jérôme Vignola. Modélisation d’une silhouette humaine par un squelette et ses parties. Master’s thesis, Université Laval, 2003.
- [15] Mehran Yazdi, Alexandra Branzan Albu, and Robert Bergevin. Morphological analysis of spatio-temporal patterns for the segmentation of cyclic human activities. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 240–243, Washington, DC, USA, 2004. IEEE Computer Society.
- [16] Deniz Yuret. From genetic algorithms to efficient optimization. Technical report, Cambridge, MA, USA, 1994.

# Annexe A

## Logiciel Développé

Cette annexe présente brièvement l'interface utilisateur que nous avons développée dans le cadre de nos recherches afin de nous permettre de faire nos diverses expérimentations rapidement et de bien visualiser nos résultats.

La figure A.1 présente le menu principal. De là, l'utilisateur a les options suivantes :

- procéder à une segmentation temporelle étape par étape, avec ou sans segmentation anatomique ;
- procéder à un lot de segmentations temporelles, avec ou sans segmentation anatomique ;
- ouvrir la page d'analyse d'indices de périodicité ;
- quitter le logiciel.

La page d'analyse d'indices de périodicité, illustrée à la figure A.2, était surtout utile lors de l'élaboration de l'indice de périodicité défini dans la section 3.2 de cet ouvrage. Il peut toutefois aider, par la visualisation, à mieux en comprendre le mécanisme. Le bouton *Signal Selection* mène l'utilisateur à la page de sélection de signaux, illustrée par la figure A.3, où il peut choisir le signal à analyser en naviguant parmi les différentes séquences de squelettes et les différents algorithmes de génération de signaux. L'utilisateur est ensuite invité à choisir l'algorithme d'indice de périodicité à analyser. Les deux marqueurs entourant le tracé du signal permettent de définir un segment à analyser. La fenêtre inférieure gauche illustre le calcul de l'indice de périodicité pour ce segment. Dans le cas illustré, nous pouvons y voir le tracé de l'autocorrélation du segment choisi, la position des maxima observés, celle des maxima attendus et une zone ombragée représentant le dernier 10% de la trace de l'autocorrélation. La fenêtre inférieure droite présente la valeur relative de l'indice calculée sur chaque segment possible du signal. Les deux axes de cette matrice sont l'indice de début et l'indice de fin du segment. Les lignes bleues représentent les indices de début et de fin du segment actuel

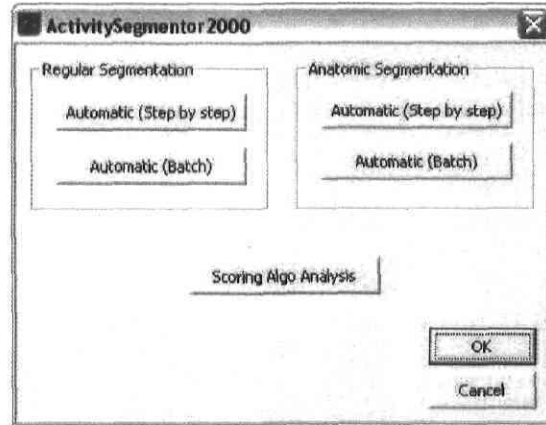


FIG. A.1 – Menu principal

et leur croisement représente la position de ce segment dans la matrice. La fenêtre supérieure droite illustre le tracé des valeurs de l'indice de périodicité pour tous les segments possibles ayant l'indice de début du segment actuel, alors que la fenêtre immédiatement en dessous fait de même pour les segments ayant le même indice de fin. Dans ces deux fenêtres, la ligne bleue positionne le segment actuel sur ce tracé.

Lorsque l'utilisateur choisit d'effectuer une segmentation temporelle étape par étape, avec ou sans segmentation temporelle, il est d'abord invité à choisir la séquence de squelettes à segmenter. Tel qu'illustré à la figure A.4, l'interface permet à l'utilisateur de visualiser la séquence choisie et, au besoin, d'appliquer un filtre moyennant à la séquence de façon à diminuer les effets du bruit qui pourrait s'y trouver.

L'utilisateur est ensuite invité à choisir l'algorithme de recherche de meilleur segment à utiliser pour la segmentation des signaux. Illustrée par la figure A.5, l'interface permet de visualiser le résultat de la segmentation et de la refaire la segmentation au besoin.

Finalement, tel qu'illustré par la figure A.6, l'utilisateur est invité à choisir l'algorithme de fusion à utiliser. À partir de cette page, il peut visualiser les résultats de la segmentation. Dans le cas incluant la segmentation anatomique, les activités cycliques sont représentées au niveau de chaque partie du corps en les traçant en couleur, le reste du squelette étant tracé en gris. Nous alternons entre le bleu et le rouge pour aider à distinguer deux activités cycliques temporellement adjacentes. Dans le cas omettant la segmentation anatomique, un cadre de couleur apparaît autour de l'image quand une activité cyclique a lieu. À ce moment, nous alternons également entre le bleu et le rouge pour distinguer deux activités cycliques temporellement adjacentes. La page permet aussi de sélectionner une segmentation de référence et de voir l'indice de similarité

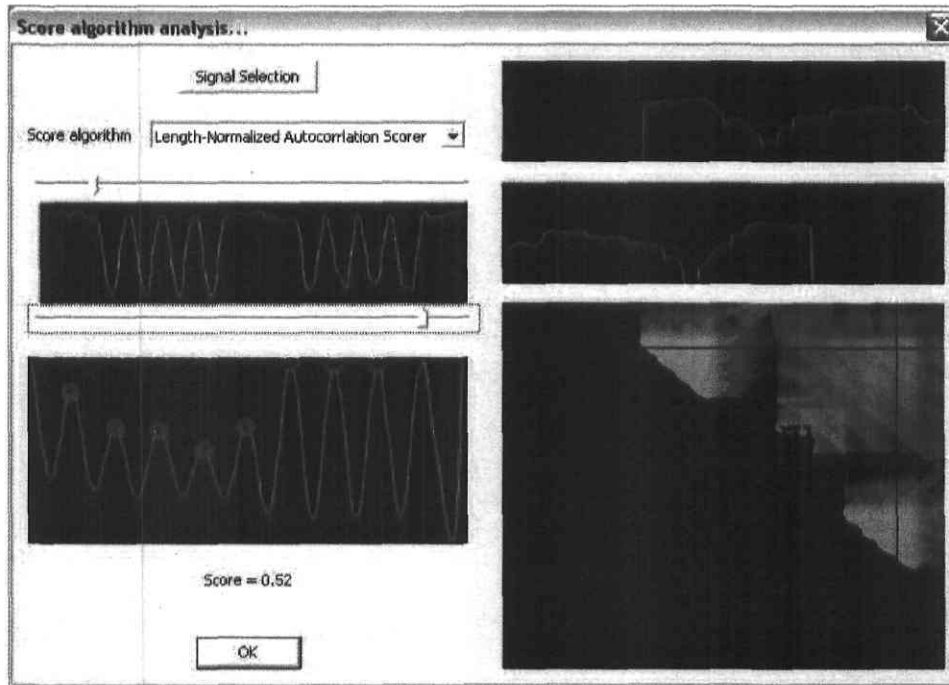


FIG. A.2 – Page d'analyse d'indices de périodicité

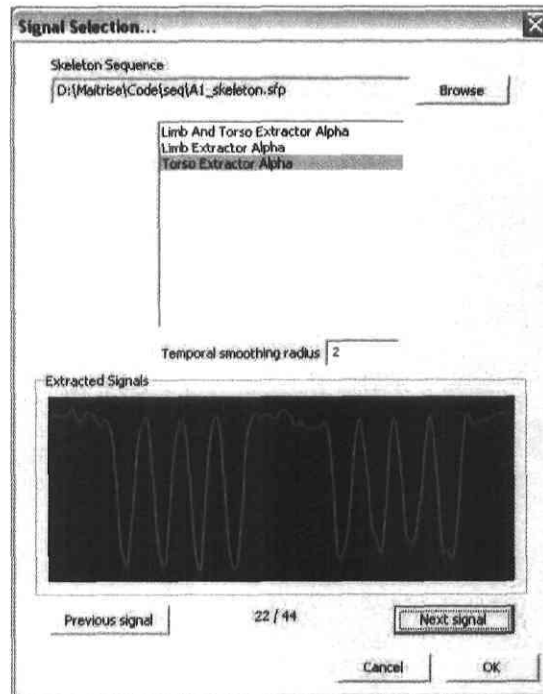


FIG. A.3 – Page de sélection de signal

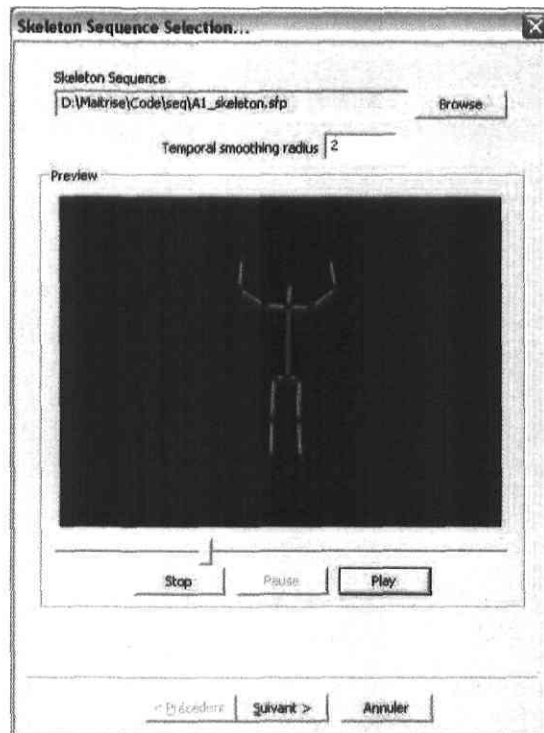


FIG. A.4 – Étape par étape - Page de sélection de la séquence de squelettes

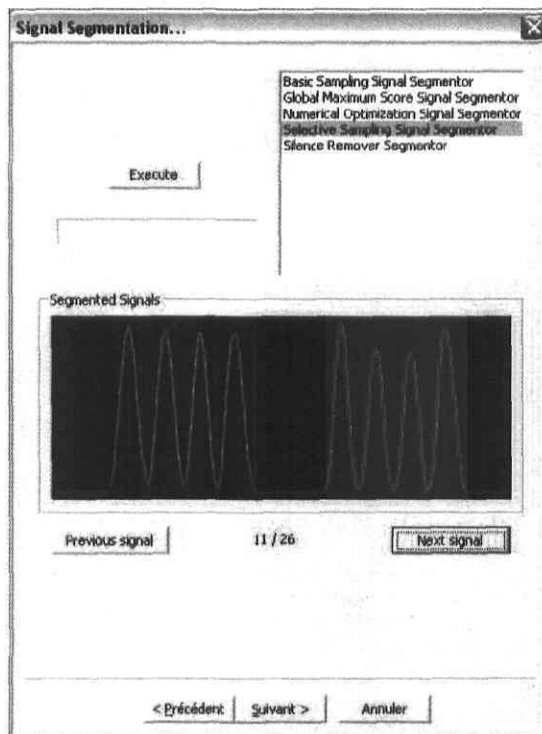


FIG. A.5 – Étape par étape - Page de segmentation des signaux

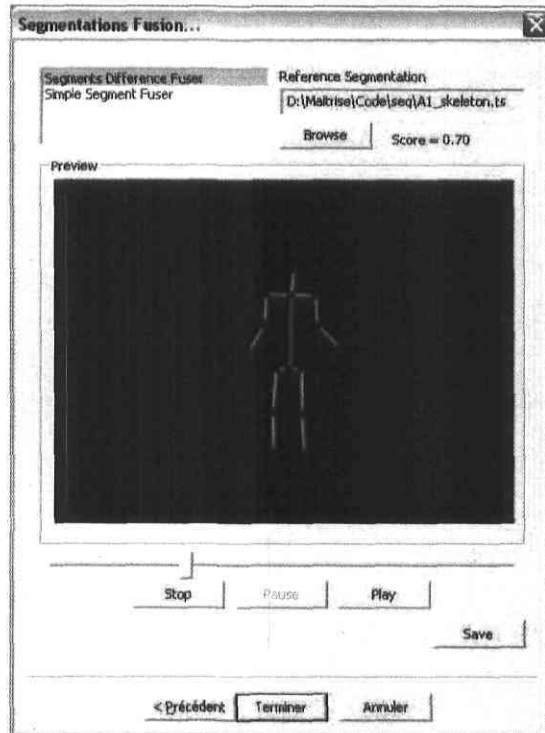


FIG. A.6 – Étape par étape - Page de fusion

obtenu. Finalement, cette page permet de sauvegarder dans un fichier la segmentation temporelle obtenue.

Lorsque l'utilisateur choisit de procéder à une segmentation de lot, avec ou sans segmentation anatomique, il se retrouve à la page illustrée à la figure A.7. Il peut alors choisir toutes les séquences, les algorithmes de génération de signaux, les algorithmes de segmentation de signaux et les algorithmes de fusion qu'il désire tester. Toutes les combinaisons de ces algorithmes seront testées contre toutes les séquences choisies. L'utilisateur peut également définir un rayon de filtre moyennant à appliquer sur la séquence et sur les signaux pour amoindrir les effets du bruit possiblement présent dans les séquences. L'utilisateur peut finalement choisir un fichier où les résultats de chaque segmentation temporelle seront enregistrés.

À tout moment dans ce logiciel, lorsque l'utilisateur doit choisir un algorithme de fusion ou de recherche du meilleur segment, la fenêtre illustrée à la figure A.8 lui est présentée. Cette fenêtre créée dynamiquement permet à l'utilisateur d'ajuster les divers paramètres requis par l'algorithme qu'il vient de choisir.

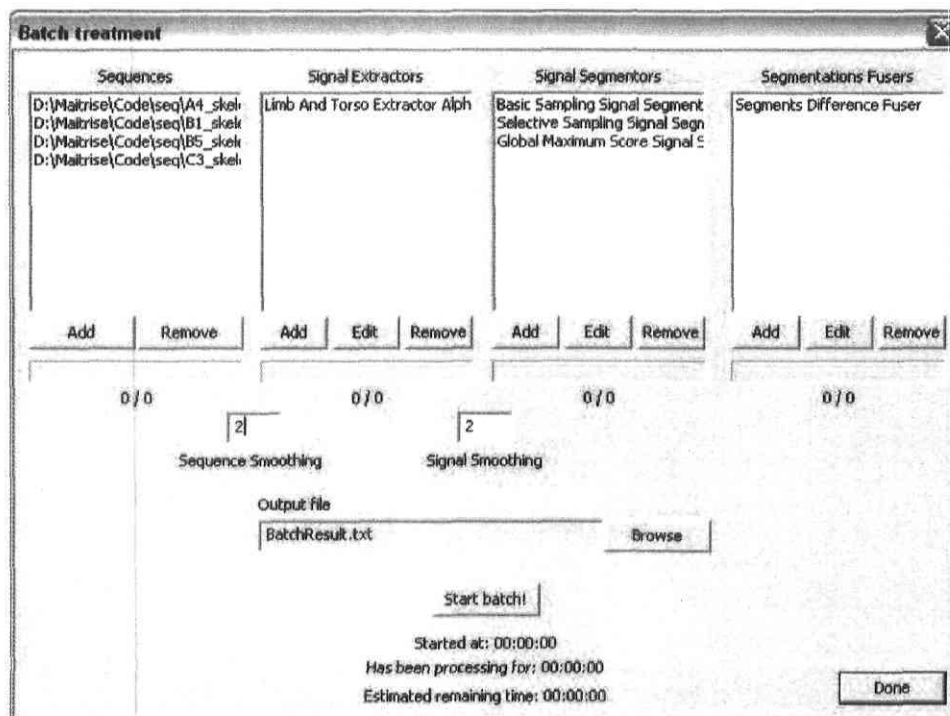


FIG. A.7 – Page de traitements de lot

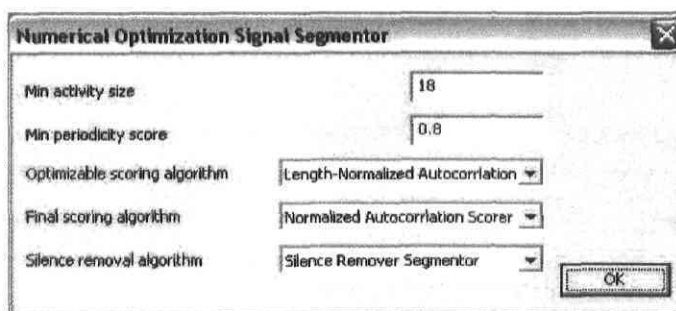


FIG. A.8 – Page dynamique d'ajustement des paramètres d'un algorithme