Modélisation efficiente du corps humain en mouvement

Clément Lemeunier

Université Claude Bernard Lyon 1

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de :

l'Université Claude Bernard Lyon 1

Ecole Doctorale 512
Infomaths

Spécialité de doctorat : Informatique

Soutenue publiquement le 18 décembre 2023, par :

Clément Lemeunier

Modélisation efficiente du corps humain en mouvement

Utilisation de méthodes d'analyse spectrale et d'apprentissage profond pour le traitement de surfaces dynamiques

---

Devant le jury composé de :

Florent Dupont Professeur, Université Lyon 1	**Directeur de thèse**
Franck Hetroy-Wheeler Professeur, Université de Strasbourg	**Examinateur**
Damien Rohmer Professeur, Ecole polytechnique (LIX)	**Rapporteur**
Stefanie Wuhrer CR, INRIA	**Rapporteure**
Florence Zara Maître de Conférences, Université Lyon 1	**Examinatrice**
Mohamed Daoudi Professeur, IMT Nord Europe	**Invité**
Florence Denis Maître de conférences, Université Lyon 1	**Invitée**
Guillaume Lavoué Professeur, Ecole Centrale de Lyon	**Invité**

Résumé

Développer des modèles capables de comprendre la dynamique du corps humain représente aujourd'hui un défi important dans le domaine de l'informatique graphique. Les évolutions récentes dans le domaine de la capture de mouvement permettent de générer des bases de données composées de modèles 4D de surfaces incluant l'information détaillée de l'apparence et du mouvement des sujets capturés. Aujourd'hui, les architectures d'apprentissage profond capables de comprendre et de générer du texte ou des images sont courantes et ont été introduites au grand public. Néanmoins, transférer les méthodes utilisées dans ces architectures à des données générées par les captures de mouvement n'est pas évident en raison de la différence de structure. En effet, le texte ou les images ont une structure régulière ou euclidienne, alors que les données générées par les captures de mouvement sont de types nuages de points ou maillages triangulaires qui n'ont plus cette structure régulière et qui résident plutôt dans un domaine non-euclidien.

L'objectif du travail de ce doctorat est premièrement de s'intéresser aux travaux de l'état de l'art qui permettent l'application de méthodes utilisées en apprentissage profond à des données résidant dans un domaine plus complexe et de plus haute dimension. Cette recherche de transfert de méthodes tombe dans le champ d'étude récemment introduit et nommé apprentissage profond géométrique (*Geometric Deep Learning*). De manière générale, les travaux s'intéressant à ce type de données ne traitent l'information surfacique que de manière statique et ne prennent pas en compte l'information dynamique du mouvement. L'intention principale est de résoudre cette problématique en combinant des techniques d'analyse de surfaces statiques à des méthodes exploitant la dimension temporelle présente dans la capture de corps humains. Le principal défi de la thèse est donc de coupler apprentissage profond, analyse de surfaces et analyse temporelle en orientant les recherches vers un aspect génératif.

Le deuxième objectif est d'explorer les méthodes d'analyse spectrale de données 3D permettant la transformation de surfaces dans le domaine des fréquences. Les structures créées par cette analyse spectrale seront étudiées afin de comprendre comment elles peuvent être exploitées pour résoudre la problématique principale. La seconde intention est donc de montrer qu'il est possible de développer des modèles d'apprentissage profond génératifs appliqués à des données 4D en utilisant seulement l'information contenue dans le domaine spectral, la transformation de surfaces dynamiques dans le domaine des fréquences permettant de s'affranchir des contraintes

introduites par la non-régularité de leur structure.

Les travaux présentés dans cette thèse représentent deux contributions. D'abord, une approche est proposée pour le traitement de données issues de captures de mouvement sans prendre en compte l'information temporelle. Cette première contribution concerne l'exploitation du traitement spectral de surfaces couplé à des techniques d'apprentissage profond, permettant le développement d'un modèle capable de créer une représentation adaptée au corps humain. Ensuite, une deuxième contribution permet la prise en compte de l'information dynamique en exploitant la première contribution couplée à une architecture capable de comprendre le contexte de séquences de données. Nos méthodes produisent des résultats compétitifs avec l'état de l'art et permettent d'ouvrir la voie vers une nouvelle manière de traiter ce type de données qui résident dans le domaine de la 3D et de la 4D, un défi actuellement important dans la littérature.

Mots clés : maillages triangulaires, analyse spectrale, apprentissage profond, analyse temporelle

Abstract

Developing models capable of understanding the dynamics of the human body represents today a significant challenge in the field of computer graphics. Recent developments in the field of motion capture make it possible to generate databases composed of 4D surface models including detailed information on the appearance and movement of the captured subjects. Today, deep learning architectures able to understand and generate text or images are common and have been introduced to the general public. However, transferring the methods used in these architectures to data generated by motion capture is not easy due to the difference in structure. Indeed, the text or images have a regular or Euclidean structure, while the data generated by motion capture are represented by point clouds or triangular meshes which no longer have this regular structure and which reside rather in a non-Euclidean domain.

The objective of this thesis is firstly to focus on state-of-the-art works which allow the application of methods used in deep learning to data residing in a more complex and higher dimensional domain. This method transfer research falls into the recently introduced field of study named *Geometric Deep Learning*. Generally speaking, the works interested in this type of data only process the surface information in a static manner and do not take into account the dynamic information of the movement. The main intention is to resolve this problem by combining static surface analysis techniques with methods exploiting the temporal dimension present in the capture of human bodies. The main challenge of the thesis is therefore to combine deep learning, surface analysis and temporal analysis by directing research towards a generative aspect.

The second objective is to explore methods for spectral analysis of 3D data allowing the transformation of surfaces in the frequency domain. The structures created by this spectral analysis will be studied in order to understand how they can be exploited to solve the main problem. The second intention is therefore to show that it is possible to develop generative deep learning models applied to 4D data using only the information contained in the spectral domain, the transformation of dynamic surfaces in the frequency domain making it possible to free themselves from the constraints introduced by the non-regularity of their structure.

The work presented in this thesis represents two contributions. First, an approach is proposed for processing data from motion capture without taking into account temporal information. This first contribution concerns the exploitation of spectral

mesh processing coupled with deep learning techniques, allowing the development of a model capable of creating a representation adapted to the human body. Then, a second contribution allows dynamic information to be taken into account by exploiting the first contribution coupled with an architecture capable of understanding the context of data sequences. Our methods produce results competitive with the state of the art and provide access to a new way of processing this type of data which resides in the domain of 3D and 4D, a currently important challenge in the literature.

Keywords : triangular meshes, spectral analysis, deep learning, temporal analysis

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

L'apparition récente de modèles capables de générer du texte ou des images sous contrôle de l'utilisateur (tel que GPT-3 [19] et DALL-E [113]) a été possible grâce à l'introduction d'architectures de réseaux de neurones massivement utilisées ces dix dernières années. Le bon fonctionnement des ces modèles peut suggérer l'idée de voir se démocratiser dans le futur des modèles similaires travaillant sur des objets à 3 ou 4 dimensions. Les avancées dans le domaine de la reconstruction de formes 3D depuis des images ont récemment permis un accès facilité aux systèmes de capture de mouvement, faisant apparaître des bases de données d'objets 4D de plus en plus denses et pouvant servir de base d'apprentissage à ces futurs modèles. Néanmoins, ces captures génèrent des représentations telles que des nuages de points ou des maillages triangulaires qui n'ont plus la structure régulière qu'ont le texte ou les images. Un nouveau champ d'étude, l'apprentissage profond géométrique (*Geometric Deep Learning*), a dernièrement vu le jour et vise à transférer les outils destinés à des données résidant dans un domaine euclidien à ces représentations résidant dans un domaine à plus haute dimension et ayant une structure non-euclidienne et non régulière. Bien qu'un grand nombre d'architectures aient été introduites ces dernières années proposant des solutions à ce problème, on peut sentir que leur habilité à comprendre les données fournies est moins élevée que celle des algorithmes travaillant sur du texte ou des images. Aussi, le temps d'entraînement de ces nouvelles architectures est bien plus grand, alors que les modèles présentés aujourd'hui au grand public sont déjà extrêmement énergivores. Il est donc aujourd'hui nécessaire de développer des méthodes qui permettront dans le futur de voir apparaître de tels algorithmes capables de comprendre ce nouveau type de données de manière efficace et économe en énergie.

Dans ce chapitre, nous verrons le contexte dans lequel s'est déroulé la thèse : premièrement, la situation globale de l'état de l'art concernant l'apprentissage profond sera abordé. Ensuite, le projet national dont notre équipe fait partie et ses objectifs seront présentés. Enfin, nous verrons comment se matérialisent les principaux défis de la thèse.

1.1 Contexte

L'apprentissage profond (*deep learning*) est une branche de l'intelligence artificielle qui est devenu un outil indispensable, offrant des avancées significatives dans divers domaines tels que le traitement du langage naturel ou encore la vision par ordinateur. Ces avancées ont été principalement facilitées par la disponibilité de vastes ensembles de données 1D telles que du texte, de l'audio ou des données temporelles et de données 2D telles que des images, ainsi que par les progrès dans la conception de réseaux de neurones profonds. L'essor de ce domaine a également été stimulé par l'amélioration des algorithmes d'optimisation, la diversification des architectures de réseaux de neurones, et la montée en puissance des cartes graphiques qui ont considérablement accéléré les calculs nécessaires à l'entraînement de ces réseaux. En conséquence, l'apprentissage profond a ouvert de nouvelles possibilités dans la résolution de problèmes complexes, conduisant à des avancées majeures dans des domaines allant de la reconnaissance vocale à la conduite autonome. Il continue de susciter un intérêt croissant et de révolutionner notre compréhension de l'intelligence artificielle.

À l'heure actuelle, les techniques d'apprentissage profond ont atteint un niveau de maturité certain pour les données 1D et 2D, permettant des applications telles que la classification, la détection ou encore la génération de contenu. Nous commencerons par présenter les concepts et architectures de ces méthodes. Ensuite, nous verrons qu'une nouvelle frontière émerge avec la nécessité de généraliser ces approches aux données 3D. La vision de l'état de l'art concernant l'apprentissage profond sera abordé dans ce chapitre de manière théorique alors que les chapitres suivants sont destinés à une approche pratique.

1.1.1 Vue d'ensemble des méthodes, architectures et modèles d'apprentissage

Les concepts d'apprentissage automatique (*machine learning*) et profond (*deep learning*) sont présents depuis longtemps dans la littérature (leurs origines peuvent être tracées jusqu'en 1943), mais c'est ces dix dernières années, après le succès important d'AlexNet [69], que ces technologies se sont révélées révolutionnaires grâce à des avancées concernant le *Big Data* et la puissance de calcul des ordinateurs. Ces deux concepts que sont l'apprentissage automatique et profond représentent deux domaines qui ont été largement adoptés. Ils concernent l'usage d'algorithmes pour apprendre depuis des données et ont pour but d'améliorer les prédictions ou les générations effectuées par ces modèles. En général, l'apprentissage profond utilise des bases de données très denses pour entraîner des réseaux de neurones afin qu'ils détectent automatiquement les caractéristiques des échantillons étudiés alors que l'apprentissage automatique utilise des méthodes de statistiques et requiert davantage d'intervention humaine. Ici, nous nous concentrerons sur la notion d'apprentissage profond génératif [25].

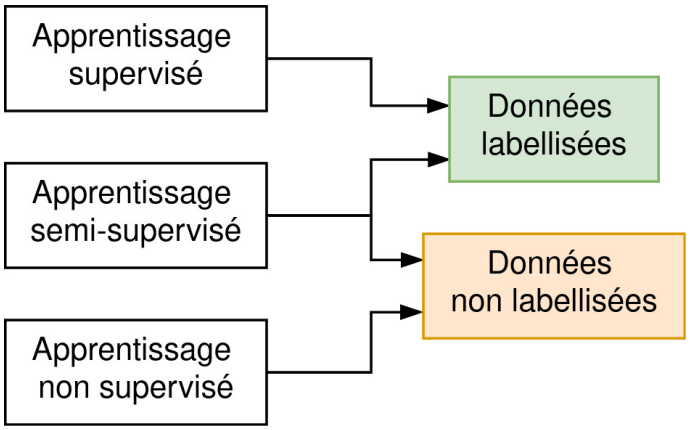


FIGURE 1.1 – Les différentes méthodes d'apprentissage.

Méthodes d'apprentissage

L'apprentissage d'un réseau de neurones peut être réalisé selon différentes méthodes, selon le type de données dont on dispose et la nature de la tâche à accomplir. Les trois principales approches sont l'apprentissage supervisé, non supervisé et semi-supervisé (voir figure 1.1). Chacune de ces méthodes possède ses particularités et trouve ses applications spécifiques.

Apprentissage supervisé. L'apprentissage supervisé repose sur l'utilisation d'un ensemble de données d'entraînement étiquetées, c'est-à-dire que chaque exemple d'entrée est associé à la sortie attendue ou à la classe à laquelle il appartient. L'objectif du réseau de neurones est d'apprendre à partir de ces exemples afin de généraliser les relations entre les entrées et les sorties, de sorte qu'il puisse produire des prédictions précises sur de nouvelles données non vues. L'apprentissage supervisé est largement utilisé dans des tâches de classification et de régression.

Apprentissage non supervisé. L'apprentissage non supervisé s'effectue sans données étiquetées. Le réseau de neurones doit découvrir les structures et les relations cachées dans les données sans aucune annotation externe. Les algorithmes d'apprentissage non supervisé sont principalement utilisés pour effectuer des tâches de regroupement (*clustering*) afin de segmenter les données en différents groupes homogènes. Cette approche permet d'identifier des motifs, des tendances ou des anomalies dans les données, facilitant ainsi la compréhension de celles-ci et ouvrant la voie à de nouvelles applications, comme de la génération de contenu. Les données traitées durant cette thèse étant principalement non annotées, et l'objectif étant de générer de nouveaux échantillons réalistes, les recherches présentées dans ce manuscrit se sont dirigées vers ce type d'apprentissage.

Apprentissage semi-supervisé. L'apprentissage semi-supervisé est une approche qui combine les aspects de l'apprentissage supervisé et non supervisé. Ici, l'ensemble de données d'entraînement comprend à la fois des exemples étiquetés et non étiquetés. L'objectif est de tirer parti de l'information contenue dans les données possédant

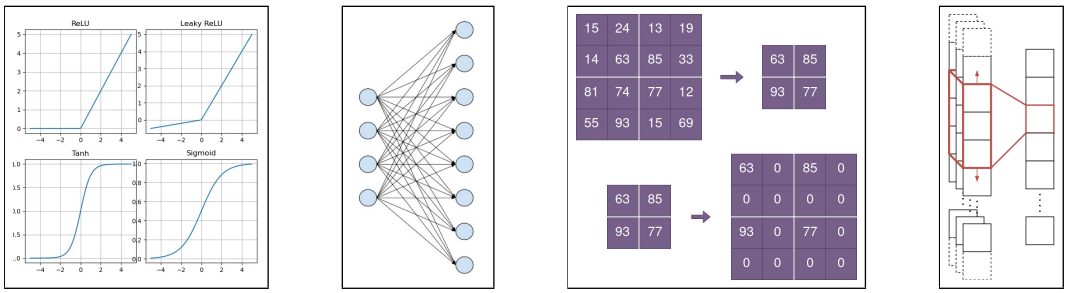


FIGURE 1.2 – Illustrations du fonctionnement de différentes couches de base. De gauche à droite : fonctions d'activation, couche entièrement connectée, couches de sous/sur-échantillonnage, couche de convolution.

une annotation tout en exploitant la richesse des données non classées pour améliorer les performances du réseau de neurones. L'apprentissage semi-supervisé est souvent utilisé lorsque la labélisation d'un grand ensemble de données est coûteuse ou difficile car il permet d'obtenir de bonnes performances en utilisant un nombre relativement réduit d'exemples annotés.

Les méthodes d'apprentissage des réseaux de neurones offrent des perspectives pour résoudre une grande variété de problèmes différents. Selon la nature et la disponibilité des données ainsi que la complexité de la tâche, la méthode la plus appropriée sera sélectionnée. Pour construire un réseau de neurones, il faut néanmoins coupler ces méthodes d'apprentissage à des opérations effectuées sur les données, représentant les couches du réseau.

Couches

Les couches d'un réseau de neurones sont les composants fondamentaux qui permettent au réseau d'apprendre en effectuant des opérations spécifiques. Elles peuvent être catégorisées en plusieurs types différents (voir figure 1.2).

Couche d'activation. La couche d'activation introduit une non-linéarité dans le réseau. Des fonctions populaires incluent ReLU (*Rectified Linear Unit*), Leaky ReLU, Tanh ou encore Sigmoid. Sans fonction d'activation, un réseau de neurones serait équivalent à une simple transformation linéaire, ce qui limiterait considérablement sa capacité à résoudre des problèmes complexes. L'ajout de non-linéarités confère au réseau une plus grande capacité d'apprentissage. Elles sont aussi cruciales pour le processus de rétropropagation (*backpropagation*) et permettent la transmission des gradients à travers le réseau afin d'ajuster les poids des neurones pendant l'apprentissage. L'introduction de fonctions d'activation contenant des paramètres apprenables a montré des améliorations concernant la qualité de l'entraînement. La recherche de nouvelles fonctions d'activation représente un domaine de recherche toujours ouvert [4].

Couche entièrement connectée. Cette couche est également connue sous le nom de couche dense (*fully connected*). Chaque neurone de cette couche est connecté

à tous les neurones de la couche précédente et de la couche suivante. C'est une couche simple qui doit être couplée à des couches d'activation pour former un réseau classique. Elle peut être assemblée dans un réseau en 2 couches seulement (*Single Layer Perceptron* ou SLP), ou plusieurs (*Multi Layer Perceptron* ou MLP). Bien que ces réseaux soient parmi les plus simples, leur étude est toujours d'actualité [44], et ils permettent encore aujourd'hui d'obtenir des résultats intéressants. Néanmoins, les réseaux construits de cette manière ont un grand nombre de paramètres et sont en général précédés de blocs de convolutions afin de réduire ce nombre.

Couche de sous/sur-échantillonnage. Les couches de sous/sur-échantillonnage (*pooling* et *upsampling*) sont des composants couramment utilisés dans les réseaux de neurones convolutifs. Elles sont utilisées pour identifier des caractéristiques à différentes résolutions. La couche de sous-échantillonnage fonctionne en divisant la région d'entrée en zones et en prenant la valeur maximale, minimale ou moyenne selon le type d'échantillonnage réalisé, la valeur maximale étant la plus courante. Il convient de noter que de l'information est perdue dans le processus, mais cette perte est comblée par la duplication en amont de l'information avec l'utilisation de plusieurs filtres en parallèle. Le sous-échantillonnage est souvent réalisé dans les architectures d'encodage. Le sur-échantillonnage est réalisé pour augmenter la résolution des données, plutôt dans les architectures de décodage. Cette couche prend une zone et la développe en répétant ou en interpolant les valeurs. Étant donnée la perte ou création d'information de ces couches, leur implémentation est liée au fonctionnement du réseau. Les choisir soigneusement est dépendant du problème et peut amener à des résultats améliorés [104, 151].

Couche de convolution. Cette couche est principalement utilisée pour extraire les caractéristiques des données en entrée en effectuant un filtrage par convolution. Le principe est de faire glisser une fenêtre représentant un filtre composé de paramètres apprenables sur des petites régions dans les premières couches du réseau. Afin d'obtenir des filtres sur des régions plus larges, la couche de convolution est couplée à des couches de sous-échantillonnage qui permettent de réduire la dimension des données tout en conservant l'information importante. Des blocs seront construits avec une couche de convolution, une couche de sous/sur-échantillonnage et une fonction d'activation, permettant de créer des réseaux qui auront moins de paramètres que ceux utilisant uniquement des couches entièrement connectées et ayant donc une meilleure capacité de généralisation. Ces réseaux convolutifs (*Convolutional Neural Networks* ou CNN) ont prouvé leur capacité d'apprentissage ces dix dernières années et sont aujourd'hui encore très utilisés pour des données 1D et 2D [67, 80].

Un réseau de neurones peu profond sera constitué d'une couche en entrée, une couche de sortie, et au plus une couche cachée. En revanche, un réseau de neurones profond est composé de plusieurs couches empilées les unes après les autres, complexifiant la fonction apprise par le réseau et donnant plusieurs niveaux de représentation des données. Couplées à un système de rétropropagation, les poids dont sont composés ces réseaux sont adaptés en fonction de l'objectif au cours de l'en-

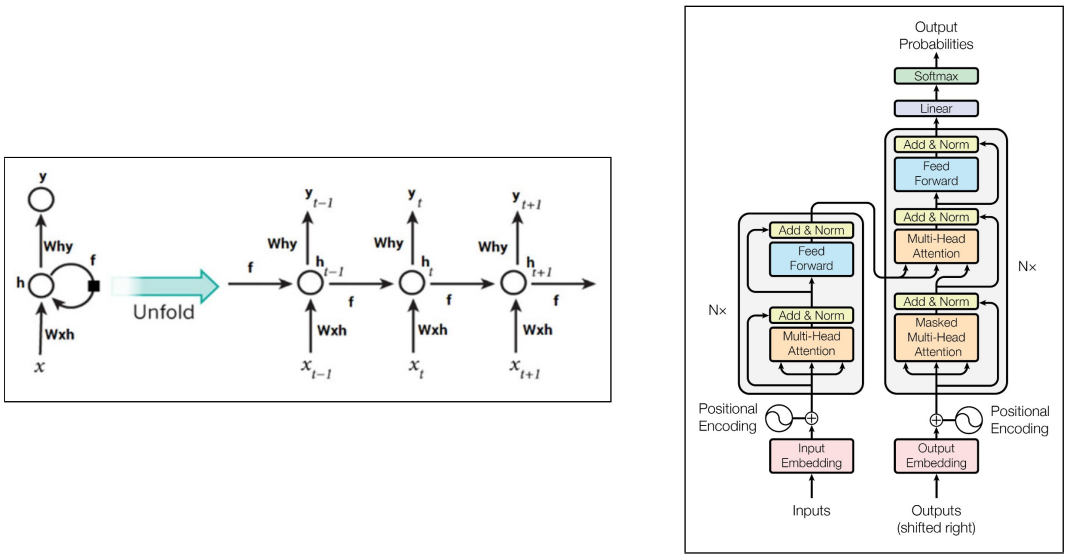


FIGURE 1.3 – À gauche, une architecture classique RNN [3]. À droite, l'architecture transformeur [135].

traînement. L'ordre dans lequel ces couches sont disposées peut varier en fonction de l'architecture du réseau et de la tâche à accomplir.

Architectures

En utilisant ces différentes couches, il est possible de créer des réseaux de différents types :

- entièrement connecté (assemblage de blocs de couches entièrement connectées et de fonctions d'activation)
- entièrement convolutif (assemblage de blocs de couche de convolution, de sous/sur-échantillonnage et d'activation, ou encore de blocs de couche de convolution réalisant elles-même les changement de dimensions, et de couches d'activation)
- convolutif et entièrement connecté (en général, un réseau entièrement connecté est ajouté en dernière partie d'un réseau convolutif)

En complément de ces réseaux "simples", il existe d'autres architectures ayant un fonctionnement plus complexe et spécifiquement dédiées à des données temporelles (voir figure 1.3).

Réseaux de neurones récurrents. Les réseaux de neurones récurrents (*Recurrent Neural Networks* ou RNNs), dont la version actuelle est basée sur les travaux d'Elman [47], sont conçus pour traiter chaque échantillon de données séquentielles étape par étape en prenant en compte l'information issue du traitement des échantillons précédents, ce qui leur permet de capturer des dépendances à long terme entre les éléments d'une séquence. Des variantes ont été introduites telles que *Long short-term memory* (LSTM) [60] ou *Gated recurrent unit* (GRU) [32]. Ces architectures restent

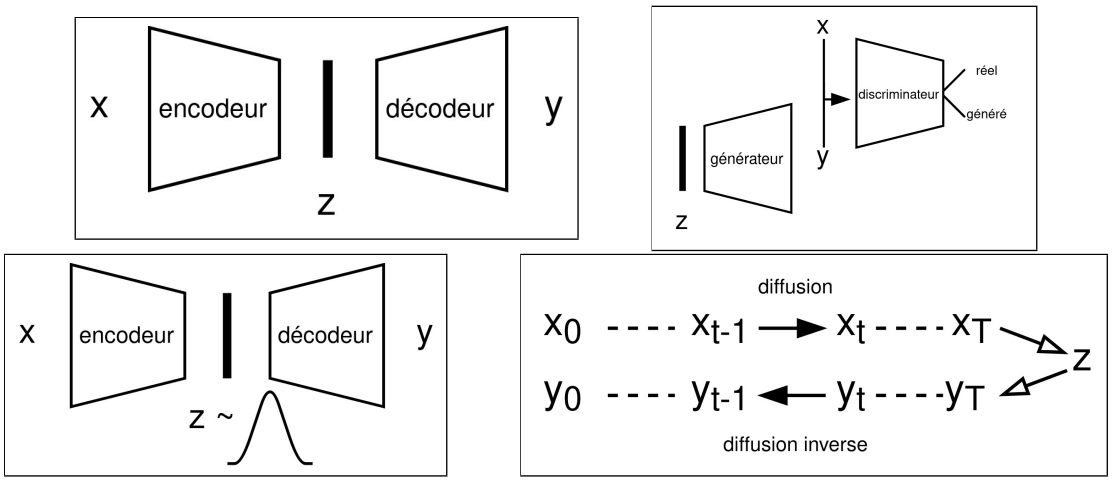


FIGURE 1.4 – Plusieurs modèles génératifs. x représente un échantillon réel, y un échantillon généré, et z une variable latente. En haut à gauche, un autoencodeur. En haut à droite, un GAN. En bas à gauche, un autoencodeur variationnel (VAE). En bas à droite, un modèle de diffusion.

populaires et de nombreux travaux les concernent encore aujourd'hui [125]. Ils ont été utilisés avec succès sur des données représentant du texte, des images ou des positions de jointures de squelettes. Cependant, les données sont traitées échantillon par échantillon et la parallélisation du traitement est difficile, rendant le processus lent et les dépendances à long terme calculées de mauvaises qualité. Pour résoudre ces problèmes, l'architecture transformeur a été introduite.

Transformeurs. Les transformeurs (*Transformers*) ont été introduits par Vaswani et al. [135] afin d'améliorer les résultats obtenus avec les RNNs. Ils sont non séquentiels, les séquences étant traitées en une passe plutôt qu'échantillon par échantillon, et utilisent la notion d'auto-attention (*self-attention*) qui permet de calculer des scores entre les différents échantillons. Cela les rend plus rapides que les RNNs (possibilité de calculs en parallèle) et plus aptes à comprendre les relations entre des échantillons très lointains. Cette architecture a été utilisée dans la deuxième contribution de cette thèse : plus d'informations sur son fonctionnement sont données dans la section 5.3.

Ces architectures peuvent être combinées pour construire des réseaux de neurones complexes, créant de nombreuses possiblités de configurations différentes. Il faut ensuite définir des modèles qui permettront d'entraîner ces architectures selon la tâche désirée (voir figure 1.4).

Modèles

Encodeur / classifieur / discriminateur. L'encodeur est un modèle qui permet en général de réduire la dimension des données en entrée. En utilisant une des architectures précédemment abordées, son but est de créer un vecteur latent à une

dimension et donc de compresser l'information. Si l'objectif est la classification, alors un score sera calculé depuis ce vecteur latent en fonction de l'étiquetage de la donnée en entrée. Ce classifieur peut aussi être nommé discriminateur.

Décodeur / générateur. À l'inverse, un décodeur ou générateur permet d'augmenter la dimension des données en entrée : depuis un vecteur latent, l'objectif est de décompresser l'information et de générer des données aux dimensions voulues.

Autoencodeur. Un autoencodeur est l'assemblage d'un encodeur et d'un décodeur. L'objectif de ce modèle est de compresser l'information en la faisant passer par un espace latent, ce qui a pour but de révéler les caractéristiques importantes (sous certaines conditions, voir chapitre 4) des données en entrée en ayant simplement pour objectif d'apprendre à minimiser la différence entre l'entrée et la sortie. Ce modèle est utilisé pour apprendre une compression avec perte efficace de manière non supervisée. Il existe plusieurs types d'autoencodeurs tels que ceux régularisés (*regularized*) ou ayant pour objectif le débruitage (*denoising*) [10].

Autoencodeur variationnel. L'autoencodeur variationnel (*Variational Autoencoder* ou VAE) est une extension de l'autoencodeur qui introduit une approche probabiliste dans la manière dont l'encodage est appris. Plutôt que d'entraîner le réseau à simplement reconstruire les valeurs en entrée, l'objectif est de minimiser une erreur supplémentaire, la divergence de Kullback-Leibler, permettant de construire un espace latent dont la distribution se rapproche d'une distribution normale. Cette propriété permet au VAE de générer de nouvelles données en échantillonnant aléatoirement dans l'espace latent. Nous verrons tout de même dans le chapitre 4 qu'un autoencodeur simple est aussi capable de générer des données réalistes sous certaines conditions. Le VAE peut présenter des problèmes d'entraînement tels qu'un problème d'équilibrage (la fonction de coût est représentée par deux termes différents et l'un peut prendre le dessus sur l'autre) ou d'éffondrement de variable (les éléments produits par le décodeur deviennent indépendants de la variable latente) [5].

Réseau antagoniste génératif. Un réseau antagoniste génératif (*Generative Adversarial Network* ou GAN) est un type de modèle génératif qui repose sur un concept de jeu entre deux réseaux de neurones : un générateur et un discriminateur. Le générateur prend un vecteur aléatoire en entrée et tente de générer des données réalistes. Le discriminateur prend en entrée des exemples de données réelles et générées, et tente de distinguer les vraies des fausses. Au fil de l'entraînement, le générateur s'améliore pour tromper de plus en plus le discriminateur, et le discriminateur s'améliore pour mieux distinguer les vraies données des fausses. Finalement, cela aboutit à un générateur capable de produire des données réalistes. Un GAN peut cependant être dûr à entraîner en raison d'un équilibre entre les deux réseaux difficile à atteindre (*Nash Equilibrium*) ou en raison d'effondrement des modes (*mode collapse* : le générateur ne produit qu'un seul exemple réaliste) [70].

Diffusion. Le modèle de diffusion est une approche de modélisation générative qui repose sur un processus de diffusion stochastique. Contrairement aux autres mo-

dèles génératifs qui créent des données à partir d'un bruit aléatoire, le modèle de diffusion utilise un processus itératif pour améliorer progressivement la qualité de l'échantillon produit. Il s'appuie sur le principe de diffusion d'une distribution de probabilité initiale pour générer progressivement une distribution finale correspondant aux données désirées. L'objectif est donc d'apprendre premièrement à bruiter un échantillon donné en entrée, puis de le débruiter, en général en utilisant un conditionnement. Ce modèle est très utilisé aujourd'hui pour traiter des données textuelles [156], temporelles [83] ou des images [39]. La génération est souvent conditionnée depuis du texte [152, 147].

Conditionné. Lorsqu'un de ces modèles génératifs est conditionné, cela signifie que le processus de génération des données est contrôlé ou guidé par une information supplémentaire appelée condition ou conditionnement. Cette information peut être sous la forme d'une étiquette, d'une catégorie, d'un texte, d'une image, ou tout autre type de données contextuelles. Le conditionnement rend les modèles génératifs beaucoup plus flexibles et utiles dans de nombreuses applications car il permet de contrôler le processus de génération et d'obtenir des résultats plus spécifiques et adaptés aux besoins de l'utilisateur.

Ces modèles ont tous des applications et des forces différentes en matière de génération de données, de compression et de représentation latente. Chacun a également ses propres avantages et inconvénients en fonction du problème spécifique qu'il vise à résoudre. Ils ont été largement appliqués à des données 1D et 2D. En revanche, en raison de la nature différente des données 3D, nous allons voir que ces modèles leur sont plus difficilement applicables.

1.1.2 Représentations de données 3D/4D

De nombreux domaines étudient des données qui ont une structure sous-jacente non-euclidienne, telles que les sciences sociales, l'imagerie médicale, la génétique, ou encore l'étude du mouvement du corps humain. Ces données géométriques sont larges et complexes mais sont des cibles naturelles pour les algorithmes précédemment présentés. Dans le domaine de l'infographie (*computer graphics*), les données 3D sont un atout précieux car elles fournissent des informations riches sur la géométrie complète des objets et des scènes détectés. Les données 3D issues des appareils de capture se présentent sous différentes formes qui varient dans leur structure et leurs propriétés. Elles peuvent être classées en deux grandes catégories : celle ayant une structure euclidienne, et celles ayant une structure non-euclidienne [1] (voir figure 1.5).

Structures euclidiennes

Certaines représentations 3D ont une structure sous-jacente euclidienne et préservent les propriétés du texte ou des images.

Projections de données 3D. Il est possible de projeter l'information contenue dans une représentation 3D vers une image. De telles projections représentent des

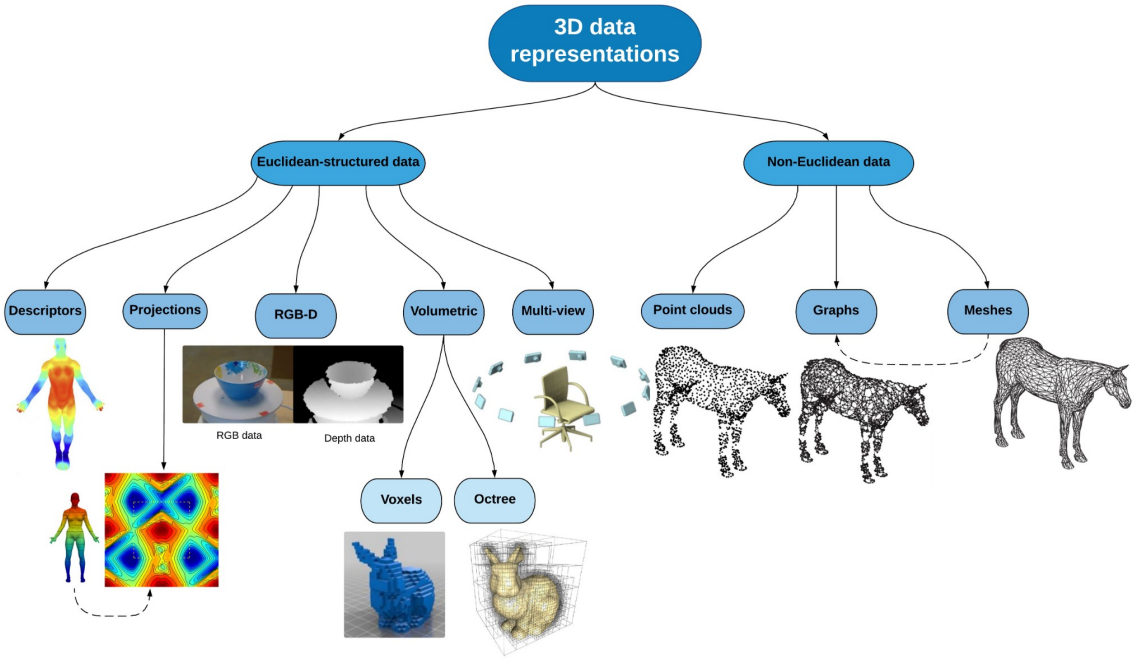


FIGURE 1.5 – Différentes représentations 3D. L'image provient de Ahmed et al. : *A survey on Deep Learning Advances on Different 3D Data Representations* [1].

données 3D en 2D tout en étant invariantes aux rotations autour de l'axe principal de la projection. Cela facilite le traitement de données 3D grâce à la structure régulière et euclidienne des projections résultantes et permet l'utilisation de modèles d'apprentissage classiques, mais ces représentations ne sont pas optimales puisque de l'information est perdue.

RGB-D. Il est aussi possible de représenter un objet 3D avec une image capturée depuis une seule caméra accompagnée de la profondeur (2.5D). Cette représentation est populaire grâce aux capteurs RGB-D fortement démocratisés comme la Kinect de Microsoft. Les bases de données créées sont beaucoup plus denses que celles représentant des objets 3D directement puisque plus facilement capturables, mais encore une fois ce type d'objet n'est pas optimal puisque de l'information est perdue dans le processus de capture.

Volumétriques. Parmi ce genre de données, on retrouve les voxels ou encore les octrees. Malgré la simplicité de ces représentations et leur capacité à coder des informations sur les formes 3D, elles souffrent de certaines limitations : elles permettent de transférer les méthodes d'apprentissage plus facilement mais ne préservent pas les propriétés intrinsèques des surfaces telle que le fait qu'elles soient lisses.

Multi-vues. Les données multi-vues sont représentées par des combinaisons d'images multiples depuis différents points de vue, et les réseaux travaillant sur des images leur sont directement applicables. En revanche, de nombreux points de vues sont nécessaires pour bien comprendre la structure de l'objet observé et de l'informa-

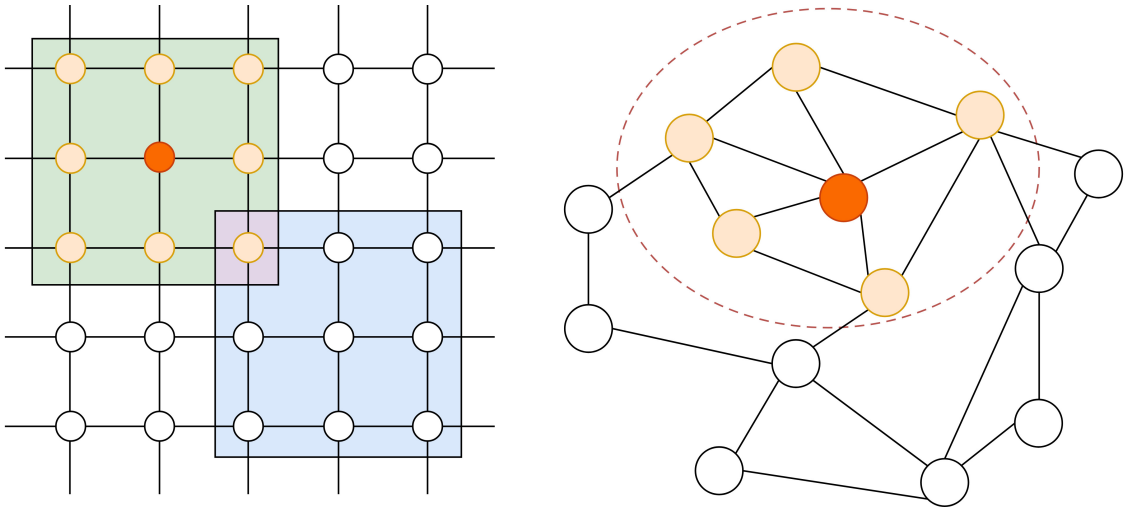


FIGURE 1.6 – Différence entre une structure euclidienne à gauche et non-euclidienne à droite. L'image provient de Liang et al. : *Survey of Graph Neural Networks and Applications* [81].

tion sera perdue si aucun algorithme de reconstruction de forme n'est utilisé afin de produire un objet 3D.

Descripteurs. Les descripteurs sont des représentations simplifiées d'objets 3D pour décrire la géométrie, la topologie, la surface ou encore la texture. Leur nature et signification dépendent du dit descripteur et ils sont souvent combinés à des modèles d'apprentissage profonds. Par exemple, un descripteur peut être représenté par l'équation de la chaleur sur une surface [126]. Plus d'informations sur des descripteurs spectraux sont donnés dans la section 3.3.

Ces différentes représentations, à l'exception des descripteurs qui seront examinés plus tard dans ce document, peuvent être efficaces et suffisantes pour décrire des objets rigides dont les déformations sont minimes. En revanche, afin de pouvoir décrire pleinement des objets déformables comme des corps humains, l'utilisation de représentations plus complexes est nécessaire.

Structures non-euclidiennes

Le deuxième type de représentations de données 3D concerne celles qui ont une structure sous-jacente non-euclidienne. Contrairement au texte ou aux images, ces représentations n'ont pas de paramétrisation globale ou un système de coordonnées commun, ce qui rend la distance entre deux noeuds les composant dépendante de la connectivité de la structure qu'elles approximent (voir figure 1.6). Les méthodes d'apprentissage classiques fonctionnant en 1D et 2D ne leur sont donc pas directement applicables.

Nuage de points 3D. Les nuages de points sont un ensemble désordonné de coordonnées spatiales qui approximent la géométrie d'un objet 3D. Les traiter est une

tâche compliquée en raison de leur manque de structure et leur manque de connectivité, ce qui peut mener à des ambiguités lors de leur analyse. Plus d'informations sur le traitement de ce type de représentation sont données dans la section 4.2.

Graphes et maillages 3D. Les graphes ou maillages 3D sont une des représentations les plus populaires, composée de l'information concernant la géométrie accompagnée d'une triangulation. Mais la géométrie n'ayant pas d'ordre, il est difficile d'appliquer des convolutions à ce genre de représentations. Aussi, il est possible d'utiliser des squelettes pour approximer certaines surfaces, qui sont similaires à des graphes. Plus d'informations sur le traitement de ce type de représentation sont données dans les sections 4.2 et 5.2.

Nuages de points, graphes et maillages 4D. Enfin, le type de données qui concerne l'objectif de cette thèse est représenté par des séquences de structures non-euclidiennes. Elles peuvent être composées d'éléments avec un nombre de noeuds et une connectivité variants (*time varying*) ou avec un nombre de noeuds et une connectivité constants (*temporally coherent*). Le traitement de ce genre de données sera abordé dans le chapitre 5.

L'utilisation de l'apprentissage profond gagne en popularité dans le domaine de la 3D. Cela implique d'utiliser les représentations riches disponibles tout en tenant compte de leurs propriétés difficiles. L'extension des modèles classiques à ce genre de données n'est pas simple en raison de leur nature géométrique complexe et des grandes variations structurelles résultant des différentes représentations. Ce domaine d'étude nommé apprentissage profond géométrique (*Geometric Deep Learning*) a récemment été introduit par Bronstein et al. [17].

1.1.3 Apprentissage profond géométrique

Récemment, avec l'apparition de bases de données d'objets 3D/4D et l'augmentation de la puissance des machines, il est devenu possible de considérer l'application des méthodes d'apprentissage profond destinées aux données 1D et 2D à des tâches appliquées à des données plus complexes. Plusieurs documents présentent un état de l'art concernant ce domaine émergent [17, 1, 145, 144, 24, 137, 29, 73, 33]. En fonction de la représentation utilisée, des challenges différents vont être abordés pour utiliser des architectures d'apprentissage profond existantes. La communauté de la vision par ordinateur s'est récemment penchée sur l'extension d'architectures aux données 3D. Certaines représentations conservent la structure régulière des données 2D et permettent l'extension directe des architectures connues. Ces représentations font partie des données 3D ayant une structure euclidienne. En revanche, d'autres types de données, comme les graphes, les nuages de points ou les maillages triangulaires, n'ont plus cette structure régulière et n'ont plus de paramétrisation globale. C'est une tâche difficile d'appliquer les architectures à ces données, alors que les comprendre grâce à des algorithmes d'apprentissage profond est un objectif aujourd'hui important.

Les difficultés lorsque l'on essaie de traiter des données 3D sont représentées par le manque d'échantillons disponibles, le manque d'étiquetage, leur nature géométrique, le fait qu'elles soient représentées de différentes manières et qu'elles comprennent du bruit. Développer des méthodes qui traitent toutes ces problématiques ensemble est aujourd'hui un défi. Beaucoup de méthodes proviennent du domaine de l'étude de graphes et sont ensuite appliquées à des maillages triangulaires, et d'autres méthodes sont directement destinées à être appliquées à des nuages de points. Ici, nous allons nous intéresser aux graphes seulement, et dans la section 4.2, plus de détails seront donnés sur les méthodes concernant les deux autres représentations.

Les premières méthodes ont été introduites sous la forme de réseaux de neurones récurrents (RNNs) [55, 119], présentant des restrictions en terme d'apprentissage, corrigées par l'utilisation de méthodes plus modernes comme des GRUs [79]. Des convolutions appliquées à des graphes ont ensuite été présentées par Duvenaud et al. [46], mais leur méthode présentait des difficultés à être appliquée à des graphes larges. Niepert et al. [103] ont appliqué des convolutions à des graphes mais une étape de prétraitement était nécessaire. Les méthodes ayant eu le plus de succès ont été amenées par Bruna et al. [20] en présentant un modèle utilisant des convolutions dans le domaine spectral, amélioré ensuite par Defferrard et al. [40] et Kipf et al. [65] en utilisant des polynomes de Chebyshev de premier ordre. Plusieurs travaux ont récemment été introduits, inspirés par l'équation de diffusion de l'information dans un graphe en utilisant l'équation de la chaleur [6, 26, 27, 133, 120, 11], représentant les contributions les plus récentes concernant l'étude de graphes en utilisant des réseaux de neurones.

La litérature concernant l'application de méthodes d'apprentissage profond à des graphes est très dense, mais la plupart des réseaux présentés reposent toujours sur le domaine spatial pour traiter l'information contenue dans ce type de données, même si elles peuvent être combinées à des méthodes spectrales. Dans cette thèse, l'idée est d'essayer de s'affranchir du domaine spatial et de ne traiter l'information que dans le domaine spectral afin de développer un modèle qui ne dépende pas de la structure et de la résolution initiale des données en entrée. Dans ce document, des travaux concernant l'analyse spectrale de maillages triangulaires sans apprentissage profond seront présentés dans la section 3.5, puis des techniques de génération de maillages basées sur ce domaine spectral seront introduites dans la section 4.2.

1.2 Objectifs

Nous avons vu les différentes représentations que peuvent avoir les données 3D. Couplées à la dimension temporelle, elles deviennent complexes à étudier et analyser. Comprendre, caractériser et reconstruire les formes et mouvements de corps humains est une tâche importante dans plusieurs applications, notamment pour la réalité virtuelle ou encore la détection d'anormalités dans l'analyse clinique. Mais la plupart des techniques récentes traitent les surfaces sans prendre en compte la dynamique. Cela est dû au fait qu'associer apprentissage profond géométrique et

étude temporelle est une tâche difficile, mais aussi au manque de données existantes pour des surfaces en mouvement. Avec l'apparition de techniques de capture et de modélisation plus efficaces, il faut penser au développement de méthodes capables de traiter ces nouvelles données en prenant en compte leur variation au cours du temps.

1.2.1 Projet Human4D

Le projet Human4D a pour but de relever ces nouveaux défis en cherchant à développer de nouvelles modélisations 4D de corps humains en mouvement. L'ambition du projet est d'aller au-delà des représentations existantes qui se concentrent soit sur l'aspect statique, soit sur l'aspect dynamique. Même si des méthodes ont été introduites ces dernières années pour résoudre ces problèmes séparément, peu de travaux s'intéressent réellement à l'association des deux, particulièrement sur des surfaces de corps humains évoluant dans le temps. Le corps humain est sujet à des déformations particulières, rendant difficile son étude et l'application directe des architectures existantes.

Human4D regroupe plusieurs équipes à travers la France. Le projet est dirigé par Hyewon SEO. Les équipes se situent à Strasbourg au laboratoire ICube, à Grenoble au centre INRIA, à Lille au laboratoire CRIStAL et à Lyon au laboratoire LIRIS. Aussi, une plateforme de capture de surfaces en mouvement est associée au projet : la salle Kinovis au centre INRIA à Grenoble. La mission de notre équipe au LIRIS est de se concentrer sur le développement d'une nouvelle représentation de surfaces dynamiques en se concentrant sur l'utilisation de méthodes d'analyse spectrale.

Les méthodes les plus populaires proposent de représenter les mouvements de corps humain à l'aide de structures articulées simples telles que des squelettes : plus d'informations sur ce sujet sont données dans la section 5. Afin de générer des surfaces à l'aide de ces méthodes, des étapes supplémentaires sont nécessaires telles que du squelettage (*rigging*) et de la mise en correspondance de surfaces avec des squelettes (*skinning*). Ces représentations ont été largement adoptées, mais elles ont une expressivité limitée car elles ne contiennent pas l'information surfacique. L'idée de nos travaux est de coupler étude spectrale et dynamique afin de proposer une architecture capable de traiter plus d'informations sans étape supplémentaire.

1.2.2 Associer étude spectrale, dynamique et apprentissage profond

L'étude de surfaces dans le domaine spectral est une méthode largement utilisée dans le domaine de l'informatique graphique. Elle repose sur l'utilisation de vecteurs propres et valeurs propres calculés depuis des opérateurs spécifiques. Elle a premièrement été utilisée pour résoudre des problèmes de compression, segmentation, lissage ou mise en correspondance. Depuis l'apparition des méthodes d'apprentissage profond, elle a naturellement été couplée à des architectures composées de réseaux de

neurones. Mais la réunion de ces deux mondes n'a pour l'instant pas été appliquée à la dynamique du corps humain. C'est ce qui est étudié dans ce document, en mettant l'accent sur l'aspect génératif des algorithmes proposés.

L'objectif de ce travail de thèse est de montrer qu'il est possible d'utiliser seulement les coefficients spectraux (dont le processus pour les calculer est expliqué dans la section 3) afin d'entraîner des modèles génératifs d'apprentissage profond. En effet, ces coefficients contiennent la géométrie de manière compressée et une surface est retrouvable depuis ces derniers à une faible erreur près si l'on utilise suffisamment de fréquences. De plus, ils permettent de s'affranchir d'une éventuelle trop haute résolution de la connectivité et permettent de donner un contrôle au réseau sur la quantité d'information à laquelle il a accès. L'objectif de notre projet étant d'entraîner un modèle d'apprentissage profond sur des bases de données denses, ce sont les coefficients spectraux qui ont été sélectionnées comme entrée de ces modèles. Les principes et le fonctionnement de ces algorithmes seront introduits dans les chapitres suivants.

1.3 Liste des publications

Cette thèse a conduit aux deux publications suivantes :

1. Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. Representation learning of 3D meshes using an Autoencoder in the spectral domain. Computers & Graphics, 107 :131-143, October 2022. [74]
2. Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. SpecTrHuMS : Spectral transformer for human mesh sequence learning. Computers & Graphics, 115 :191-203, October 2023. [75]

Les résultats en lien avec la première publication seront présentés dans le chapitre 4, et ceux en lien avec la deuxième seront présentés dans le chapitre 5.

Chapitre 2

Bases de données et outils

Afin de pouvoir implémenter des algorithmes capables de créer des séquences de maillages humains à l'aide de méthodes d'apprentissage, il a premièrement fallu manipuler des bases de données contenant ce type d'objets. Dans ce chapitre, un modèle 3D réaliste de corps humains et une large base de données représentant des mouvements humains sont premièrement introduits. Ensuite, plusieurs outils permettant la création et l'entraînement simplifiés de réseaux de neurones ainsi que la visualisation de données générées sont présentés.

2.1 Bases de données

Les modèles d'apprentissage profond ont besoin de bases de données denses pour pouvoir être entraînés, et les résultats produits sont principalement impactés par la qualité des échantillons utilisés. Lorsqu'il s'agit de corps humains représentés en 3 dimensions, l'acquisition s'avère plus difficile qu'avec du texte ou des images. Dans cette section, le formalisme SMPL ainsi que la base d'entraînement dense AMASS qui ont permis d'entraîner les modèles représentant les contributions de cette thèse sont présentés, ainsi que des données issues de la salle d'acquisition KINOVIS.

2.1.1 SMPL

Auparavant, l'animation de maillages 3D impliquait de manuellement créer des squelettes correspondant à une surface (*rigging*), de modifier l'orientation des articulations les composant afin d'obtenir une pose voulue, de générer la surface correspondante puis de la sculpter manuellement, ce qui est un processus fastidieux demandant beaucoup d'efforts de la part d'artistes et générant parfois des modélisations non réalistes. Cela a motivé l'introduction de formalismes permettant de représenter plus facilement des corps humains.

Skinned Multi-Person Linear (SMPL) est un modèle entraîné permettant de représenter des corps humains en utilisant diverses identités et poses. La première version est introduite par Loper et al. [87]. Le processus classique de squelettage prend en entrée une référence, des positions d'articulations, des poids associant

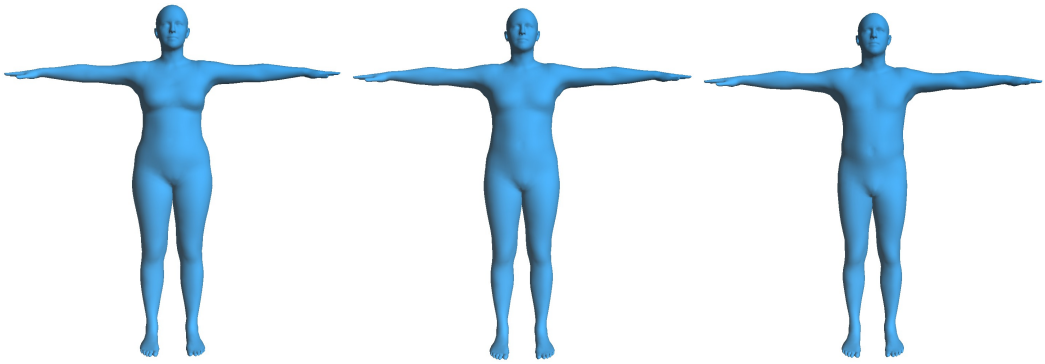


FIGURE 2.1 – Les trois identités de référence utilisées dans SMPL. De gauche à droite : femme, neutre et homme.

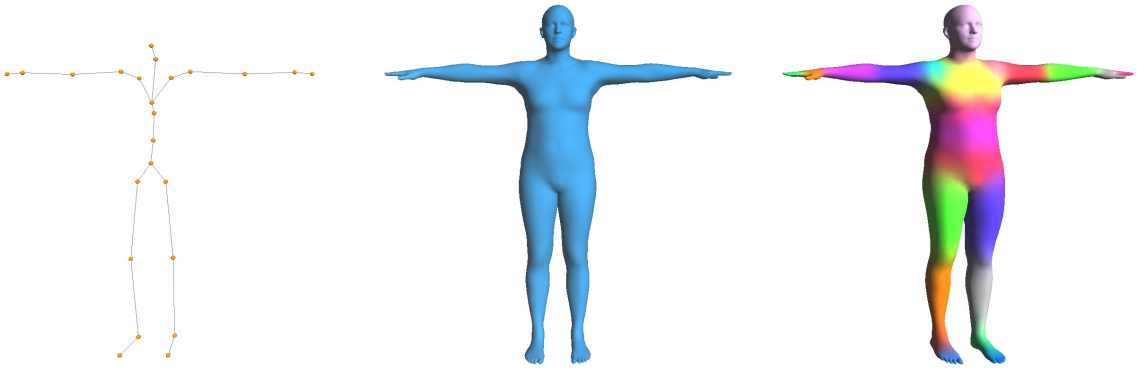


FIGURE 2.2 – Squelette, maillage et poids de l'identité neutre dans une pose basique.

chaque jointure du squelette aux sommets de la connectivité de référence et une pose voulue pour générer un maillage triangulé. Le problème est que les modèles classiques de squelettage (*linear blend skinning, LBS*) génèrent des déformations qui ne sont pas réalistes, surtout au niveau des articulations. SMPL corrige cela en proposant un modèle de squelettage amélioré entraîné sur des scans réalistes et présentant moins d'artéfacts que des techniques simples de LBS. Le maillage en sortie a la même topologie que celui de référence : toutes les surfaces générées avec SMPL auront donc la même connectivité. Il est possible de manipuler l'apparence de la référence, donnant ainsi un contrôle sur l'identité du maillage en sortie. Cela autorise la production de représentations de n'importe quelle identité dans n'importe quelle pose, et permet donc au modèle de fonctionner avec des processus d'animation. Des versions améliorées de SMPL ont été présentées ensuite : SMPL-H [116] qui ajoute des possibilités de contrôle sur les mains et SMPL-X [106] qui couple cette première amélioration avec un contrôle augmenté du visage. Afin de montrer des exemples de visualisations de maillages générés à l'aide de SMPL, le visualiseur smplxpp est utilisé.

Dans la figure 2.1, les 3 identités de référence disponibles sont présentées. Ces identités correspondent à un ensemble de paramètres de base. Leur pose est repré-

sentée par un ensemble de position des articulations du squelette. Grâce à des poids reliant les jointures et les points de la surface, le maillage peut être généré. La figure 2.2 montre un squelette à gauche, les poids reliant articulations et sommets à droite, et le maillage résultant au milieu.

La figure 2.3 présente la différence entre les modèles SMPL, SMPL-H et SMPL-X. SMPL-H permet d'avoir un contrôle supplémentaire sur les mains, mais garde la même connectivité que SMPL. SMPL et SMPL-H génèrent des maillages ayant 6 890 sommets, alors que ceux générés avec SMPL-X possèdent 10 475 sommets.

Afin de générer des maillages, il est possible de manuellement sélectionner des positions et angles de jointures afin de produire des poses réalistes souhaitées. Néanmoins, cela demanderait trop de temps afin de générer des bases de données assez denses pour entraîner des modèles. La figure 2.4 présente des maillages générés en utilisant des positions d'articulations de squelette aléatoires. On voit que les surfaces générées ne sont pas réalistes, et que ce n'est pas une solution viable pour générer des bases de données. Mais la paramétrisation SMPL peut être utilisée en la couplant à des données issues de captures de mouvement en utilisant AMASS.

2.1.2 AMASS

AMASS [89] est une large base de données qui unifie plusieurs ensembles de captures de mouvement en les représentant avec SMPL. Elle est augmentée au fur et à mesure que de nouvelles captures sont adaptées au formalisme AMASS. Les bases de données sont téléchargeables à l'adresse suivante : `https://amass.is.tue.mpg.de/`. Elles sont stockées sous forme de paramètres SMPL, ce qui permet de fortement réduire la place occupée sur le disque dur (30 fois moins en comparaison avec la géométrie des surfaces directement). En revanche, cela implique un prétraitement qui peut être coûteux en temps afin d'obtenir la position des jointures des squelettes ou des sommets des surfaces.

BABEL [110] est une base de données qui permet d'annoter les animations contenues dans AMASS, et la page web accueillant le projet permet de les visualiser `https://human-movement.is.tue.mpg.de/explore/`.

Les bases de données proposées par AMASS sont des ensembles de paramètres représentant l'identité d'une personne et des positions et angles des articulations de squelettes dans le temps représentant le mouvement effectué. Pour générer des surfaces en utilisant AMASS, il faut donc extraire ces paramètres contenus dans des fichiers *npz* puis générer des maillages en utilisant le formalisme SMPL. Afin d'accélérer le processus d'entraînement d'un réseau, la méthode la plus efficace est d'abord d'effectuer cette extraction puis de stocker l'information surfacique générée. Mais stocker directement les sommets des surfaces est très coûteux en terme d'espace occupé sur le disque dur, et la méthode utilisée dans ce document concernera plutôt le stockage de coefficients spectraux, introduits dans le chapitre 3. Plus

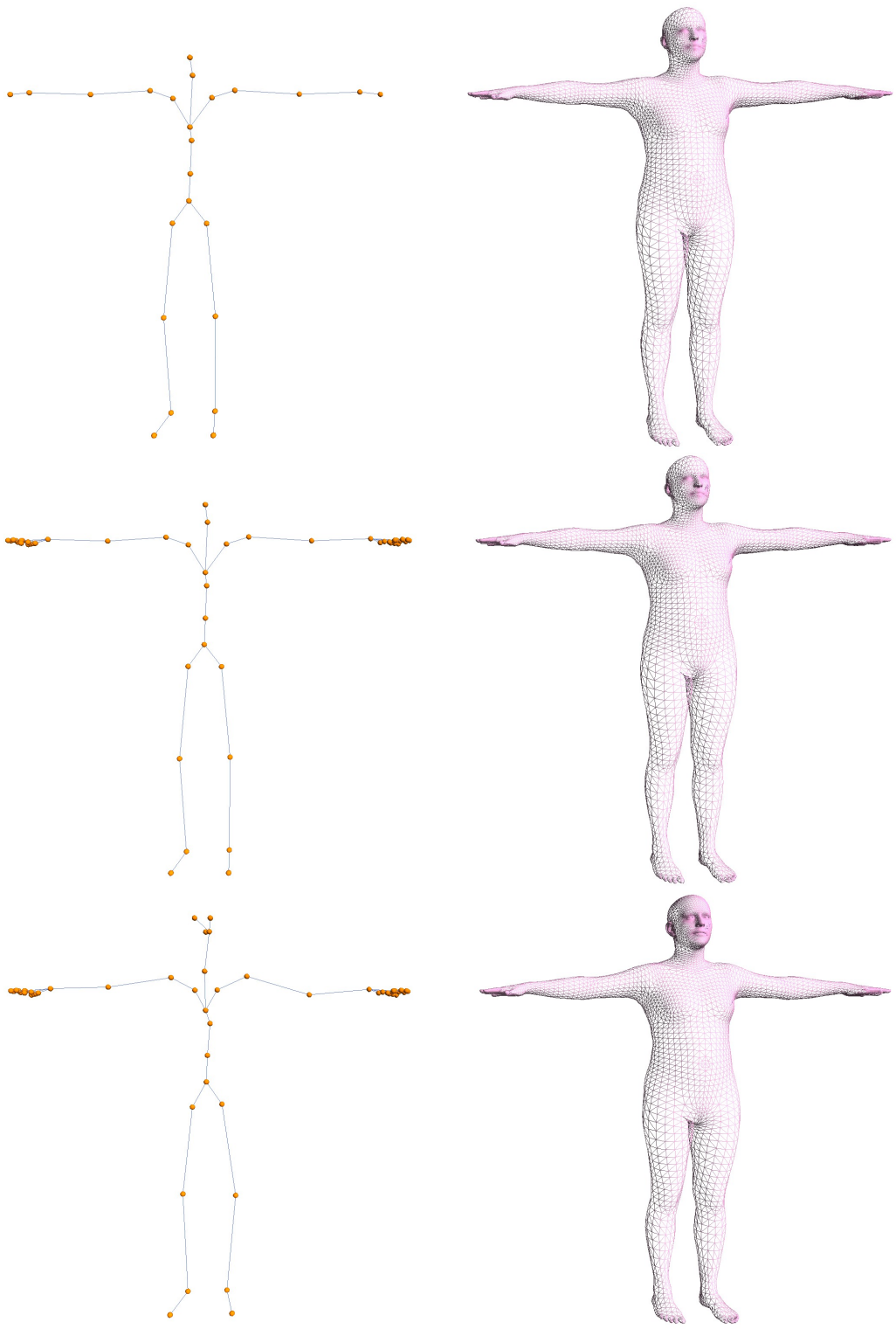


FIGURE 2.3 – Squelettes et connectivités des modèles. En haut : SMPL avec 6 890 sommets. Au milieu : SMPL-H avec 6 890 sommets. En bas : SMPL-X avec 10 475 sommets.

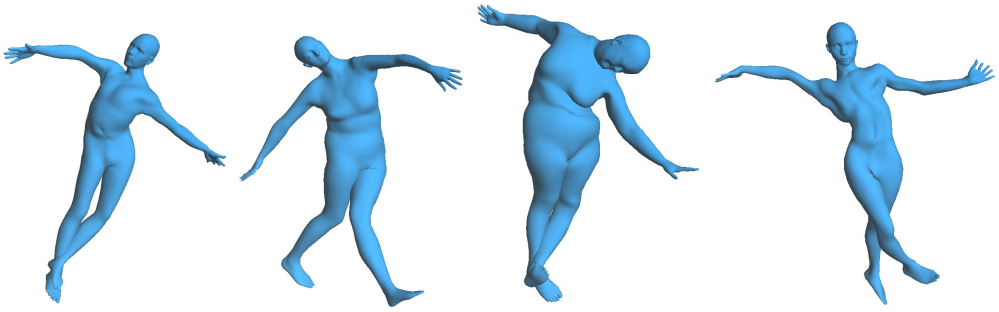


FIGURE 2.4 – Maillages non réalistes générés en utilisant des positions de squelettes aléatoires. Cette méthode n'est pas viable pour créer une base de données d'entraînement.

d'informations sur la manière de stocker ces surfaces seront données dans la partie 4.3.

SMPL permet donc de représenter efficacement des corps humains en mouvement en donnant la possibilité de stocker l'information de manière compressée mais en rendant un prétraitement obligatoire afin d'obtenir l'information correspondant aux surfaces. Cette méthode permet de générer des surfaces avec une connectivité constante ne possédant pas de bruit et peu de détails géométriques. Cette représentation de formes humaines simplifiées va permettre l'application de notre méthode comme nous le verrons plus tard. Néanmoins, les données brutes que nous visons à traiter dans de futurs travaux et issues de captures de mouvement sont en général représentées de manière plus complexe. C'est le cas des données fournies par la plateforme Kinovis.

2.1.3 KINOVIS

Kinovis est un environnement multi-caméras composé de deux plateformes complémentaires qui permet de capturer des surfaces en mouvement. Une première plateforme basée à l'INRIA Rhône-Alpes Grenoble est équipée d'un système de caméras couleur et d'un système de capture de mouvement. Elle permet la capture d'actions de formes mobiles telles que la marche ou la course. Les modèles spatio-temporels produits encodent des informations géométriques d'apparence dans le temps. Une deuxième plateforme installée aux Hôpitaux de Grenoble est équipée de caméras à rayons X et de caméras couleur. Elle permet la capture d'informations internes sur les parties du corps en mouvement en plus des informations de forme externes.

Dans la figure 2.5, on peut voir des exemples de maillages issus de la première plateforme. Chaque maillage a un nombre de sommets et une connectivité différents, comme illustré dans la figure 2.6. Aussi, les données peuvent présenter du bruit et des changements de topologie tels que des parties du corps collées entre elles comme les mains et la taille.

FIGURE 2.5 – Exemples de maillages issus de la plateforme KINOVIS. Chaque maillage possède un nombre de sommets et une connectivité différents.

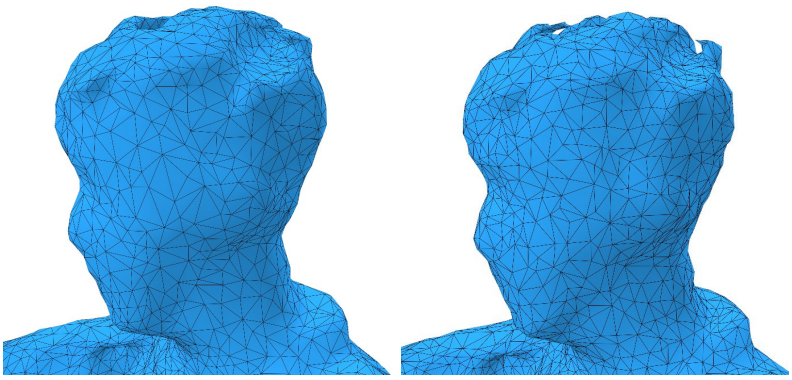


FIGURE 2.6 – Différence de connectivité des maillages KINOVIS.

L'objectif du projet dans lequel cette thèse s'inscrit était de mettre à disposition un ensemble dense de surfaces extrêment détaillées et en mouvement afin d'entraîner des modèles d'apprentissage profond (voir https://kinovis.inria.fr/4dhumanoutfit/). Mais la complexité du Règlement général sur la protection des données (RGPD) a fait que les bases de données n'ont pu être disponibles qu'à la fin de la thèse. Pour cette raison, les modèles ont été entraînés sur des données issues d'AMASS. De plus, étant donnée la différence de structure entre les données issues de la plateforme Kinovis et AMASS, des expériences supplémentaires auraient été nécessaires afin de faire fonctionner nos processus sur les maillages issus de la plateforme, ce que nous verrons dans les chapitres suivants.

2.2 Outils

Dans cette section, des détails sont donnés sur les outils utilisés durant la thèse, ainsi que sur les modifications qui leur ont été apportées afin de pouvoir les adapter à la problématique.

2.2.1 Apprentissage

Plutôt que de réimplémenter les algorithmes existants, des librairies les mettant à disposition ont été utilisées. Ces algorithmes d'apprentissage profond ont aussi besoin de puissance de calcul rendue disponible grâce à la possibilité d'accès au supercalculateur Jean Zay.

PyTorch

Implémenter des modèles d'apprentissage implique de créer des modèles, d'optimiser leurs paramètres, de les sauver/charger, et enfin de les utiliser pour prédire des résultats. Il existe plusieurs librairies permettant cela, telles que Tensorflow ou PyTorch. Chacune ont leur particularités et sont utilisées dans différents domaines, permettant aux développeurs de fabriquer ces modèles plus facilement. Par exemple, Tensorflow est plus utilisée dans le domaine de l'industrie puisqu'elle permet de déployer des modèles entraînés en production plus facilement. En revanche, PyTorch, plus simple d'utilisation, est préférable dans le cadre du domaine de la recherche puisqu'il est plus simple de prototyper et de tester rapidement avec cette librairie. Pour cette raison, et parce que beaucoup de modèles en lien avec le sujet de recherche sont implémentés avec cette dernière, notre choix s'est orienté vers l'utilisation de PyTorch.

PyTorch Lightning

La librairie PyTorch permet la création simplifiée de modèles d'apprentissage profond. En revanche, la logique d'entraînement et de test reste fastidieuse, surtout

lors de la phase de prototypage. PyTorch Lightning, une librairie à code source ouvert, permet de faciliter cette logique en réduisant drastiquement le nombre de lignes de code à écrire. Elle a été créée pour résoudre les problèmes de répétition de code tout en offrant une structure modulaire et extensible pour encourager une approche plus structurée et organisée du développement. Aussi, elle permet une visualisation des scores d'entraînement en proposant une utilisation facile de Tensorboard, une application web qui permet de visualiser en temps réel l'avancement des calculs. Cette librairie est utilisée dans les codes fournis des deux contributions de cette thèse.

Ray Tune

Une autre tâche fastidieuse est la recherche de paramètres optimaux pour les modèles implémentés. Ces hyperparamètres sont utilisés pour contrôler le processus d'apprentissage, et les possibilités de configuration sont en général très nombreuses : taux d'apprentissage (*learning rate*), nombre d'échantillons donnés par étape (*batch size*), valeur de décrochage ou d'abandon (*dropout*), nombre de couches du modèle, nombre d'étapes maximale, etc... Au lieu de tester un entraînement potentiellement long pour chaque configuration de paramètres, il est possible d'utiliser la librairie Ray Tune qui permet d'utiliser des algorithmes de l'état de l'art tels que *Population Based Training* (PBT) ou *HyperBand/ASHA* afin de trouver automatiquement les hyperparamètres qui donneront les meilleurs résultats. Durant la thèse, des tests ont été réalisés en utilisant Ray Tune afin d'effectuer une recherche optimisée d'hyperparamètres.

LibTorch

Les librairies précédentes sont utilisées avec le langage *Python*, qui peut parfois s'avérer lent étant donné qu'il est interprété. Pour pallier ce problème, il est possible d'utiliser LibTorch, similaire à PyTorch mais utilisant du code *C++*. Plusieurs expériences ont été réalisées à l'aide de cette librairie durant la thèse, amenant à des processus beaucoup plus rapides mais impliquant des temps de développement et des codes plus long. Pour que les phases de prototypage soient plus rentables et que les bases de code correspondant aux contributions soient plus accessibles après publication, des librairies utilisant le langage *Python* ont été préférées.

Jean Zay

Jean Zay est un supercalculateur installé à l'IDRIS, centre national de calcul du CNRS, depuis 2019. Il est possible d'effectuer une demande d'attribution d'heures de calcul et environ 15 000 heures nous ont été attribuées. Afin de pouvoir lancer des calculs sur Jean Zay de manière efficace, plusieurs outils ont été développés durant la thèse : connexion automatisée en *ssh*, installation personnalisée d'environnements conda, compilation à distance de code *C++*, lancement automatisé de *jobs*, etc... Les

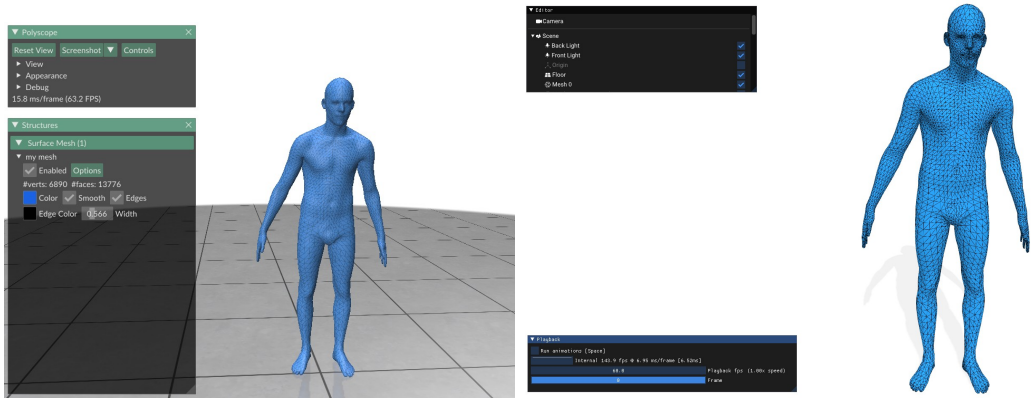


FIGURE 2.7 – Les deux visualiseurs utilisés : Polyscope à gauche et aitviewer à droite.

cartes graphiques disponibles sur Jean Zay sont des NVIDIA V100. Une machine locale a tout de même été utilisée pour certains entraînements avec la configuration suivante : *NVIDIA GeForce RTX 2070 Max-Q*. L'utilisation de Jean Zay nous a permis d'entraîner simultanément et rapidement plusieurs modèles de tests.

2.2.2 Visualisation

Étant donné que les travaux portent sur des données représentant des surfaces, des librairies de visualisation 3D ont été utilisées et modifiées afin de tester les modèles créés.

Polyscope

Polyscope est un visualiseur 3D utilisable avec *Python* ou *C++* qui présente une interface simple d'utilisation. Avec cet outil, il est facile de visualiser des maillages triangulaires ainsi que des quantités à afficher sur chaque sommet. Polyscope a été utilisé pour la visualisation de données statiques. Il a ensuite fallu trouver un visualiseur plus performant en terme de fréquence d'affichage de surfaces 3D pour des séquences de maillages. Il est possible de rendre l'affichage plus efficace pour des animations en l'utilisant avec du *C++*, mais cela implique des temps de développement plus longs et des bases de codes plus denses. Une alternative donnant la possibilité d'utiliser du code *Python* tout en proposant un affichage plus efficace de ces animations a donc été sélectionnée : aitviewer.

aitviewer

aitviewer est un ensembe d'outils qui permet la visualisation et l'interaction avec des séquences de données 3D. Il est utilisable en *Python*, permet d'ajouter facilement des interfaces, et permet de créer des vidéos de séquences générées.

Ces deux visualiseurs ont été exploités pour observer les données générées par les modèles implémentés. Cela signifie qu'ils ont été modifiés pour être capables

de charger et d'afficher les coefficients spectraux donnés en entrée aux modèles et décrits dans la section 3, d'afficher les maillages en résultant ou encore de réaliser des opérations d'interpolation dans les espaces latents créés par les réseaux (voir section 4). Ces modifications sont disponibles dans les bases de code fournies.

Nous avons vu dans ce chapitre les bases de données et les outils qui ont été utilisés, permettant la génération, le traitement et l'affichage de données 3D. Nous allons ensuite nous intéresser aux méthodes d'analyse spectrale appliquées à ce genre de représentations.

Chapitre 3

Traitement spectral de maillages

L'analyse spectrale de maillages triangulaires est une branche de l'informatique graphique qui permet de comprendre la structure et les propriétés de surfaces discrétisées en utilisant des outils similaires à la transformée de Fourier. La transformée de Fourier est utilisée pour analyser des signaux en les décomposant en une somme de fonctions sinusoïdales. De manière similaire, la transformée spectrale de maillages triangulaires utilise des opérateurs, matrices décrivant les relations entre les sommets des surfaces, en les décomposant en valeurs et vecteurs propres. L'analyse spectrale de maillages triangulaires a été introduite dans les années 90 et largement utilisée jusque dans les années 2010 avant de laisser place à des processus plutôt orientés vers l'apprentissage profond. Elle concerne de nombreuses applications telles que la compression de maillage, la reconnaissance de formes ou encore la mise en correspondance.

Dans ce chapitre, nous allons explorer les concepts fondamentaux de l'analyse spectrale de maillages triangulaires, en commençant par rappeler le fonctionnement de la transformée de Fourier et son lien avec l'opérateur Laplacien. Nous allons également montrer quelques résultats appliqués aux maillages de corps humains traités dans cette thèse, et enfin examiner les différentes applications de l'analyse spectrale de maillages triangulaires.

3.1 Transformée de Fourier et Laplacien

Ce qui a motivé le développement du traitement spectral appliqué à des surfaces était une analogie avec l'analyse de Fourier discrète classique. Pour introduire le traitement spectral de maillages, au lieu de directement manipuler des données 3D, nous commencerons par observer le lien entre transformée de Fourier et opérateur Laplacien sur des données à une dimension. Ensuite, des opérations de compression et de lissage seront présentées en utilisant l'opérateur Laplacien sur des signaux à deux dimensions représentant des courbes.

3.1.1 Signal à une dimension

La transformée de Fourier et de Laplace sont deux outils mathématiques puissants largement utilisés dans le domaine du traitement du signal pour analyser et comprendre le comportement des signaux à une dimension. Elles permettent de passer d'une représentation temporelle d'un signal à une représentation fréquentielle ou complexe, offrant ainsi une perspective différente pour étudier les caractéristiques et les propriétés d'un signal.

Transformée de Fourier

Joseph Fourier exprime au XIXe siècle l'idée que toute fonction peut être décomposée en somme de sinus et de cosinus. Cela a permis d'introduire la transformée de Fourier continue directe 3.1 et inverse 3.2.

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-2\pi i f t}\, dt \tag{3.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{2\pi i f t}\, df \tag{3.2}$$

Ici, la transformée d'un signal temporel $x(t)$, t représentant le temps, à la fréquence f est désignée par le nombre complexe $X(f)$. L'évaluation de l'équation 3.1 pour toutes les valeurs de f produit le signal dans le domaine fréquentiel. Pour chaque fréquence, la magnitude du nombre complexe $X(f)$ représente l'amplitude d'une sinusoïde complexe, et l'argument représente le décalage de phase de cette sinusoïde. Si une fréquence n'est pas présente dans le signal temporel d'origine, la transformée aura une valeur de 0 pour celle-ci. La transformée de Fourier inverse fournit quant à elle un processus de synthèse qui recrée le signal d'origine à partir de sa représentation dans le domaine fréquentiel.

La transformée de Fourier continue d'un signal temporel est une fonction continue de la variable f. En informatique, pour représenter un signal quelconque, il n'est possible d'utiliser qu'un nombre fini de valeurs, et il est nécessaire de prendre en compte la version discrète de ces formules. Un signal continu est premièrement échantillonné ou discrétisé afin de ne garder qu'une suite discrète de valeurs de ce dernier, donnant un signal discret $x[n]$, avec $n \in [0, N-1]$, N étant le nombre d'échantillons. Depuis ce signal échantillonné, il est ensuite possible d'appliquer une transformée de Fourier discrète directe (*Discrete Fourier Transform* ou DFT) 3.3 puis inverse (*Inverse Discrete Fourier Transform* ou IDFT) 3.4.

$$X(k) = \sum_{n=0}^{N-1} x[n] \cdot e^{-2\pi i k n/N} \tag{3.3}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{2\pi i k n/N} \tag{3.4}$$

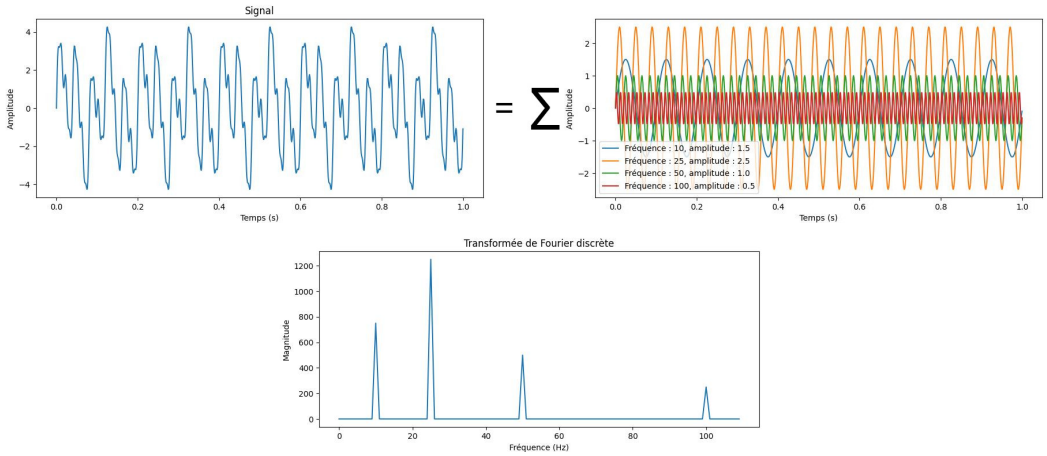


FIGURE 3.1 – En haut à gauche, le signal étudié, composé de sommes de signaux représentés à sa droite. En bas, les modules de la transformée de Fourier du signal.

La transformée de Fourier discrète permet de calculer les coefficients spectraux correspondant à un signal échantillonné. Par exemple, prenons un signal artificiel représenté dans la figure 3.1 en haut à gauche, composé de la somme de quatre signaux de différentes fréquences et amplitudes représentés à sa droite. Alors sa transformée de Fourier permet de visualiser les fréquences et les magnitudes des signaux dont il est composé, représentées en bas dans la figure 3.1. La transformée de Fourier est un moyen de déterminer quelles fréquences sont présentes dans un signal temporel ainsi que leur puissance.

Il est possible de reconstruire le signal d'origine depuis ces coefficients spectraux en utilisant une transformée de Fourier inverse. On peut voir sur la figure 3.2 plusieurs reconstructions partielles utilisant différents nombre de fréquences. Plus on utilise de fréquences, plus le signal est correctement reconstruit. Il est donc possible de construire des versions filtrées passe-bas du signal d'origine en n'utilisant que les basses fréquences, ou passe-haut en n'utilisant que les hautes. Les opérations de transformées directe et inverse permettent de réaliser des opérations de filtrage en manipulant le signal dans le domaine des fréquences.

L'évaluation de la transformée de Fourier discrète est réalisée pour plusieurs valeurs différentes de fréquences. Pour chacune de ces fréquences, elle est calculée comme la somme d'un produit entre l'échantillon n du signal d'origine et le terme $e^{-2\pi ikn/N}$, avec $n \in [0, N-1]$. L'expression de la transformée de Fourier discrète 3.3 est équivalente à celle d'un produit matriciel :

$$X = \Phi \cdot x \tag{3.5}$$

La matrice Φ est la matrice de la DFT et est représentée par :

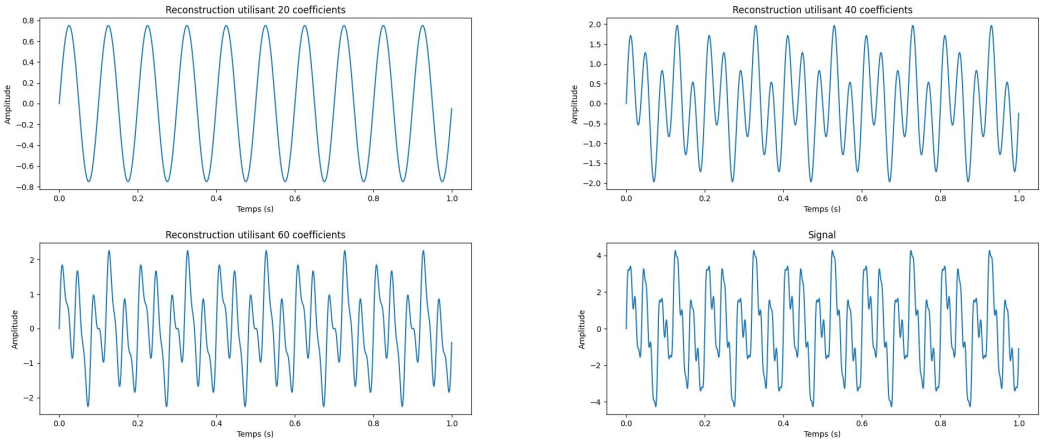


FIGURE 3.2 – Reconstruction du signal en utilisant une transformée inverse et différents nombre de coefficients. De gauche à droite et de haut en bas : 20, 40, 60 et 100 fréquences.

$$\Phi = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-2i\pi\frac{1}{N}} & \cdots & e^{-2i\pi\frac{N-1}{N}} \\ 1 & e^{-2i\pi\frac{2}{N}} & \cdots & e^{-2i\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-2i\pi\frac{N-1}{N}} & \cdots & e^{-2i\pi\frac{(N-1)^2}{N}} \end{bmatrix} \tag{3.6}$$

Les algorithmes de transformée de Fourier rapide (*Fast Fourier transform* ou FFT) utilisent les symétries contenues dans cette matrice afin de réduire le temps de calcul de cette transformée.

Les colonnes de cette matrice sont des vecteurs orthogonaux entre eux. Finalement, la transformée de Fourier discrète peut être vue comme la projection d'un signal sur une base orthogonale définie par la matrice Φ. Maintenant, voyons comment cette base est reliée à l'opérateur de Laplace.

Laplacien

L'opérateur de Laplace ou Laplacien est un opérateur différentiel donné par la divergence du gradient d'une fonction :

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f \tag{3.7}$$

Intuitivement, le Laplacien d'une fonction en un point mesure la courbure moyenne locale de cette fonction en ce point. Il est possible d'approximer cet opérateur en Laplacien discret en utilisant la méthode des différences finies afin de pouvoir définir la matrice du Laplacien. Si l'on considère maintenant le signal comme périodique, il est possible de le prendre en compte comme étant les coordonnées d'un contour fermé échantillonné uniformément. Depuis la connectivité de ce contour, il est possible de créer la matrice du Laplacien de taille $n * n$, selon sa définition :

$$L_{ij} = \begin{cases} \deg(v_i), & \text{si } i = j, \\ -1, & \text{si } i \text{ et } j \text{ sont adjacents}, \\ 0, & \text{sinon.} \end{cases} \tag{3.8}$$

Le Laplacien encode la connectivité simple du signal : chaque échantillon est relié au précédent et au suivant, y compris le premier et le dernier en les considérant comme connectés. Cette matrice L a l'allure suivante :

$$L = \begin{bmatrix} 2 & -1 & 0 & \dots & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & -1 & 2 & -1 \\ -1 & \dots & \dots & 0 & 0 & -1 & 2 \end{bmatrix} \tag{3.9}$$

et peut être calculée comme $L = D - A$, avec D la matrice diagonale des degrés de chaque point du signal et A la matrice d'adjacence, qui seront identiques pour tous les signaux de la même longueur.

Cette matrice Laplacienne (*Graph Laplacian*), peut être considérée comme une forme matricielle de l'opérateur de Laplace discret. Ici, elle a d'abord été calculée sur un graphe simple représentant un signal périodique, mais elle peut aussi être définie pour des graphes plus complexes, comme nous le verrons dans la section 3.2. L'analyse spectrale appliquée à des graphes ressemble à la transformée de Fourier discrète classique, mais au lieu d'utiliser la base standard de sinusoïdes complexes, elle utilise les vecteurs propres de la matrice Laplacienne associée au graphe.

Ces vecteurs sont obtenus à l'aide d'une décomposition en éléments propres de la matrice L en utilisant par exemple la fonction eig de *numpy*. Si l'on compare les vecteurs propres obtenus avec ceux formant la base de la transformée de Fourier discrète (voir figure 3.3 en haut), on observe un comportement identique : le premier est constant et les suivants montrent des oscillations de plus en plus élevées. Il est ensuite possible de calculer les valeurs propres correspondant aux vecteurs de la matrice de la DFT associées à L ($L\Phi_i = \lambda_i \Phi_i$). Les valeurs propres correspondant aux vecteurs de la DFT existent et sont identiques aux valeurs propres calculées depuis L (voir figure 3.3 en bas), signifiant que les vecteurs de la DFT sont bien des vecteurs propres de L. Cela montre que la DFT permet de projeter les valeurs du signal sur une base de vecteurs propres qui peuvent être obtenus depuis un opérateur, le Laplacien.

Les vecteurs propres issus du Laplacien sont représentés par une matrice Φ de taille $n * n$. Comme L est symétrique, les valeurs propres sont réelles et Φ est orthogonale. Ces vecteurs propres servent de base pour projeter le signal dans le domaine des fréquences avec une simple multiplication de matrice $X = \Phi^T x$ et exprimer le

signal x en une somme de ces derniers. X est un vecteur colonne dont chaque élément est un coefficient spectral X_i qui est la projection du signal dans la direction du i-ème vecteur propre.

Ici, il est important de noter que ces vecteurs propres et valeurs propres ne dépendent pas du signal mais simplement de sa longueur, ce qui ne sera plus le cas lorsque des graphes faits d'une connectivité plus complexe seront étudiés.

3.1.2 Contour fermé

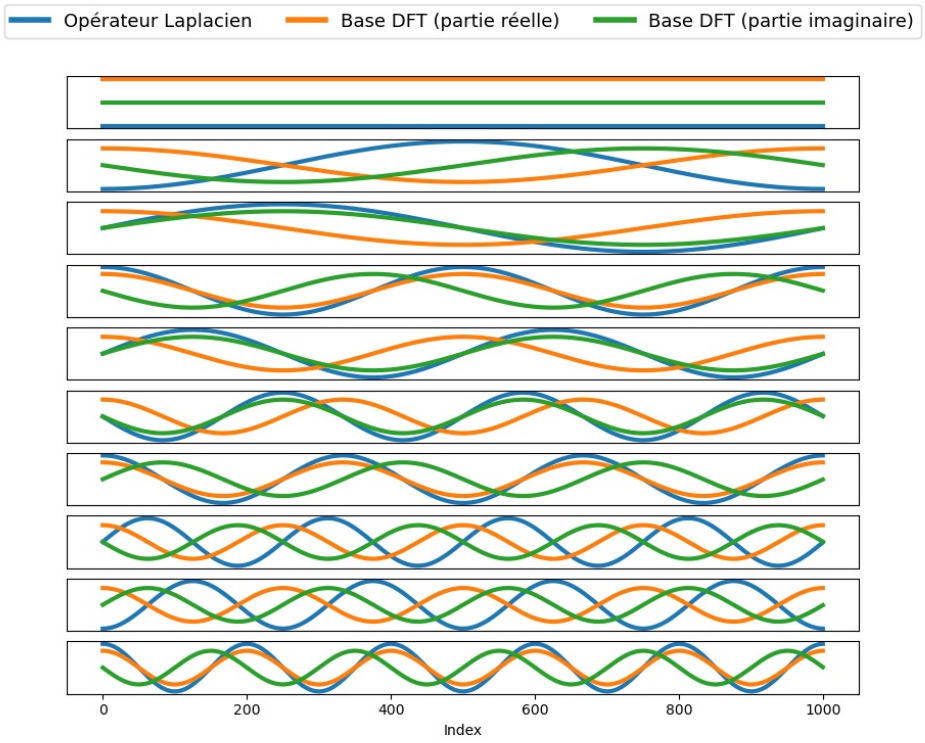
Prenons maintenant le contour représenté dans la figure 3.4 en haut comprenant 426 échantillons. Ce contour est composé d'une séquence de points 2D dont la première coordonnée évalue la position en abscisse et la deuxième en ordonnée de chaque échantillon. Chaque coordonnée est représentée séparément dans la figure 3.4 en bas. Depuis ce contour, il est possible de construire une matrice de Laplace de la même manière que dans les formules 3.8 et 3.9, que l'on peut ensuite décomposer en éléments propres afin d'obtenir des vecteurs propres, représentant une base sur laquelle on peut projeter chaque composante du contour. Cette transformation permet de représenter la courbe 2D dans le domaine fréquentiel et permet d'obtenir deux ensembles de coefficients spectraux.

La figure 3.5 montre la reconstruction partielle du contour d'origine en n'utilisant qu'une partie des coefficients spectraux calculés. Il est clair qu'en utilisant simplement la moitié des fréquences disponibles, le contour reconstruit approxime déjà correctement le contour d'origine. En bas de la figure 3.5 se trouve l'erreur entre la courbe reconstruite et la courbe d'origine en fonction du nombre de fréquences utilisées : les coefficients de basses fréquences contiennent la majorité de l'information. Cette compaction de l'information dans le domaine spectral peut par exemple être utilisée dans des applications de compression. Nous verrons dans les chapitres suivants que nos contributions reposent sur cet aspect de l'analyse fréquentielle.

Pour un contour, l'analyse fréquentielle est donc réalisée de la même manière que pour des signaux à une dimension en isolant chaque coordonnée de ce dernier. Lorsque l'on souhaite traiter des surfaces représentées par un ensemble de points 3D couplé à une triangulation, l'utilisation du Laplacien repose sur le même principe mais son calcul est réalisé différemment en raison de la connectivité plus complexe de la structure étudiée.

3.2 Opérateurs

De manière générale, les méthodes d'analyse spectrale appliquées à des maillages concernent l'étude ou la manipulation des valeurs propres, vecteurs propres ou de projections dérivées d'un opérateur. Bien avant que ces méthodes arrivent au domaine de l'informatique graphique, beaucoup de connaissances proviennent du domaine de l'analyse spectrale de graphes issues du papier pionnier de Fiedler et al. [50].



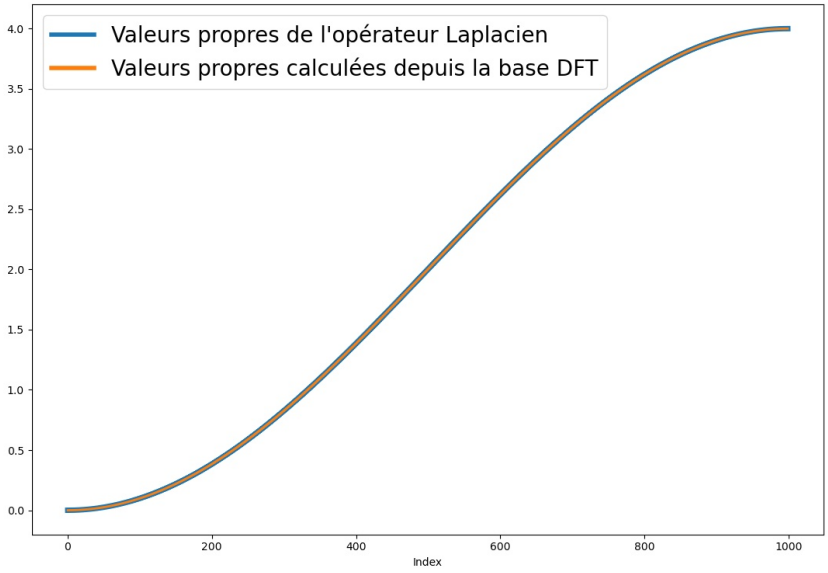


FIGURE 3.3 – En haut, sur chaque ligne, comparaison des 10 premiers vecteurs propres de la matrice Laplacienne L et des vecteurs formant la base de la transformée de Fourier discrète. En bas, les valeurs propres de L et les valeurs propres calculées depuis les vecteurs formant la base de la transformée de Fourier.

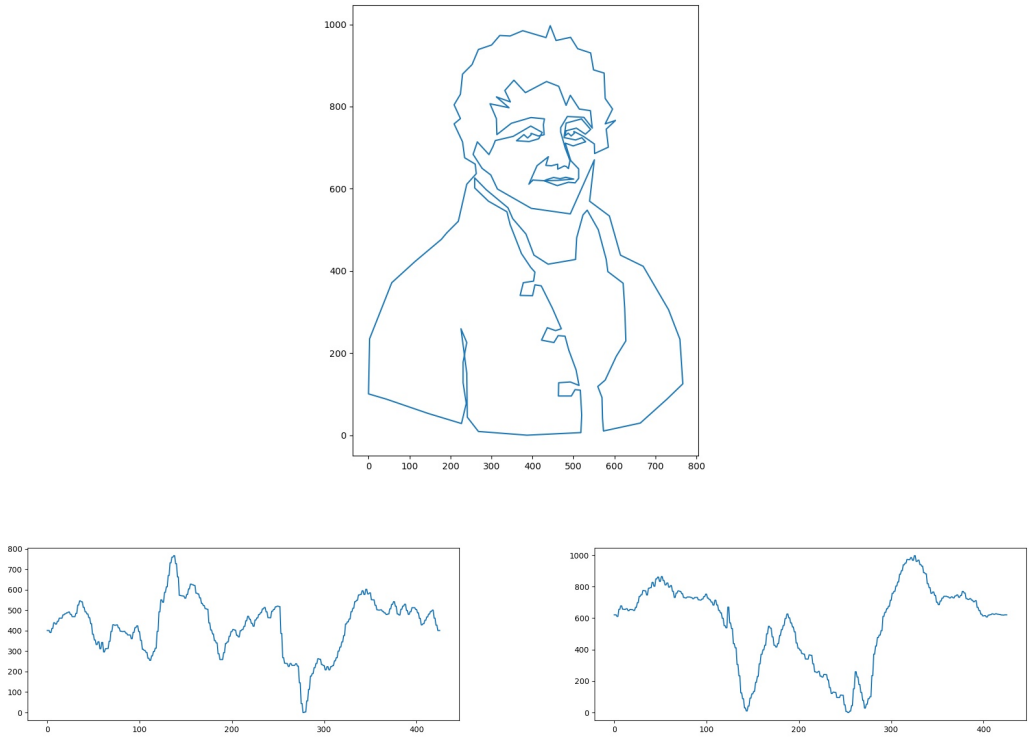


FIGURE 3.4 – En haut, une courbe représentée par une séquence de points 2D. En bas, la représentation séparée des deux coordonnées.

Dans cette section, nous allons étudier le comportement de deux opérateurs permettant de transformer la géométrie d'un maillage 3D en coefficients spectraux. Ces maillages sont en général représentés par un ensemble de points 3D et une connectivité. Cela permet de représenter une surface 2D plongée dans un espace à trois dimensions en utilisant les sommets et faces de ces maillages. Dans la figure 3.6, nous pouvons voir à gauche un exemple d'un tel maillage, ainsi que la connectivité le représentant à sa droite. L'analyse spectrale de maillages repose sur l'exploitation des valeurs propres, vecteurs propres et projections provenant d'opérateurs définis de plusieurs manières différentes.

La plupart des méthodes spectrales se basent sur un processus commun qui peut être divisé en trois étapes :

- Une matrice représentant un opérateur est définie comme incorporant les relations entre les sommets du maillage. Ces relations peuvent prendre en compte seulement la connectivité ou combiner information topologique et géométrique.
- Cette matrice est décomposée en éléments propres, donnant valeurs et vecteurs propres.
- Selon l'application, les valeurs propres, vecteurs propres ou une projection sur les vecteurs propres sont utilisés.

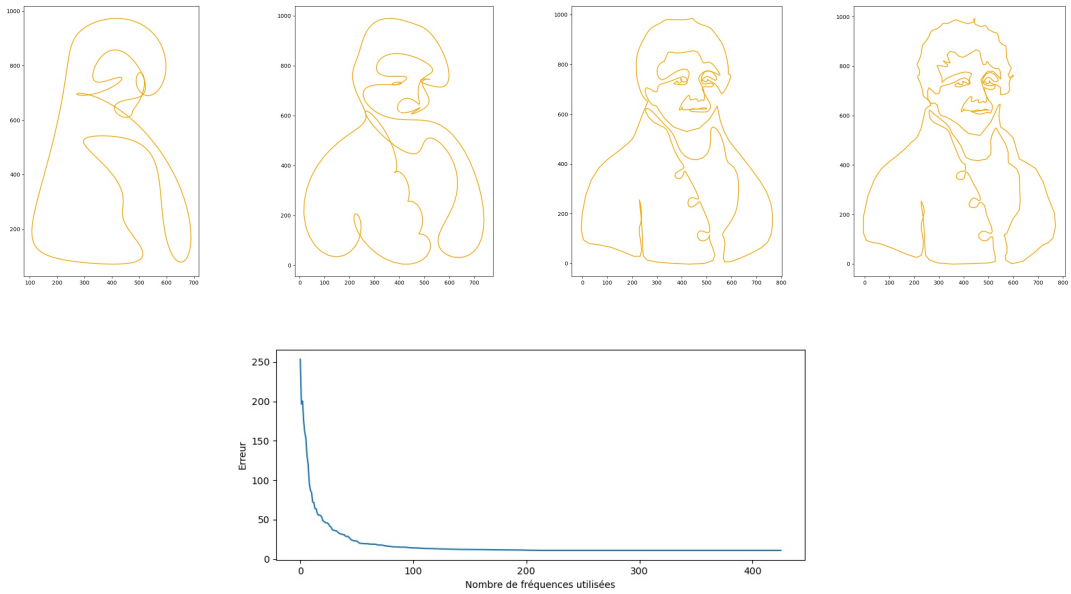


FIGURE 3.5 – En haut, plusieurs reconstructions en utilisant différents nombres de coefficients : 25, 50, 150 et 213 sur les 426 disponibles. En bas, la distance moyenne entre chaque échantillon de la courbe reconstruite et de la courbe d'origine en fonction du nombre de fréquences utilisées.

Les deux opérateurs abordés ensuite, la matrice Laplacienne (*Graph Laplacian*) et l'opérateur de Laplace-Beltrami, sont une généralisation à des surfaces courbes du Laplacien dans l'espace Euclidien. Le premier ne dépend pas de la géométrie et les vecteurs propres calculés peuvent être utilisés avec toutes les surfaces possibles utilisant la même triangulation, et il peut donc être nommé Laplacien topologique ou combinatoire. Le deuxième est la version géométrique du premier : l'opérateur de Laplace-Beltrami.

3.2.1 Matrice Laplacienne (*Graph Laplacian*)

La matrice Laplacienne découle du Laplacien classique, appliqué ici à des graphes, et provient du papier fondateur de Taubin et al. [129]. Dans ce cas de figure, la matrice du Laplacien sera différente de celle calculée dans les parties précédentes. En effet, la connectivité n'est plus représentée par de simples liens entre chaque échantillon mais par une connectivité plus complexe (voir figure 3.6 à droite). Néanmoins, la formule reste la même : $L = D - A$. Ainsi, depuis la seule connectivité du graphe ou du maillage, on pourra définir cette matrice que l'on pourra ensuite décomposer en vecteurs propres et valeurs propres.

Un maillage triangulaire est représenté par un ensemble de points P couplé à une triangulation T. Chaque point $p_i \in P$ est représenté par ses coordonnées cartésiennes $p_i = (x_i, y_i, z_i), i \in [1, n]$, n étant le nombre de sommets du maillage. Chaque face $t_i \in T$ est représentée par les indices des sommets la constituant $t_i =$

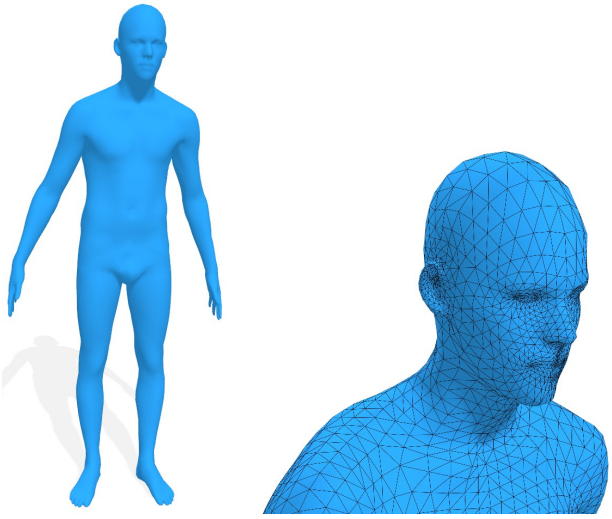


FIGURE 3.6 – À gauche, un maillage. À droite, sa connectivité.

Nombre d'éléments propres	FAUST	Kinovis
4096	∼28s	∼518s
2048	∼36s	∼187s
1024	∼9s	∼43s
512	∼3.5s	∼11s

TABLE 3.1 – Comparaison des temps de calcul des éléments propres. FAUST correspond à un maillage avec une connectivité SMPL ayant 6 890 sommets. Kinovis correspond à un maillage issu de la plateforme de l'INRIA ayant 16 418 sommets. Le fait que le calcul des vecteurs propres pour 2 048 fréquences soit plus long que pour 4 096 vient du solveur utilisé.

$(i_1, i_2, i_3), i \in [1, f]$, f étant le nombre de faces. Les matrices D, A et donc L peuvent être construites à partir de T.

La matrice L est une matrice carrée de taille $n * n$ depuis laquelle on peut calculer les k premières valeurs propres λ_i, représentées par des scalaires, et les k premiers vecteurs propres ϕ^i de dimension n avec $i \in [1, k]$ et $k <= n$. Les paires de valeurs propres et vecteurs propres satisfont l'équation $-L\phi^i = \lambda_i \phi^i$. À l'aide d'un algorithme de décomposition en éléments propres, il est possible d'obtenir les k vecteurs propres correspondant aux k valeurs propres de plus faibles magnitudes. Ces vecteurs propres correspondent aux colonnes de la matrice suivante :

$$\Phi = \begin{pmatrix} \phi_1^1 & \phi_1^2 & \dots & \phi_1^k \\ \vdots & \vdots & \vdots & \vdots \\ \phi_n^1 & \phi_n^2 & \dots & \phi_n^k \end{pmatrix} \tag{3.10}$$

Le calcul de ces éléments propres est réalisé à l'aide de la fonction eigs de Matlab : le processus est dépendant de l'implémentation intégrée à la plateforme. Il est possible d'utiliser des fonctions similaires provenant de numpy ou scipy en *Python* mais les temps de calcul sont plus longs. La table 3.1 présente les temps de calcul pour deux surfaces différentes, l'une correspondant à une connectivité SMPL ayant 6 890 sommets (FAUST), l'autre correspondant à un maillage issu de la plateforme Kinovis ayant 16 418 sommets. Les calculs sont réalisés sur une machine locale (voir 1). Plus la connectivité a de sommets, plus le temps de calcul de ces éléments propres est long, même pour un nombre similaire d'éléments propres voulu. Mais comme nous l'avons vu précédemment pour des signaux, et comme nous allons le voir ensuite pour des surfaces, il est possible d'analyser un objet dans le domaine des fréquences sans avoir à calculer tout son spectre.

Dans la figure 3.7, plusieurs de ces vecteurs sont représentés sur un maillage en associant une couleur à chaque sommet en fonction de la valeur du vecteur propre en ce sommet (les valeurs du vecteur propre sont normalisées entre -1 et 1). On peut voir que celui correspondant à la première fréquence est identique partout, signifiant qu'il a un contrôle sur toute la surface. Ceux de basses fréquences présentent des oscillations faibles, signifiant qu'ils ont un contrôle sur des zones larges. Ceux de hautes fréquences présentent des oscillations élevées, signifiant qu'ils ont un contrôle plus localisé. Ils présentent un comportement oscillatoire et peuvent être considérés comme les modes de vibration ou les harmoniques de la surface, et les valeurs propres peuvent être considérées comme les fréquences. Il est possible d'observer leur capacité de contrôle en manipulant les coefficients spectraux obtenus depuis une transformée spectrale.

Les coefficients spectraux sont calculés à partir des coordonnées cartésiennes des sommets par la transformation spectrale $C = \Phi^T \cdot P$. Ils sont représentés par un ensemble de coordonnées fréquentielles 3D : $c_i = (u_i, v_i, w_i), i \in [1, k]$ (voir figure 3.8 en haut à gauche), ordonnées en fonction de la fréquence à laquelle elles sont associées. Une géométrie approximant celle d'origine peut être calculée en utilisant la multiplication de matrice $\hat{P} = \Phi \cdot C$, correspondant à une transformée spectrale inverse partielle. En fonction du nombre de fréquences k utilisées, la géométrie retrouvée introduira plus ou moins d'erreurs (voir figure 3.9). Comme pour les signaux 1D, la notion de compaction de l'énergie apparaît : la plupart de l'information contenue dans la géométrie du maillage est compactée dans les basses fréquences, signifiant qu'elle peut être approximée ou compressée efficacement avec un faible nombre de coefficients spectraux.

Une modification des coefficients spectraux amène à des perturbations sur la géométrie du maillage dans le domaine spatial après une transformée spectrale inverse, comme illustré dans la figure 3.8. Les perturbations simulées ici sont simplement des translations d'un coefficient sur un axe. Le coefficient spectral de fréquence i possède un contrôle sur les zones affectées par le vecteur propre correspondant illustré dans la figure 3.7. Dans la figure 3.8, les modifications apportées aux spectres sont naïves et mènent naturellement à des perturbations qui ne sont pas réalistes.

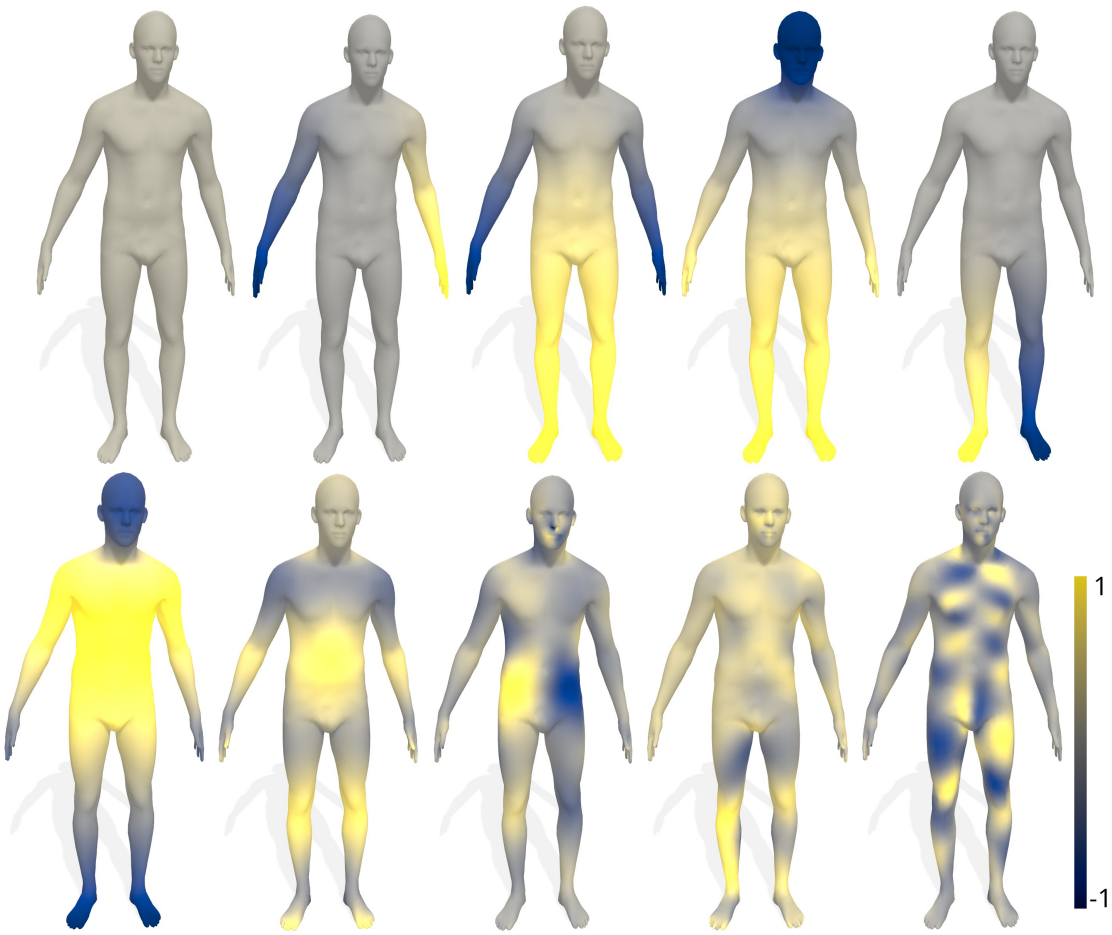


FIGURE 3.7 – Plusieurs vecteurs propres correspondant, de gauche à droite et de haut en bas, aux fréquences : 0, 1, 2, 3, 4, 5, 32, 64, 128 et 256.

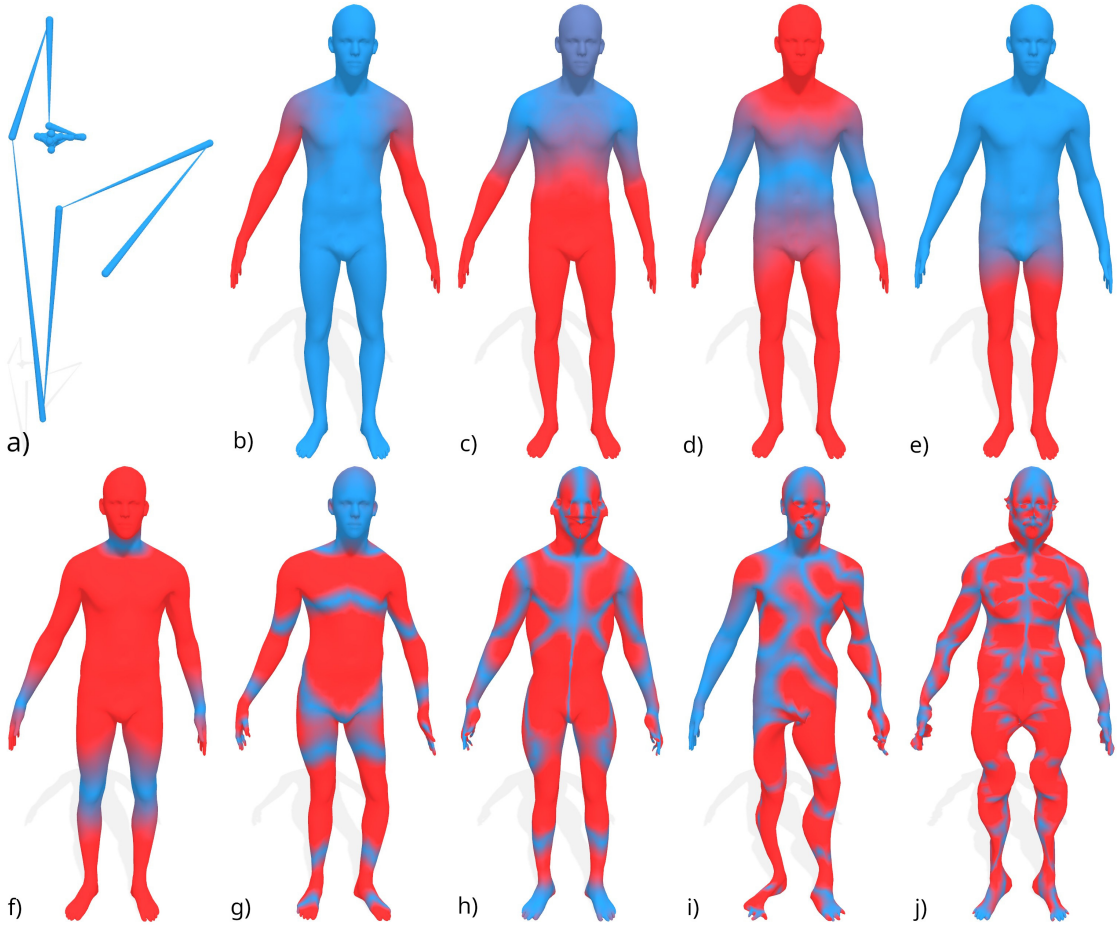


FIGURE 3.8 – a) : exemple de coefficients spectraux. De b) à j) : perturbations lorsque différents coefficients spectraux sont affectés correspondant respectivement aux fréquences : 1, 2, 3, 4, 5, 32, 64, 128 et 256. La couleur de chaque sommet indique sa distance avec le sommet correspondant du maillage de référence, allant du bleu (pas d'erreur) au rouge (erreur maximale) : la perturbation d'un coefficient de fréquence i modifie la zone qu'affecte le vecteur propre i.

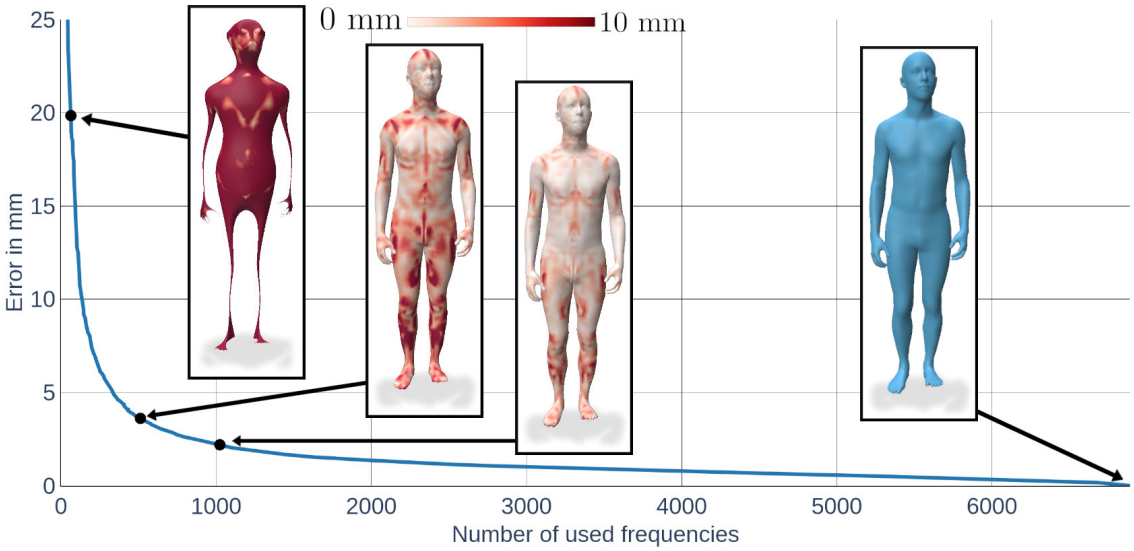


FIGURE 3.9 – Distances moyennes entre les sommets correspondant de transformation directe puis inverse en fonction du nombre de vecteurs propres choisi. Les maillages affichés correspondent respectivement à 64, 512, 1 024 et 6 890 fréquences. En utilisant toutes les fréquences disponibles (6 890, le nombre de sommets), la géométrie exacte est récupérée.

Le concept de nos travaux repose sur ces deux principes (manipulation des coefficients spectraux et compaction de l'énergie) : plutôt que d'apprendre à un algorithme comment faire évoluer les sommets de la surface dans le domaine spatial directement, il est possible de lui apprendre comment se comportent une partie des coefficients dans le domaine des fréquences. Cela permet d'introduire un contrôle sur la quantité d'information traitée (le nombre de fréquences donné), et permet aux modèles d'apprentissage profond présentés dans les chapitres suivants d'être entraînés sur des bases de données plus denses sans augmenter le temps de traitement. Le mouvement d'un corps humain concerne des variations globales et non les détails des surfaces qui sont plutôt liées à la morphologie. L'algorithme aura donc accès à l'information importante contenue dans les basses fréquences.

Dans les figures 3.10 et 3.11, la compaction de l'énergie dans le domaine spectral est illustrée pour des maillages représentant une même surface discrétisée différement en fonction du pourcentage ou non du nombre total de coefficients. Pour chaque exemple, le maillage de base est un maillage issu de la plateforme Kinovis, et ceux composés de moins de sommets sont obtenus en appliquant un algorithme de décimation d'arêtes (*Quadric Edge Collapse Decimation*). Les courbes représentent l'erreur entre le maillage d'origine décimé ou non et le maillage reconstruit en fonction du nombre de fréquences disponibles utilisées. En utilisant des maillages ayant plus de sommets, la capacité de compression reste efficace. Cela montre qu'il est possible d'utiliser ces principes sur des surfaces plus détaillées.

Finalement, on peut s'intéresser à la structure du domaine spectral en obser-

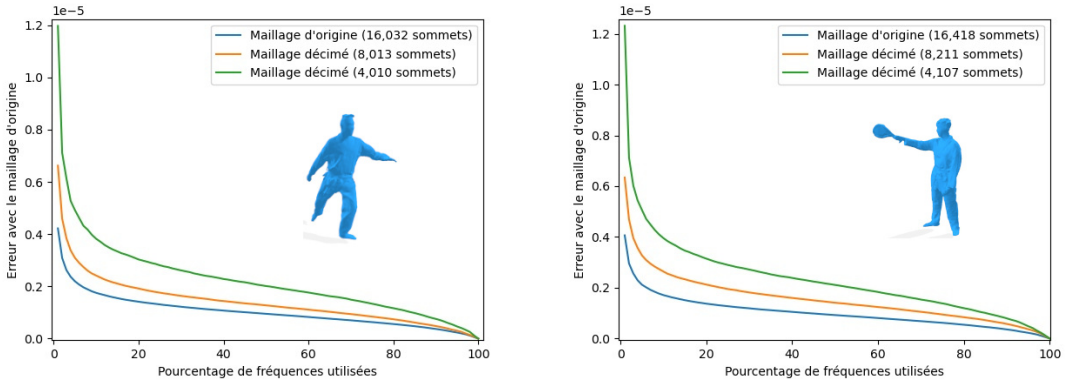


FIGURE 3.10 – Erreur de reconstruction en fonction du pourcentage de fréquences utilisées. Pour chaque maillage (décimé ou non), l'erreur exprime la distance moyenne entre les sommets correspondants de celui d'origine et celui reconstruit après une transformée directe puis inverse.

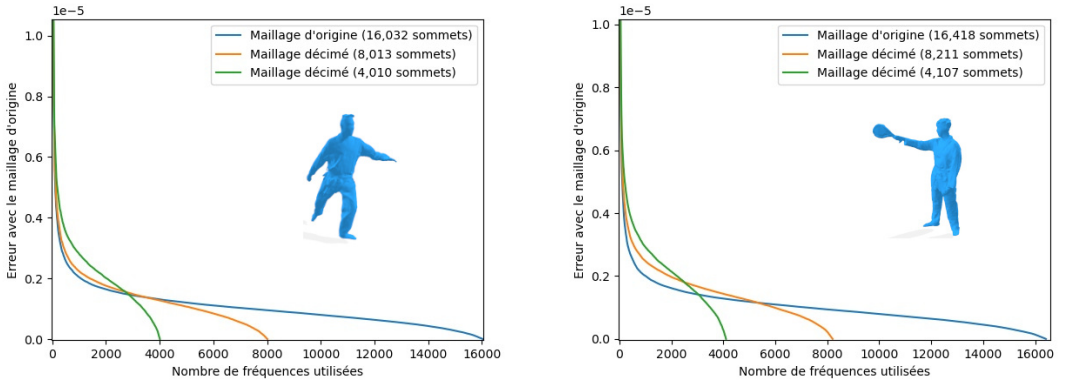


FIGURE 3.11 – Erreur de reconstruction en fonction du nombre de fréquences utilisées. Pour chaque maillage (décimé ou non), l'erreur exprime la distance moyenne entre les sommets correspondants de celui d'origine et celui reconstruit après une transformée directe puis inverse.

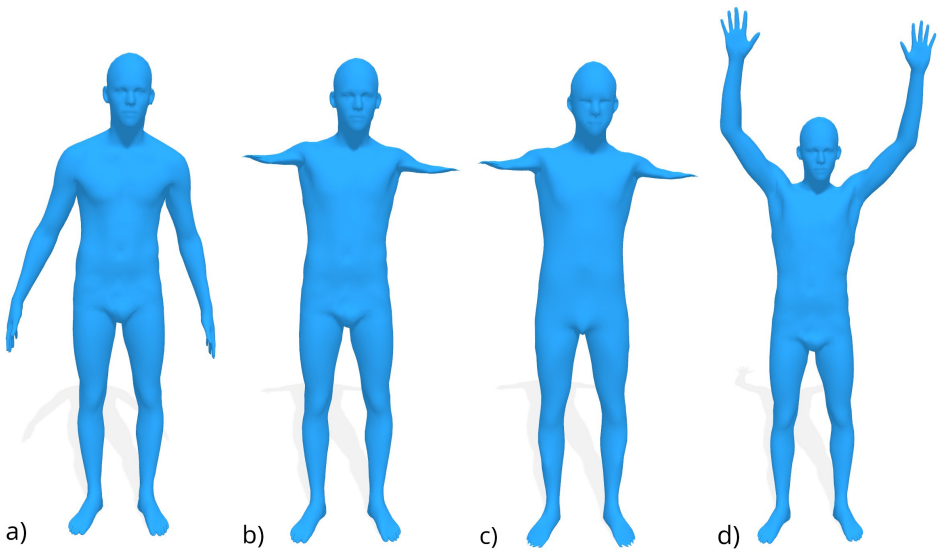


FIGURE 3.12 – Exemple d'interpolation entre les deux maillages a) et d). b) est une interpolation dans l'espace spatial et c) est une interpolation dans l'espace spectral en utilisant 512 fréquences sur les 6 890 disponibles.

vervant le comportement d'une interpolation dans ce dernier. Soient deux maillages M_0 et M_1, ayant la même connectivité, les géométries P_0 et P_1 dans le domaine spatial, et pour coefficients spectraux C_0 et C_1. Deux interpolations linéaires sont possibles en utilisant les formules $P_0 * (1 - a) + P_1 * a$ et $C_0 * (1 - a) + C_1 * a$. Après avoir appliqué une transformation inverse aux coefficients interpolés, il est possible de comparer le résultat de l'interpolation de la géométrie directement et des coefficients spectraux. La figure 3.12 présente ce résultat avec une valeur de $a = 0.5$. Le domaine des fréquences se comporte de la même manière que le domaine spatial, dû au fait que les transformations spectrales directe et inverse sont des transformations linéaires, à l'exception de la suppression des hautes fréquences et donc des détails visible sur la figure 3.12.

L'interpolation dans le domaine spectral de deux surfaces différentes avec une même topologie est possible puisque la matrice Laplacienne est construite depuis la connectivité seulement. L'opérateur de Laplace-Beltrami présenté ensuite est quant à lui construit depuis la géométrie du maillage.

3.2.2 Opérateur de Laplace-Beltrami

Après les résultats présentés par Taubin et al. [129], d'autres travaux [109, 100, 77] ont suggéré d'utiliser un processus différent pour construire l'opérateur : le schéma classique utilisant des cotangentes (*classic cotangent scheme*) [109]. Ainsi, l'opérateur de Laplace-Beltrami dépend de la géométrie de chaque maillage. Il est construit comme $L = A^{-1}W$ avec :

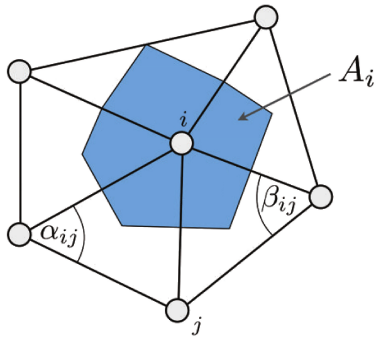


FIGURE 3.13 – Angles et aires de Voronoi. L'image provient de http://rodolphe-vaillant.fr/entry/20/compute-harmonic-weights-on-a-triangular-mesh.

$$A = \begin{pmatrix} A_0 & & \\ & \ddots & \\ & & A_{n-1} \end{pmatrix}, W = \begin{pmatrix} W_0 & & \\ & \ddots & \\ & & W_{n-1} \end{pmatrix} \tag{3.11}$$

où A_i est l'aire du polygone formé par les centres des cellules du diagramme de Voronoï des triangles situés autour du sommet i et $W_i = \frac{1}{2*A_i} \sum_{j \in N(i)} (cot\alpha_{ij} + cot\beta_{ij})$, voir figure 3.13.

L'opérateur L est aussi une matrice de taille $n * n$, avec n le nombre de sommets du maillage, et le spectre peut être calculé en résolvant le problème aux valeurs propres généralisé : $-W\phi^i = \lambda_i A\phi^i$. Comme pour la décomposition utilisée pour la matrice Laplacienne, k valeurs propres et vecteurs propres sont obtenus qui ont un comportement similaire : oscillations différentes en fonction de la fréquence associée et possibilité de projection pour obtenir les coefficients spectraux. Comme pour la version combinatoire de l'opérateur, les vecteurs propres présentent un comportement harmonique et peuvent être considérés comme les modes de vibration ou les harmoniques de la surface (voir figure 3.7), et les valeurs propres peuvent être considérées comme les fréquences.

La construction de l'opérateur de Laplace-Beltrami le rend invariant aux déformations presque isométriques et aux changements de discrétisations. Cela signifie que pour une surface représentant un sujet (une personne, une identité), les vecteurs et valeurs propres calculés seront invariants à la discrétisation ou topologie utilisée et à la pose dans laquelle le sujet est capturé. Cette propriété est surtout assurée dans les basses fréquences [105], les hautes fréquences pouvant présenter des inversements d'ordre ou de signe. C'est pourquoi cette base est utilisée dans le processus des cartes fonctionnelles (*Functional Maps*) dont plus de détails seront donnés dans la section 3.4.

Maintenant, voyons pourquoi l'utilisation du Laplacien topologique est plus adaptée dans notre cas. La figure 3.14 présente une comparaison de reconstruction en

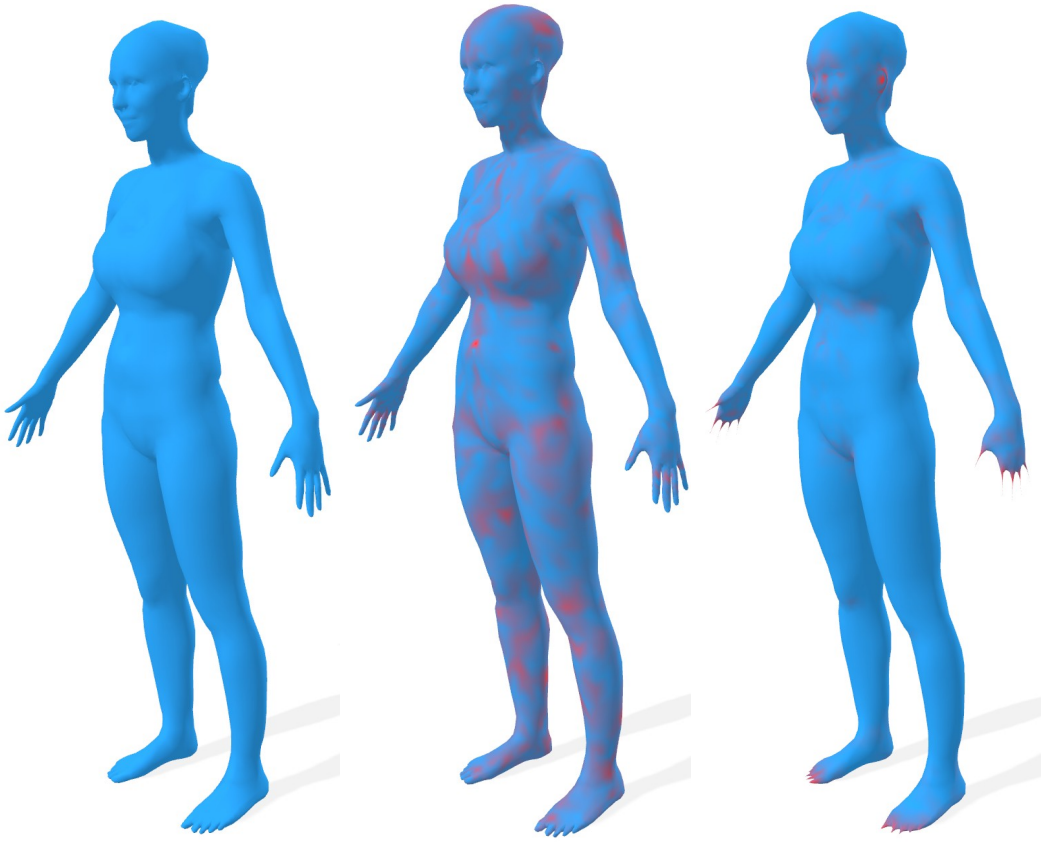


FIGURE 3.14 – Comparaison de reconstructions en utilisant 2 048 coefficients sur les 6 890 disponibles. À gauche : le maillage d'origine, au milieu : en utilisant la matrice Laplacienne, à droite : en utilisant l'opérateur de Laplace-Beltrami. La couleur indique la distance entre les sommets correspondant du maillage d'origine et celui reconstruit, le bleu étant une distance nulle et le rouge une distance maximum. Dans ce cas, la matrice Laplacienne permet de mieux reconstruire les détails (mains, visage, pieds...) mais introduit globalement plus d'erreurs sur le corps.

utilisant les deux opérateurs introduits jusqu'ici en utilisant le même nombre de coefficients (2 048). Il est clair que l'opérateur ne prenant en compte que la topologie a une meilleure capacité de reconstruction, notamment au niveau du visage, des mains et des pieds. Les maillages reconstruits en utilisant un nombre limité de fréquences seront de meilleure qualité en utilisant cet opérateur, ce qui représente un premier avantage.

Ensuite, on voit que projeter un signal issu d'une surface sur les vecteurs propres d'un opérateur est similaire à une transformée de Fourier. Néanmoins, une différence à prendre en compte est que dans le cas 1D, n'importe quel signal peut être projeté sur la même base à partir du moment où ils ont la même longueur puisque la connectivité sera toujours la même. En revanche, dans le cas d'une surface 2D plongée dans un espace 3D, la connectivité peut être changeante entre deux maillages. Cela donne un calcul de l'opérateur différent en fonction de la connectivité et de la géométrie dans le cas de l'opérateur de Laplace-Beltrami.

Pour deux surfaces différentes mais ayant la même connectivité, la base de décomposition spectrale de la matrice Laplacienne sera identique, alors que la base de décomposition spectrale de l'opérateur de Laplace-Beltrami sera différente. En revanche, pour deux surfaces identiques mais représentées par une discrétisation différente, la décomposition du Laplacien topologique sera différente alors que celle de l'opérateur de Laplace-Beltrami, en raison de ses propriétés, est censée être identique puisque l'opérateur est robuste aux changements de topologie. Ainsi, en considérant que notre objectif est d'étudier des surfaces dans un unique domaine spectral, l'utilisation d'un opérateur topologique ou géométrique dépend de la situation dans laquelle on se trouve :

- cas d'une base de données composée d'une seule triangulation comme SMPL :
  - le Laplacien topologique sera identique pour tous les maillages et la décomposition en éléments propres ne sera effectuée qu'une seule fois.
  - l'opérateur de Laplace-Beltrami sera différent pour tous les maillages et la décomposition en éléments propres sera effectuée pour chaque échantillon. Il est tout de même possible de ne calculer qu'une seule fois la décomposition en éléments propres sur un maillage de référence, mais la projection d'autres maillages sur les vecteurs propres obtenus depuis cette dernière introduira une erreur. Il vaut donc mieux utiliser le Laplacien topologique.
- cas d'une base de données composée de plusieurs triangulations comme Kinovis :
  - le Laplacien topologique sera différent pour chaque maillage et la décomposition spectrale devra être recalculée pour chaque échantillon, rendant le processus long et difficilement utilisable en pratique. De plus, les espaces dans lesquels seront projetées les géométries seront différents, et l'étude des coefficients spectraux n'aura pas de sens sans synchronisation spectrale.
  - l'opérateur de Laplace-Beltrami sera aussi différent pour chaque maillage. Ensuite, si pour une même identité cet opérateur est censé être robuste

aux déformations presque isométriques, le nombre de sommets des discrétisations sera différent et la projection directe de la géométrie sur une autre base ne sera pas possible en raison des dimensions différentes des matrices. Dans ce cas, il faudra calculer des synchronisations spectrales comme nous le verrons ensuite (voir section 3.4), ou remailler tous les échantillons de la même manière pour se retrouver dans la situation d'une connectivité commune.

Dans notre cas, l'utilisation du Laplacien topologique est donc plus adaptée, surtout en raison de l'utilisation de la base de données AMASS utilisant le formalisme SMPL. Pour des travaux futurs, il est tout de même intéressant de voir comment il est possible de synchroniser des domaines spectraux. Pour pouvoir calculer la ressemblance entre les vecteurs propres de deux maillages différents, nous pouvons utiliser un processus introduit il y a une dizaine d'années : les cartes fonctionnelles [105] (*Functional Maps* ou FM). Leur fonctionnement sera expliqué et des expériences seront présentées après avoir abordé d'autres opérateurs, et surtout après avoir abordé la notion de descripteurs utiles à leur calcul.

3.2.3 Autres opérateurs

Selon la manière dont est construit l'opérateur, il possèdera des caractéristiques le rendant invariant à certaines transformations ou le rendant capable de capturer certains types d'information. Le Laplacien topologique est invariant à toute transformation pour une même connectivité : les valeurs et vecteurs propres seront alors identiques pour n'importe quelle surface représentée, et les coefficients spectraux seront fonction de la géométrie. En revanche, les valeurs et vecteurs propres seront différents si une même surface est représentée par une connectivité différente. L'opérateur de Laplace-Beltrami est, quant à lui, invariant aux transformations presque isométriques : les valeurs et vecteurs propres sont censés être identiques pour une même surface représentée dans une pose différente avec la même connectivité. Pour deux surfaces identiques mais avec une discrétisation différente, les descripteurs qui en découlent sont censés être identiques. Cela le rend très utile pour des applications de mise en correspondance de surfaces (*shape matching*) comme dans le processus de calcul des FM, à défaut de ne pas être complètement robuste à des changements de connectivité comme nous le verrons dans la section 3.4. Pour cette raison, l'opérateur de Laplace-Beltrami est dit intrinsèque, signifiant qu'il ne dépend pas de l'immersion de la surface dans l'espace 3D, c'est à dire de la pose ou de la discrétisation de l'objet. D'autres opérateurs ont été introduits dans la littérature (voir l'état de l'art de Wang et al. [142]) afin qu'ils soient capables de capturer une information extrinsèque.

Opérateur de Dirac. Crane et al. [37] ont introduit l'opérateur de Dirac. En calculant des valeurs qui encodent des remises à l'échelle et des rotations, ils décrivent la géométrie extrinsèque d'une déformation en utilisant des quaternions. Krostikov et al. [68] ont aussi travaillé sur l'opérateur de Dirac et montrent qu'il détecte la courbure principale d'une surface et capture mieux les détails des formes que le

Laplacien lié à la courbure moyenne. L'opérateur de Dirac est récent et la littérature le concernant est moins abondante que celle concernant le Laplacien. Il est aussi coûteux en performance : des améliorations futures pourraient le rendre plus accessible.

Opérateur de Dirac relatif. En 2017, Liu et al. [85] utilisent un nouvel outil, l'opérateur de Dirac relatif (*Relative Dirac*), qui mène à une famille d'opérateurs allant de complètement intrinsèque jusqu'à extrinsèque. Ye et al. [149] ont aussi travaillé sur ce même type d'opérateur, mais cette fois-ci unifié. La discrétisation de ce Dirac, à la fois intrinsèque et extrinsèque, doit encore faire ses preuves dans plusieurs applications afin d'être approuvée. Il reste encore à comprendre son rôle et quelles informations géométriques elle peut capturer dans différentes situations. Hoffmann et al. [61] donnent plus de détails mathématiques.

Opérateur de Steklov (Dirichlet-to-Neumann). Wang et al. [141] ont travaillé sur l'opérateur de Dirichlet-to-Neumann, aussi appelé Steklov, et proposent une alternative extrinsèque au Laplacien en amenant des justifications théoriques. Cet opérateur fournit une information volumétrique sur la géométrie du maillage. Il est conclu que leur méthode a besoin d'une optimisation pour améliorer le temps de calcul.

L'information contenue dans ces opérateurs extrinsèques est supérieure à celle capturée par le Laplacien topologique ou l'opérateur de Laplace-Beltrami, et il serait intéressant d'essayer de s'en servir pour développer un processus plus adapté aux différentes poses possibles du corps humains. Néanmoins, comme pour l'opérateur de Laplace-Beltrami, un calcul dudit opérateur serait nécessaire pour chaque échantillon, ralentissant fortement la vitesse du programme. De plus, comme dit précedemment, leur calcul reste coûteux en temps, et des recherches plus poussées sur la manière dont ils sont calculés seraient nécessaires pour qu'ils soient intégrés à notre processus, ce qui est hors du cadre de cette thèse.

À partir de ces opérateurs, il est possible de calculer des signatures, aussi appelées descripteurs, qui vont permettre de représenter les sommets d'une surface en utilisant des vecteurs de plus haute dimension et contenant plus d'information.

3.3 Signatures / descripteurs

De nombreuses applications concernant l'analyse de surfaces ont besoin que ces dernières soient représentées dans un format plus informatif en raison de l'expressivité limitée des seuls sommets modélisant ces surfaces. Un descripteur, aussi appelé signature, est un vecteur de dimension supérieure qui permet de décrire une surface dans un espace compact étant donné un point de cette dernière. Un état de l'art des descripteurs disponibles a été rédigé par Rostami et al. [117].

Dans la suite de cette section, λ_i et ϕ^i représentent respectivement les valeurs et vecteurs propres de l'opérateur de Laplace-Beltrami (voir section 3.2 pour plus de détails).

Signature de Point Globale. Rustamov et al. [118] ont proposé un descripteur de caractéristique locale appelé Signature de Point Globale (*Global Point Signature* ou GPS). Cette signature est basée sur les vecteurs propres de l'opérateur de Laplace-Beltrami en chaque point de la surface. Elle est invariante sous des déformations isométriques de la forme mais souffre d'un problème de commutation des fonctions propres lorsque les valeurs propres associées sont proches les unes des autres. Elle est calculée comme suit :

$$GPS(p) = (\frac{1}{\sqrt{\lambda_1}}\phi_1(p), \frac{1}{\sqrt{\lambda_2}}\phi_2(p), \frac{1}{\sqrt{\lambda_3}}\phi_3(p), \dots,) \tag{3.12}$$

où λ_i est la i-ème valeur propre, ϕ_i le i-ème vecteur propre et p un point de la surface discrétisée.

Signature du noyau de la chaleur. La signature du noyau de la chaleur (*Heat Kernel Signature* ou HKS) est introduite par Sun et al. [126]. Elle est définie par :

$$HKS(p) = \sum_{i=0}^{\infty} \exp^{-\lambda_i t} \phi_i^2(p) \tag{3.13}$$

En général, i ne va pas aller jusqu'à l'infini mais plutôt jusqu'à un nombre limité de fréquences. HKS mesure l'information du voisinage des points et a été utilisée dans des applications de comparaison de surface. Cette mesure assigne à chaque point de la surface un vecteur qui représente à chaque pas de temps t le montant de chaleur retenue en ce point. Ainsi, la diffusion de la chaleur sur la surface peut être calculée graĉe à cette matrice de taille $n * t$, n étant le nombre de sommets. Cette signature est construite sur les vecteurs propres de l'opérateur de Laplace-Beltrami et permet de mesurer l'information géométrique intrinsèque d'une forme. Elle est robuste aux déformations presque isométriques et stable contre du potentiel bruit contenu dans l'approximation de la surface. Mais une analyse de Fourier montre que HKS est fortement dominée par l'information des basses fréquences qui correspondent aux propriétés macroscopiques d'une surface. Deux extensions connues sont *Scale Invariant HKS* [18] et *Volumetric HKS* [115].

Signature du noyau d'onde. La signature du noyau d'onde (*Wave Kernel Signature* ou WKS) a été introduite par Aubry et al. [7]. Elle est tirée de la mécanique quantique pour capturer des détails multi-échelles de formes 3D. L'équation de Schrödinger est utilisée pour représenter le mouvement de particules mécaniques quantiques sur la surface. Comme pour HKS, chaque point de la surface sera représenté par un vecteur, mais cette fois-ci avec une probabilité de mesurer une particule quantique en ce point avec une fonction de distribution d'énergie initiale. WKS est donc paramétrée en utilisant des fréquences plutôt que le temps, et peut contrôler l'accès aux hautes fréquences. Cette signature permet une description plus précise que HKS. Elle est aussi intrinsèque, informative et stable.

Signature de point locale. La signature de point locale (*Local Point Signature* ou LPS) est introduite par Wang et al. [139]. Elle est aussi basée sur l'opérateur de

Laplace-Beltrami, est plus résiliente que HKS ou WKS aux transformations et aux changements d'échelle, de rotation et de connectivité en combinant les avantages d'un descripteur intrinsèque et extrinsèque. Mais le calcul de ce descripteur, qui a besoin d'une décomposition pour chaque sommet, est très coûteux en temps de calcul.

Signature de décomposition de l'énergie d'ondelettes. La signature de décomposition de l'énergie d'ondelettes (*Wavelet Energy Decomposition Signature* ou WEDS) a été proposée par Wang et al. [140]. Comparé à LPS, WEDS utilise des ondelettes pour capturer des informations à la fois locales et globales, ce qui est plus efficace.

Ces signatures découlent de l'opérateur de Laplace-Beltrami et sont donc intrinsèques, les amenant à être utilisées dans des applications qui concernent de la mise en correspondance de surfaces (*shape matching*) comme les cartes fonctionnelles (*Functional Maps*).

3.4 Cartes fonctionnelles (*Functional Maps*)

Comme nous l'avons vu précédemment, une base de données peut être faite de maillages n'ayant pas la même connectivité ni le même nombre de sommets. Lorsque les surfaces contenues dans cette base de données représentent des objets qui peuvent être mis en correspondance, comme des corps humains, une tâche intéressante est de pouvoir définir des liens entre des sommets ou des zones des différents objets (tête, bras, mains, etc...). Cette problématique est depuis longtemps étudiée dans le domaine de l'informatique graphique et de nombreuses méthodes ont été proposées pour la résoudre.

Ovsjanikov et al. [105] ont introduit la notion de cartes fonctionnelles (*Functional Maps* ou FM). Définir des liens entre deux surfaces équivaut à définir une carte sommet à sommet entre ces deux dernières. Néanmoins, les méthodes basées sur le calcul dans le domaine spatial d'une carte sommet à sommet souffrent en général d'une sensibilité au bruit ou de la difficulté à sélectionner une échelle appropriée. Les FM résolvent cette problématique en considérant une mise en commun plus générale entre des fonctions définies sur ces surfaces en passant par le domaine spectral.

Le processus est le suivant. Étant données deux surfaces dont on cherche les correspondances, l'opérateur de Laplace-Beltrami est d'abord calculé depuis la géométrie et la topologie de ces deux dernières, puis est décomposé en valeurs propres et vecteurs propres, donnant deux représentations spectrales correspondant aux deux surfaces depuis lesquelles des descripteurs (WKS) sont ensuite calculés. L'opérateur de Laplace-Beltrami est utilisé puisqu'il est censé être invariant aux transformations presque isométriques et aux changements de connectivité. L'objectif est ensuite de calculer une matrice carrée C (représentant la FM) qui va minimiser plusieurs énergies, la principale étant la différence entre les descripteurs projetés dans le domaine spectral et synchronisés grâce à la matrice. C permet finalement de calculer une

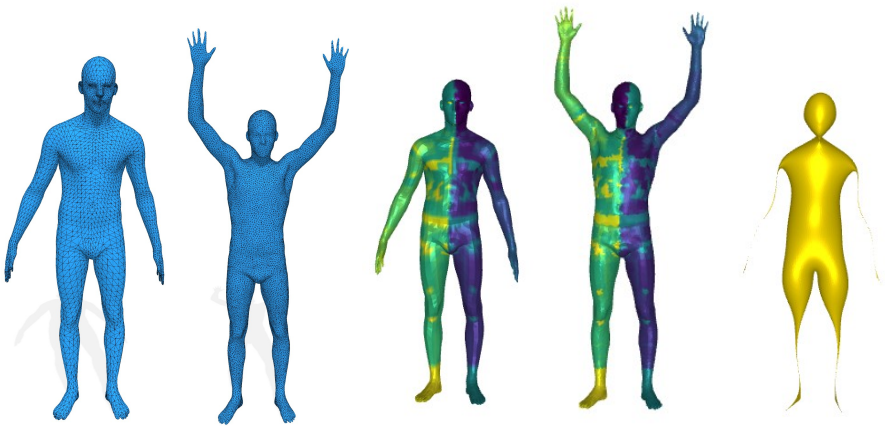


FIGURE 3.15 – Les deux premiers maillages sont ceux étudiés pour calculer la carte fonctionnelle. Le troisième est représenté en utilisant une couleur pour chaque sommet en fonction de l'indice de ce dernier. Le quatrième représente le résultat de la carte point à point calculée. Le dernier représente le premier maillage exprimé avec la connectivité du deuxième après une synchronisation spectrale. Une inversion au niveau du cou entraîne une mauvaise synchronisation spectrale.

carte sommet à sommet entre les deux surfaces étudiées, et surtout de pouvoir synchroniser les domaines spectraux de deux surfaces n'ayant pas la même connectivité ni le même nombre de noeuds.

Dans la théorie, cet outil pourrait être extrêmement utile afin de pouvoir étudier des bases de données composées de maillages n'ayant pas la même connectivité dans un espace spectral commun. Néanmoins, les caractéristiques de l'opérateur de Laplace-Beltrami et son implémentation font que son utilisation reste encore aujourd'hui difficile à appliquer. En effet, bien que l'opérateur de Laplace-Beltrami soit censé être invariant aux déformations presque isométriques, il est tout de même relativement sensibles à ces mouvements, surtout dans les hautes fréquences, en présentant des inversions ou des changements de vecteurs propres. Par ailleurs, l'implémentation des cartes fonctionnelles est fortement renforcée par l'utilisation de repères en désignant des sommets correspondants sur les deux surfaces, nécessitant une potentielle annotation d'utilisateurs.

Dans la figure 3.15, un calcul de carte fonctionnelle est présenté, réalisé à l'aide du code disponible à cette adresse : https://github.com/RobinMagnet/pyFM. Le premier maillage est extrait de FAUST [13] et correspond à la paramétrisation SMPL. Le deuxième maillage est aussi extrait de FAUST et représente la même identité que le premier dans une pose différente mais a été remaillé uniformément. Les deux surfaces n'ont donc pas le même nombre de sommets ni la même connectivité. Après un calcul de la carte fonctionnelle et de la carte sommet à sommet correspondante, nous pouvons voir une asymétrie apparaître au niveau du cou. Cela donne une mauvaise synchronisation spectrale, et donc une mauvaise expression de la géométrie du premier maillage avec la connectivité du second, visible sur le dernier maillage de la

figure.

De plus, le calcul de cette carte fonctionnelle est conséquent : 1 à 2 minutes en utilisant une configuration locale (voir 1). Cela serait inenvisageable pour une base de données comprenant beaucoup d'échantillons. Par ailleurs, la synchronisation spectrale est réalisée sur seulement 50 coefficients spectraux, ce qui amène à un maillage généré après synchronisation ne présentant pas les détails contenus dans les hautes fréquences. Calculer cette synchronisation en utilisant plus de fréquences amènerait à des temps de calcul encore plus longs.

De nombreux travaux ont été publiés depuis l'introduction des cartes fonctionnelles essayant d'améliorer la qualité des liens trouvés entre les surfaces, en utilisant par exemple des modèles d'apprentissage profond. Même si aujourd'hui encore ce processus ne permet pas de trouver des correspondances parfaites, on peut imaginer que dans le futur soient utilisés d'autres opérateurs ou descripteurs permettant un calcul plus efficace de ces correspondances.

3.5 Applications

Dans cette section, nous allons explorer l'exploitation du domaine spectral dans la littérature en ce qui concerne les maillages triangulaires. Comme indiqué dans la section 1.1.3, l'analyse spectrale de maillages triangulaires est étroitement liée aux avancées dans le domaine de l'analyse spectrale des graphes. Cependant, de nombreuses recherches se sont concentrées spécifiquement sur l'analyse des maillages triangulaires. Dans cette section, nous discuterons de l'utilisation de l'information spectrale sans recourir à l'apprentissage profond, tandis que dans la section consacrée aux travaux connexes sur l'apprentissage profond statique (section 4.2), nous examinerons les techniques de génération de maillages utilisant ce domaine spectral couplé à des réseaux de neurones. Des études sont disponibles concernant cet état de l'art : [124, 153, 142]. L'analyse de maillage discret peut être considérée comme une spécialisation de l'analyse de graphes. Depuis l'introduction du Laplacien topologique [129] et de l'opérateur de Laplace-Beltrami [109, 100, 77], de nombreuses applications ont été présentées qui utilisent un opérateur pour étudier une surface.

Certaines utilisent directement les valeurs propres obtenues après une décomposition de l'opérateur de Laplace-Beltrami, par exemple pour de l'indexation de surfaces [153]. Cependant, étant donnée l'invariance aux transformations presque isométriques de l'opérateur, il est difficile de reconnaître correctement une forme 3D en utilisant uniquement les valeurs propres (*one cannot hear the shape of a drum* [54]). Des méthodes plus récentes proposent tout de même des solutions en utilisant des procédures numériques [36] ou des réseaux de neurones [94, 95, 107].

D'autres méthodes utilisent les vecteurs propres, en plus des valeurs propres, pour la segmentation de maillages, la reconstruction de surfaces, la détection de symétrie et le partitionnement (*clustering*) notamment [153]. Dans ce cas, il est par

exemple possible de calculer des descripteurs, comme vus dans la section 3.3, afin d'évaluer des correspondances entre des formes représentant une même surface dans différentes poses, comme les cartes fonctionnelles [105]. Le processus les calculant a ensuite été amélioré en utilisant l'apprentissage profond [84, 42]. Néanmoins, comme pour l'utilisation des seules valeurs propres, il est difficile de récupérer une surface depuis un descripteur.

Finalement, quelques méthodes utilisent directement la projection des sommets d'une surface sur les vecteurs propres pour la compression ou le tatouage numérique [153], ou pour filtrer/renforcer certaines fréquences [134]. Cette projection de la géométrie sur les vecteurs propres, donnant des coefficients spectraux, a peu été utilisée dans la littérature à l'exception de travaux appliquant des convolutions à des graphes ou à des surfaces en utilisant le théorème de convolution (une convolution dans le domaine spatial est une multiplication dans le domaine spectral). Mais ces méthodes réalisent tout de même des opérations dans le domaine spatial, les rendant toujours dépendantes de la résolution et de la discrétisation des surfaces étudiées.

3.6 Conclusion

Dans ce chapitre, nous avons vu comment le traitement spectral de maillages triangulaires est lié à la transformée de Fourier et au Laplacien, comment l'introduction de plusieurs opérateurs dans la littérature permettent d'étudier des surfaces dans le domaine spectral, et comment la projection des sommets d'un maillage sur les vecteurs propres d'un opérateur permettent d'obtenir les coefficients spectraux, représentant la géométrie de l'objet étudié d'une manière compactée. Dans le prochain chapitre, nous allons voir comment ces coefficients spectraux peuvent être exploités afin d'entraîner des modèles d'apprentissage profond.

Chapitre 4

Apprentissage d'une représentation adaptée à des maillages 3D

L'objectif de ce chapitre est de présenter le processus qui nous a permis d'aboutir à une architecture rendant possible la publication [74] qui concerne le traitement statique de l'information. Après avoir introduit la problématique, nous examinerons les travaux similaires de l'état de l'art. Ensuite, nous exposerons les expériences préliminaires réalisées afin de montrer quelles ont été les méthodes utilisées pour aboutir à notre architecture. Finalement, nous présenterons des résultats comparant notre méthode à celles de l'état de l'art.

4.1 Introduction

Le succès des méthodes d'apprentissage profond ces dix dernières années et la disponibilité accrue de bases d'entraînement issues de captures de formes en mouvement rendent nécessaire l'introduction de modèles capables d'analyser, comprendre et générer ce nouveau type de données résidant dans un domaine à 4 dimensions. Avant cela, il est nécessaire d'orienter le travail de thèse vers le développement d'algorithmes traitant des échantillons statiques sans prendre en compte l'information temporelle.

Les couches les plus courantes de techniques d'apprentissage en profondeur telles que les convolutions, l'agrégation (*pooling*) et le sur-échantillonnage (*upsampling*) permettent de généraliser les poids appris à des données non vues afin de classer, segmenter ou reconstruire/générer à partir d'un espace latent, comme nous l'avons vu dans le premier chapitre. Par exemple, les autoencodeurs sont des outils utiles pour extraire des caractéristiques importantes des échantillons observés de manière non supervisée. Cette architecture est celle que nous utilisons principalement dans nos expériences puisque les données sont en général non annotées. Il est donc nécessaire de développer des algorithmes capables de traiter l'information de manière non supervisée. De plus, en forçant les données en entrée à passer par un goulot d'étranglement, nous espérons obtenir réseau capable de construire un espace latent représentant fidèlement la variété des échantillons d'entrée comme toutes les poses

possibles d'un corps humain par exemple. Cette propriété intéressante offre un moyen de générer de nouvelles données par interpolation dans cet espace de plus petite dimension créé par le réseau. Lors de l'apprentissage sur des images, l'utilisation des convolutions est bien définie, puisque le domaine a une structure euclidienne. Mais lors de l'apprentissage sur des graphes ou des surfaces, puisque l'information se situe maintenant dans un domaine non-euclidien, l'application des architectures connues utilisant des convolutions, de l'agrégation ou du sur-échantillonnage devient difficile.

La première idée serait de traiter directement la sortie brute des modélisations 3D, mais les méthodes prenant des nuages de points en entrée manquent souvent de l'information disponible dans une connectivité lorsque les objets étudiés sont des formes humaines qui peuvent être déformées avec des transformations presque isométriques. La deuxième idée est d'ajouter des liens à ces nuages de points afin de les transformer en maillages triangulaires avec beaucoup de sommets, puis en les donnant à des réseaux de neurones. Les méthodes existantes ne sont pas, à l'heure actuelle, en mesure de traiter efficacement des données 3D se trouvant sur une grille non structurée avec une connectivité changeante, en particulier lorsqu'il s'agit de corps humains. Ainsi, la plupart du temps, les modèles étudiés nécessitent un type d'entrée plus simplifié comme des maillages avec une connectivité constante et un petit nombre de sommets, comme ceux générés par le formalisme SMPL [87] qui paramétrise une surface par des angles d'articulation 3D et un espace de faible dimension.

Les méthodes de l'état de l'art traitent ces maillages à connectivité constante soit dans le domaine spatial, soit dans le domaine spectral. En général, elles sont contraintes par le nombre de sommets, ce qui rend l'apprentissage coûteux et long lorsque les maillages ont beaucoup de nœuds. Nous visons à résoudre ces problèmes en utilisant des méthodes d'analyse spectrale d'une manière différente : nous cherchons à ne prendre en compte que l'information géométrique importante contenue dans les basses fréquences des coefficients spectraux afin d'alléger la quantité de données fournies d'une part, et à réaliser la totalité de l'entraînement dans le domaine spectral afin de réduire le temps d'apprentissage d'autre part. Un tel modèle permettra ensuite de créer une représentation adaptée au corps humain.

4.2 État de l'art

Afin de transférer les techniques d'apprentissage profond aux données 3D, les premières méthodes utilisent des voxels ou des images. Les méthodes utilisant des nuages de points ne reposent que sur les coordonnées cartésiennes de chaque point, alors que les méthodes utilisant des graphes ou maillages exploitent cette connectivité, et peuvent être utilisées dans deux domaines différents : spatial ou spectral. Les lecteurs intéressés peuvent se référer aux synthèses [146, 17, 1, 145, 148, 144, 24, 137]. Ici, nous nous concentrerons sur les nuages de points et maillages triangulaires.

4.2.1 Nuages de points

La plupart des méthodes basées sur les nuages de points proviennent des travaux de Charles et al. [28] (PointNet) et Qi et al. [111] (PointNet++). PointNet applique des convolutions à un nuage de points en traitant les noeuds un par un, signifiant que l'architecture ne traduit pas la structure locale de l'objet. PointNet++ applique plutôt des convolutions à des voisinages de noeuds, rendant plus accessible la structure locale de l'objet. Aumentado-Armstrong et al. [9] ont défini un VAE capable de séparer l'information intrinsèque et extrinsèque dans le domaine spectral de manière non supervisée. Cosmo et al. [35] ont introduit une contrainte géométrique forte en faisant en sorte que l'espace latent préserve les distances géodésiques calculées à l'aide des travaux de Crane et al. [38] sur les surfaces générées. Rakotosaona et al. [112] ont présenté une architecture à double autoencodeur, l'une extrayant les caractéristiques des nuages de points et l'autre extrayant les caractéristiques des longueurs d'arête. Ensuite, en liant les deux espaces latents, le réseau est capable d'interpoler de manière réaliste entre deux nuages de points en utilisant l'espace latent des longueurs d'arête. Toutes ces architectures utilisent PointNet comme encodeur et permettent d'étudier une topologie variable. Cela réduit les limitations dans le type de données d'entrée utilisées, mais les rend également incapables de rendre compte de la corrélation locale entre les sommets voisins : tout en ayant l'avantage d'utiliser des architectures simples et efficaces, leur capacité à révéler les caractéristiques locales des surfaces est souvent inférieure à celle des structures ayant accès à l'information de connectivité. Plus récemment, Thomas et al. [132] ont introduit KPConv qui traite chaque nœud d'un nuage de points en le pondérant en fonction des distances euclidiennes avec son voisinage, ayant ainsi plus d'informations sur la corrélation locale de la surface. Cependant, ces liens sont obtenus à l'aide d'un algorithme des K plus proches voisins, et des liens locaux erronés peuvent être trouvés. Les méthodes utilisant les architectures PointNet ou KPConv manquent alors d'informations de connectivité, en particulier lors du traitement de surfaces représentant des corps humains, et les méthodes exploitant des structures de type maillages triangulaires sont souvent plus performantes.

Le principal problème lors du transfert d'architectures d'apprentissage profond connues vers des maillages est que la grille représentant la surface manque de structure générale. Les architectures utilisant la connectivité peuvent être distinguées en deux catégories : celles utilisant des calculs dans le domaine spatial et celles utilisant des calculs dans le domaine spectral.

4.2.2 Domaine spatial

Dans le domaine spatial, la connectivité d'un maillage est complexe et irrégulière : il faut spécifier un ordre pour pouvoir parcourir les nœuds. Afin de pouvoir utiliser une fenêtre glissante sur les sommets, les méthodes basées dans le domaine spatial doivent définir des convolutions basées sur les relations entre ces nœuds en calculant des poids entre eux. L'opération de convolution sur ces graphes irréguliers a été définie de différentes manières. Premièrement, ces poids peuvent être

statiques, appris avec un calcul de prétraitement. Des techniques avec modèles de mélange/paramétrisation locale ont été utilisées : Masci et al. [99] ont appliqué des filtres à des patchs locaux représentés en coordonnées polaires géodésiques. Boscaini et al. [15] ont exploité la même idée en formulant des patchs intrinsèques locaux sur les maillages, et Fey et al. [49] ont utilisé des systèmes de pseudo-coordonnées locales prédéfinis sur les graphes. De plus, des techniques avec des convolutions en spirale ont été introduites dans les travaux préliminaires de Lim et al. [82] avec SpiralNet. Bouritsas et al. [16] ont utilisé des convolutions en spirale couplées à un autoencodeur basé sur l'architecture CoMA avec Neural3DMM [114]. Gong et al. [53] ont proposé une version améliorée avec SpiralNet++ [53]. Hanocka et al. [58] ont appliqué convolutions et sous-échantillonnages à des maillages en utilisant l'information contenue dans les arêtes. Huang et al. [62] ont introduit une fonction de coût basée sur des déformations aussi rigides que possible (*As-Rigid-As-Possible* ou ARAP) qui est combinée avec un réseau inspiré de FeastNet de Verma et al. [136]. Milano et al. [101] ont regroupé les caractéristiques des arêtes et des faces à l'aide de mécanismes d'attention. Ensuite, ces filtres peuvent être appris dynamiquement. Monti et al. [102] ont introduit MoNet, un modèle de mélange avec des poids appris. Verma et al. [136] ont présenté FeastNet, un opérateur de convolution de graphe permettant le calcul de correspondances dynamiques entre les poids du noyau et les nœuds voisins avec une connectivité arbitraire. Zhu et al. [155], inspirés de la méthode utilisant des spirales, ont proposé des convolutions pondérées par sommet.

4.2.3 Domaine spectral

Une autre voie de recherche utilise la théorie des graphes dans le domaine spectral. En transformant d'abord les nœuds d'un graphe à l'aide des vecteurs propres du Laplacien topologique (*Graph Laplacian*), une couche de convolution peut être écrite comme un produit. L'un des premiers travaux utilisant cette méthode a été introduit par Bruna et al. [20]. Mais les calculs induits par la transformée spectrale directe et inverse étant coûteux, Defferrard et al. [40] ont utilisé des polynômes de Chebyshev tronqués et Kipf et al. [65] n'ont utilisé que des polynômes de Chebyshev du premier ordre qui ont entraîné des processus plus rapides. Cela a abouti à un autoencodeur proposé par Ranjan et al. [114]. D'autres polynômes ont également été utilisés, comme ceux de Cayley par Levie et al. [76] et ceux de Zernike par Sun et al. [127]. Mais ces méthodes utilisent encore le domaine spatial pour le traitement des données d'entrée, ce qui ralentit le processus. Marine et al. [93], avec son extension [96], ont présenté un modèle qui apprend à reconstruire une forme 3D lorsqu'elle est uniquement représentée comme valeurs propres de l'opérateur de Laplace-Beltrami en entrée, tout en en créant un processus génératif qui utilise des interpolations ou des transferts de style. De même, Pegoraro et al. [107] génèrent une forme 3D à partir d'un modèle également basé sur des valeurs propres mais provenant d'un mélange de plusieurs opérateurs. Plus récemment, Sharp et al. [121] adoptent la diffusion pour obtenir un modèle indépendant de l'échantillonnage, de la résolution et de la représentation, mais seulement destiné à la classification ou la segmentation : de nouvelles expériences sont nécessaires pour rendre ce modèle génératif.

De manière générale, les méthodes de l'état de l'art sont soit trop lentes pour pouvoir traiter des bases de données denses, soit destinées à d'autres tâches que la génération de données. Nous visons à résoudre ces problèmes en utilisant des techniques d'analyse spectrale d'une manière différente : en tirant profit des propriétés du domaine des fréquences telles que la compaction de l'énergie et l'ordre des valeurs propres, notre objectif est de créer une représentation adaptée au corps humain qui soit indépendante du nombre de sommets des surfaces étudiées et de transférer les opérations simples utilisées en apprentissage profond à des maillages triangulaires.

4.3 Bases de données

Pour entraîner efficacement des modèles d'apprentissage profond, la première étape consiste à créer une base de données qui va jouer un rôle essentiel dans l'entraînement. Généralement, une grande quantité d'exemples est nécessaire pour comprendre des phénomènes complexes. Plus la quantité est importante, plus le modèle a de chances d'identifier des motifs significatifs et de bien généraliser lorsqu'il est confronté à de nouveaux échantillons. Pour comprendre l'impact de la taille de ces bases, nous allons en créer deux distinctes présentant des dimensions et des distributions différentes. Le même formalisme (SMPL [87]) sera utilisé pour garantir que les surfaces générées aient la même connectivité. Cette approche nous permettra d'observer comment la taille des bases de données influence l'apprentissage. Un nombre d'échantillons trop faible pourra potentiellement entraîner des problèmes tels que le surajustement, la mauvaise capacité à généraliser, et la non stabilité des performances. Ces impacts seront visualisés lors de l'entraînement de modèles sur ces deux ensembles.

DFaust. La première base de données est tirée des travaux de Bogo et al. [14]. Elle est composée de 41 220 maillages, soit 10 identités effectuant plusieurs actions. Les données sont divisées en 32 535 échantillons pour l'entraînement, représentant les 8 premières identités, et 8 685 échantillons pour le test représentant les 2 dernières identités. Cette base de données contient un faible nombre d'exemples mais permettra de tester en premier lieu des architectures sans entraînement trop long.

AMASS. La deuxième base de données utilisée correspond aux travaux de Mahmood et al. [89]. Elle est une unification de plusieurs autres bases de données en ajustant le modèle SMPL [87] aux capteurs de mouvement. Elle est en constante évolution et de nouvelles données issues de nouvelles captures sont intégrées au fur et à mesure. Nous suivons le protocole classique pour diviser les données : 1 pose sur 100 est sélectionnée dans la partie médiane à 90 % de chaque séquence (le début et la fin de l'animation ne sont pas pris en compte), ce qui donne 111 327 maillages pour l'entraînement et 10 733 pour le test. Les identités ne sont pas partagées entre les ensembles de données d'apprentissage et celui de test.

Il est possible, en utilisant SMPL, de contrôler l'orientation et l'identité des surfaces générées. Dans un premier temps, afin de faciliter la tâche des réseaux, les

surfaces créées seront toutes centrées et orientées dans la même direction. À la fin du chapitre, des expériences supplémentaires montreront comment il est possible de fournir cette information de translation et d'orientation. Pour rappel, des illustrations de maillages générés avec le formalisme SMPL et la manière dont ils sont obtenus sont présentés dans la section 2.1.

Afin de visualiser la différence de représentation, une version contenant l'information spatiale (les positions des sommets) et une autre version contenant l'information spectrale (les coefficients spectraux) sont créées. La figure 4.1 montre les moyennes et écarts types des deux représentations pour la base de donnée DFaust. Dans le domaine spatial, ces valeurs sont exprimées pour chaque sommet (6 890 au total). Les moyennes et écarts types sont relativement identiques pour tous les sommets, à l'exception de l'axe Y pour lequel les variations sont légèrement plus importantes en raison de mouvements verticaux. En revanche, dans le domaine spectral, les basses fréquences possèdent des moyennes et écarts types bien plus élevées que les fréquences moyennes. Cette différence d'amplitude est liée à la compaction de l'énergie : les coefficients spectraux basse fréquence correspondant à des zones globales de la surface ont une contribution plus élevée que ceux des hautes fréquences correspondant aux détails.

L'étude de la distribution d'un jeu de données est une étape importante dans la conception de modèles d'apprentissage profond. Elle permet de mieux comprendre la nature des données manipulées, ce qui peut aider à prendre des décisions lors de l'implémentation des modèles, notamment concernant la manière de calculer la fonction de coût. Un modèle d'apprentissage va donner plus d'importance à des valeurs qui ont une grande amplitude, aux détriment des autres ayant des amplitudes plus faibles. Dans le domaine spatial, il n'y a pas d'intérêt à donner plus d'importance à certains sommets. C'est pourquoi il est nécessaire de standardiser chaque coordonnées avant de calculer la fonction de coût afin de pouvoir donner une importance égale à chaque zone de la surface. En revanche, dans le domaine des fréquences, il sera plus judicieux de calculer cette fonction de coût sur des données non standardisées puisque des erreurs dans les hautes fréquences auront moins d'impact.

La figure 4.2 présente les moyennes et écarts types des deux représentations spatiale et spectrale pour la base de donnée AMASS. Étant donné que les maillages sont centrés et orientés, il n'y pas de grande différence entre DFaust et AMASS. En revanche, nous verrons dans la section 4.7 que si les informations de translation et de rotation sont utilisées, les deux bases de données sont différentes (AMASS contient une plus grande variété de mouvements).

Maintenant que des bases de données ont été créées, nous allons pouvoir entraîner plusieurs types de réseaux de neurones depuis ces dernières.

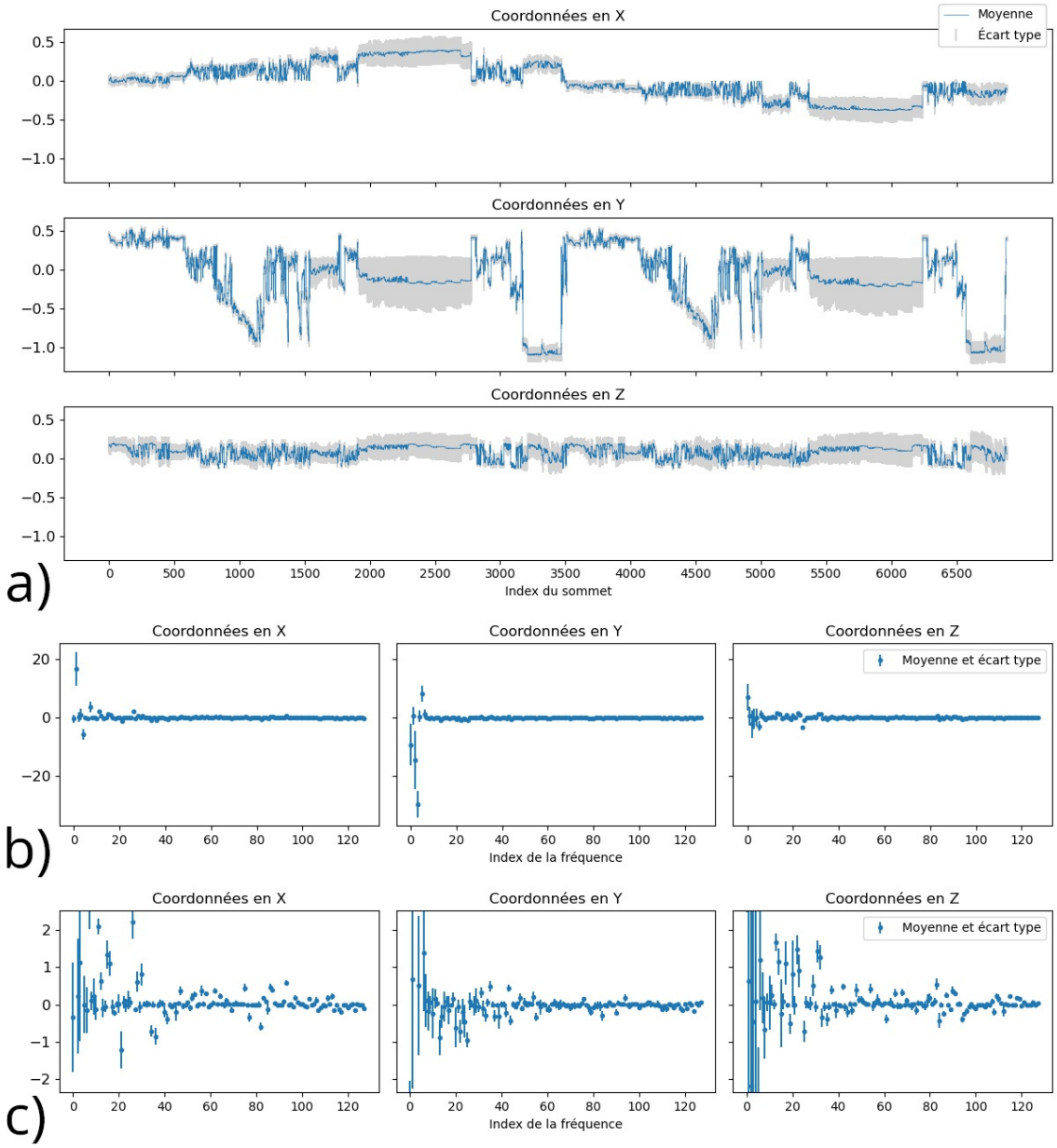


FIGURE 4.1 – Illustration des moyennes et des écarts types de la base de données DFaust. Les maillages sont centrés à l'origine et orientés dans la même direction. a) axes X, Y et Z représentés dans le domaine spatial. b) axes représentés dans le domaine spectral. c) axes représentés dans le domaine spectral en utilisant un zoom. Pour le domaine spectral, seulement 128 fréquences sont représentées.

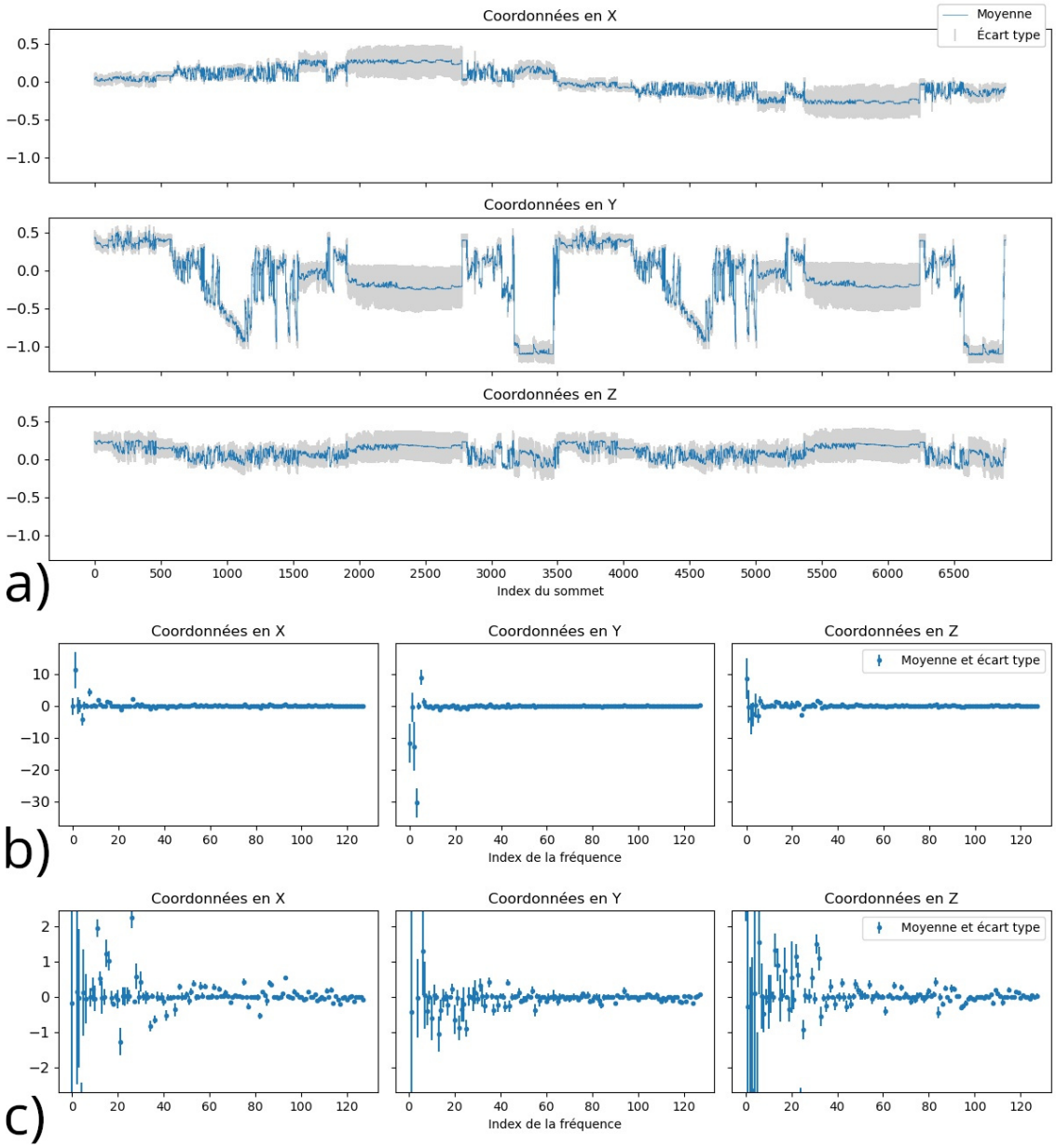


FIGURE 4.2 – Illustration des moyennes et des écarts types de la base de données AMASS. Les maillages sont centrés à l'origine et orientés dans la même direction. a) axes X, Y et Z représentés dans le domaine spatial. b) axes représentés dans le domaine spectral. c) axes représentés dans le domaine spectral en utilisant un zoom. Pour le domaine spectral, seulement 128 fréquences sont représentées.

4.4 Expériences préliminaires

Notre objectif fondamental est d'optimiser le processus d'entraînement des réseaux de neurones en éliminant les informations non essentielles qui se trouvent dans les détails superficiels des données afin de créer une représentation adaptée au corps humain. Il est bien établi qu'un signal peut être fidèlement approximé en utilisant un ensemble relativement restreint de coefficients spectraux comme présenté dans le chapitre dédié au traitement spectral des maillages (voir chapitre 3), en particulier ceux correspondant aux basses fréquences. Notre idée est de se baser sur cette notion de compaction énergétique des signaux. En alimentant un réseau de neurones exclusivement avec des coefficients spectraux porteurs d'une quantité significative d'énergie, nous pouvons résoudre plusieurs problèmes inhérents au traitement des maillages triangulaires. L'un de ces défis majeurs réside dans le nombre élevé de sommets et leur absence d'ordre naturel. La compaction spectrale permet de réduire considérablement la quantité de données qui est transmise au modèle, ce qui permet d'améliorer l'efficacité et la performance globale du réseau. De plus, l'ordre des données en entrée peut être défini de manière structurée en fonction des valeurs propres et donc de la différence d'amplitude entre les coefficients spectraux. Cela permet de créer un ordre significatif dans les données brutes, ce qui simplifie considérablement le processus d'analyse et d'interprétation. Dans cette section, nous verrons en détail les étapes qui nous ont permis d'implémenter une architecture capable de comprendre comment le corps humain est représenté et comment il peut se déformer. Des expériences préliminaires seront réalisées afin de comprendre le comportement de différentes architectures.

4.4.1 Autoencodeur complètement connecté

Afin de débuter les tests concernant la recherche d'architectures de modèles prenant en entrée des coefficients spectraux, nous nous sommes concentrés sur des réseaux utilisant des couches simples. La base de données n'étant pas annotée, et l'objectif étant de générer de nouveaux échantillons réalistes, un candidat intéressant est l'autoencodeur qui permet d'apprendre de manière non supervisée et qui permet de produire un espace latent depuis lequel il est possible d'interpoler des données. Pour les couches le composant, le choix le plus simple est un assemblage de couches entièrement connectées et de couches d'activation. Les tests seront réalisés sur 4 différentes tailles d'espace latent (8, 16, 32, 64) et 512 fréquences. Le fonctionnement de ce réseau est illustré dans la figure 4.3. Il n'y a pas de fonction d'activation après la couche dense donnant l'espace latent et la couche dense donnant les coefficients spectraux en sortie afin que les valeurs calculées ne soient pas dans un intervalle tronqué. Durant l'entraînement, il n'y a pas besoin de calculer les deux transformées spectrales puisque les coefficients spectraux sont directement stockés et que la fonction de coût du réseau est calculée entre les coefficients spectraux en entrée et en sortie.

Dans la figure 4.4, différentes valeurs calculées sont présentées. À gauche, la valeur de la fonction de coût calculée entre les coefficients spectraux en entrée et

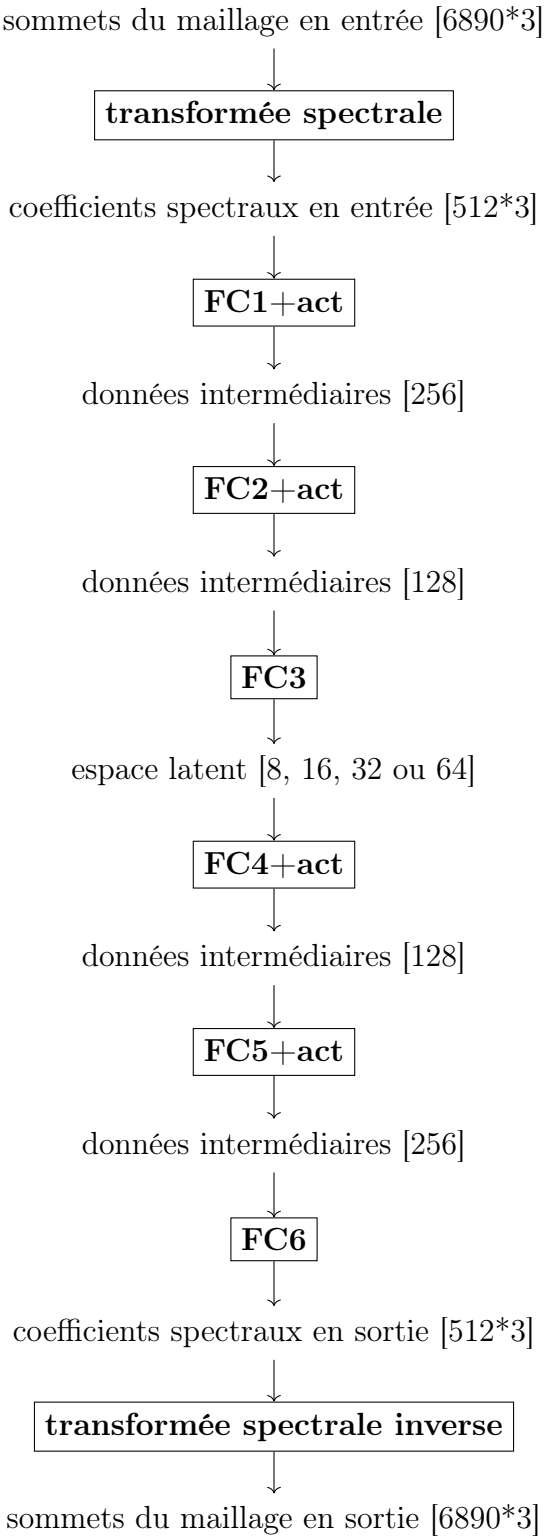


FIGURE 4.3 – Processus du réseau entièrement connecté. Les dimensions sont entre crochets et les opérations sont en gras, **FC** étant une transformation effectuée par une couche dense et **act** une couche d'activation. L'entraînement est réalisé sur les coefficients spectraux directement, sans transformées spectrales directe et inverse.

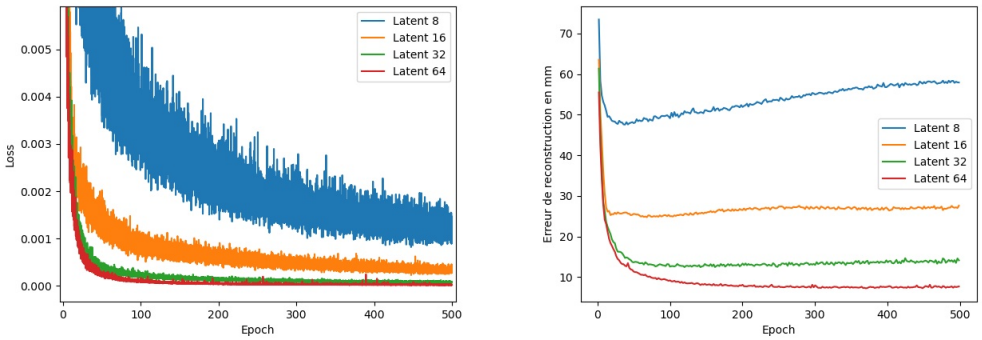


FIGURE 4.4 – Différentes valeurs calculées lors de l'entraînement du réseau entièrement connecté en fonction de la taille de l'espace latent. À gauche, la fonction de coût. À droite, l'erreur moyenne en millimètres sur l'ensemble de test entre les maillages en entrée et en sortie.

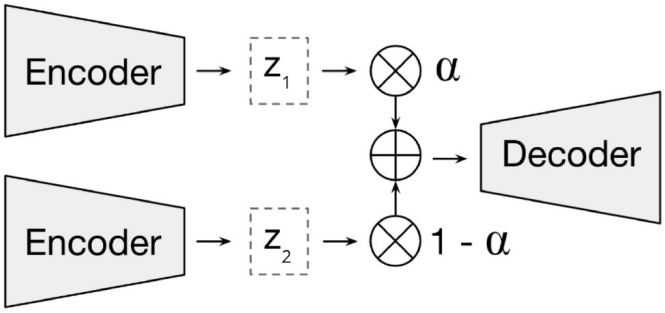


FIGURE 4.5 – Fonctionnement de l'interpolation dans l'espace latent.

en sortie est affichée en fonction du temps d'entraînement : pour tous les modèles, l'apprentissage fonctionne (la fonction de coût diminue) et le réseau apprend à reconstruire les échantillons de l'ensemble d'entraînement. Naturellement, les modèles ayant un espace latent de taille plus grande parviennent mieux à reconstruire les échantillons d'entraînement. À droite, le score de reconstruction sur la base de test est présenté. Le score de reconstruction est calculé comme la distance moyenne en millimètres entre chaque sommet dans le domaine spatial d'un maillage en entrée et en sortie. On voit ici que les modèles ayant un espace latent de taille plus grande parviennent mieux à généraliser sur les échantillons de l'ensemble de test. Un phénomène intéressant est aussi mis en valeur sur le graphique de droite : les courbes correspondant aux espaces latents de taille 8 et 16 "remontent" au bout d'un certain temps d'entraînement. Cela reflète un surapprentissage (*overfitting*) : le réseau apprend trop à reproduire les échantillons d'entraînement et n'arrive plus à généraliser. Plusieurs éléments peuvent causer cela, comme le trop grand nombre de paramètres du modèle ou le nombre insuffisant d'échantillons dans la base d'entraînement.

Ici, cela est probablement causé par la petite taille de l'espace latent. Néanmoins, cette taille réduite permet de créer un phénomène intéressant. La qualité

FIGURE 4.6 – Interpolations dans des espaces latents de plusieurs dimensions. Pour chaque ligne, l'interpolation est réalisée entre les deux maillages des colonnes "Mesh 1" et "Mesh 2". Les colonnes "Latent 8, 16, 32, 64" correspondent à l'interpolation illustrée dans la figure 4.5, et la colonne "Domaine spatial" correspond à une interpolation dans le domaine spatial directement. Chaque interpolation est calculée en utilisant $\alpha = 0.5$.

d'un autoencodeur peut en effet être mesurée par sa capacité de compression et de généralisation en mesurant l'erreur entre des données en entrée et en sortie. Mais une autre fonctionnalité d'un autoencodeur est sa capacité à détecter les caractéristiques importantes d'une base de données, refletée par la manière dont est construit l'espace latent, et observable en interpolant deux échantillons dans ce dernier (figure 4.5).

Dans la figure 4.6, plusieurs interpolations sont illustrées en utilisant les espaces latents des modèles présentés dans la figure 4.4. Pour chaque ligne, l'interpolation est réalisée entre les deux maillages aux extrémités. Les colonnes "Latent 8, 16, 32, 64" correspondent à une interpolation dans l'espace latent et la colonne "Domaine spatial" correspond à une interpolation dans le domaine spatial directement.

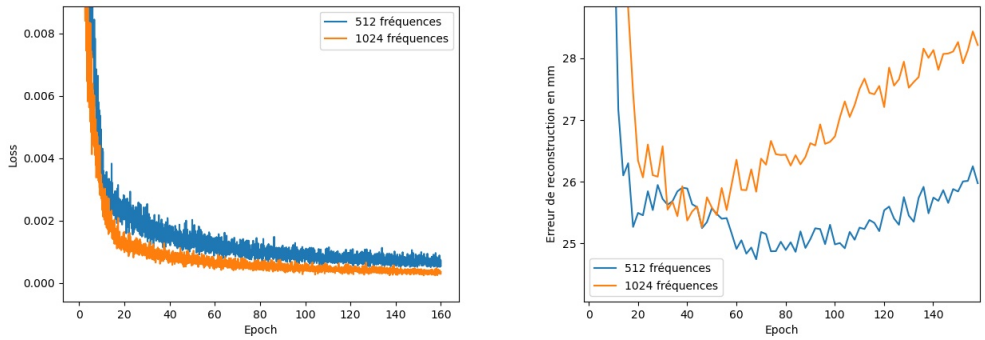


FIGURE 4.7 – Différentes valeurs calculées lors de l'entraînement du réseau entièrement connecté en fonction du nombre de fréquences utilisées. À gauche, la fonction de coût. À droite, l'erreur moyenne en millimètres sur l'ensemble de test entre les maillages en entrée et en sortie.

Chaque interpolation est calculée en utilisant $\alpha = 0.5$. On voit que les modèles ayant un espace latent de taille faible (8 et 16) sont capables de créer des interpolations réalistes qui conservent les longueurs de bras des maillages. En revanche, les modèles ayant un espace latent plus grand (32 et 64) produisent des interpolations qui se rapprochent plus de ce que l'on obtient directement dans le domaine spectral ou spatial (pour rappel, une interpolation dans le domaine spatial ou spectral est identique, voir section 3.2.1). Un modèle qui crée un espace latent dont la structure est identique à l'espace en entrée n'est utile que si l'on cherche simplement à compresser l'information. Un modèle créant un espace latent représentant correctement des échantillons réalistes de l'espace en entrée est lui capable de capturer les informations importantes de ce dernier, au détriment d'une bonne capacité de reconstruction. Maintenant, un objectif intéressant est d'essayer de fournir plus de fréquences au modèle afin qu'il ait accès aux détails contenus dans les fréquences plus hautes tout en ayant une capacité de génération intéressante.

La figure 4.7 présente l'évolution de la fonction de coût et de la reconstruction pour deux modèles utilisant des couches entièrement connectées et différents nombres de coefficients spectraux : 512 fréquences (858K paramètres) pour l'un et 1 024 fréquences (1.6M paramètres) pour l'autre. Bien que la fonction de coût soit plus basse pour le modèle utilisant 1 024 fréquences, l'erreur moyenne sur l'ensemble de test reste plus grande. Cela signifie que le modèle utilisant 1 024 fréquences surapprend plus que le modèle utilisant 512 fréquences, et qu'un modèle utilisant des couches entièrement connectées n'est pas viable si l'on cherche d'une part à augmenter le nombre de fréquences données au réseau, et d'autre part à diminuer ce surapprentissage. Il faut donc utiliser une autre architecture qui va permettre d'obtenir le même genre de résultat mais en utilisant moins de paramètres.

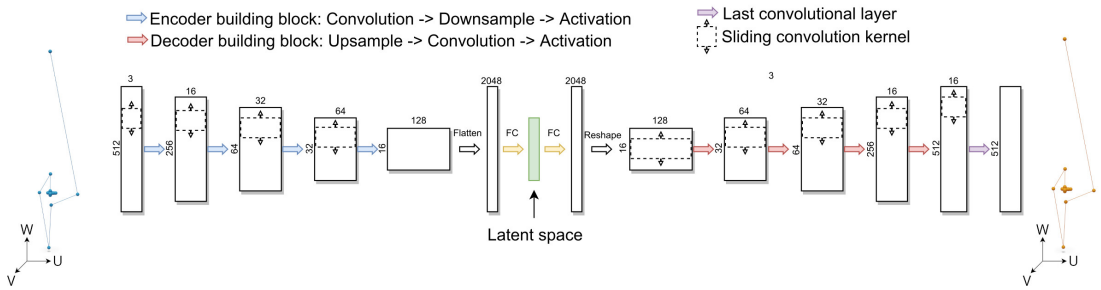


FIGURE 4.8 – Illustration d'un autoencodeur convolutif utilisant 512 fréquences.

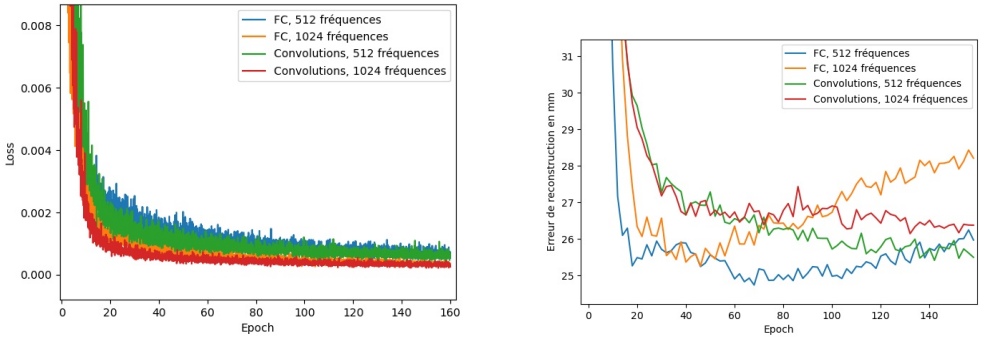


FIGURE 4.9 – Différentes valeurs calculées lors de l'entraînement de réseaux entièrement connectés et convolutifs en fonction du nombre de fréquences fournies et de l'architecture utilisée. À gauche, la fonction de coût. À droite, l'erreur moyenne en millimètres sur l'ensemble de test entre les maillages en entrée et en sortie.

4.4.2 Autoencodeur convolutif

Un autoencodeur convolutif (*Convolutional Neural Network* ou CNN), est représenté par la même architecture que précédemment mais en y ajoutant des couches de convolutions. Une telle architecture utilisant 512 fréquences est illustrée dans la figure 4.8. L'autoencodeur convolutif ajoute au modèle standard de l'autoencodeur des couches de convolution qui sont particulièrement efficaces pour extraire des caractéristiques de manière locale. Grâce aux opérations de sous-échantillonnage, ces convolutions sont aussi capables de détecter des caractéristiques de manière globale. Le nombre de paramètres est aussi réduit avec cette architecture en comparaison à l'utilisation de couches entièrement connectées (603K paramètres pour 512 fréquences et 1.1M pour 1 024 fréquences).

La figure 4.9 compare les fonctions de coût et les scores sur l'ensemble de test entre des modèles utilisant 512 ou 1 024 fréquences et entre des architectures utilisant des couches entièrement connectées ou des convolutions. On voit que les modèles utilisant des convolutions ne présentent pas de surapprentissage et que le score de

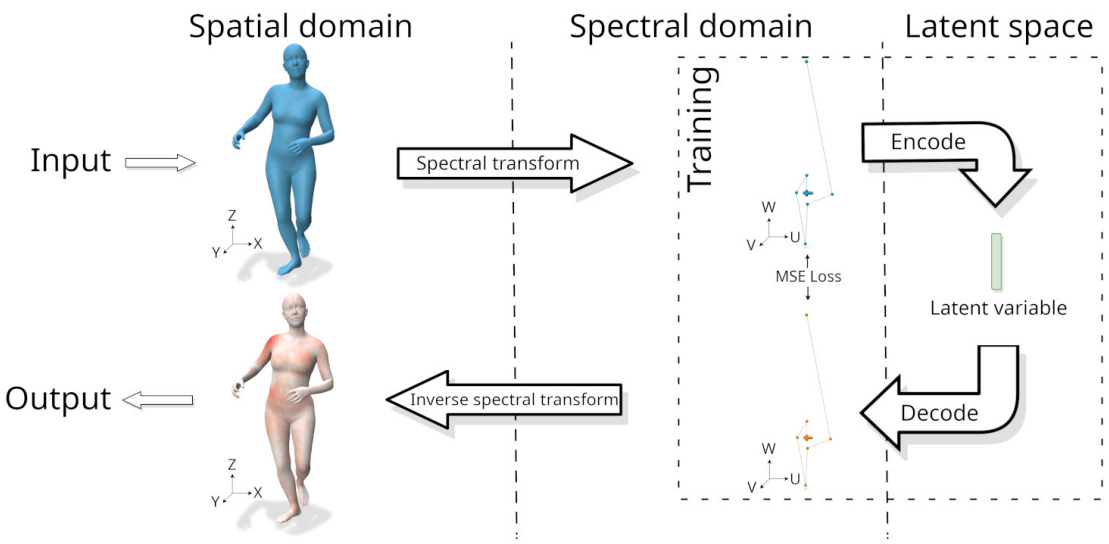


FIGURE 4.10 – Illustration du processus général proposé. Un maillage, situé dans le domaine spatial (coordonnées x, y, z) est transformé en coefficients spectraux (coordonnées u, v, w). Ensuite, un autoencodeur est utilisé afin de reconstruire ces coefficients spectraux en les faisant passer à travers un espace latent. La phase d'entraînement est entièrement réalisée dans le domaine spectral. Enfin, un maillage peut être récupéré à partir des coefficients spectraux en sortie en utilisant une transformée spectrale inverse.

reconstruction décroît, contrairement à ceux des architectures utilisant des couches entièrement connectées. Par ailleurs, le modèle utilisant 1 024 fréquences et des convolutions est capable de mieux généraliser que le modèle utilisant 1 024 fréquences et des couches entièrement connectées. Cela montre qu'utiliser des convolutions permet de considérer plus de fréquences sans observer de surapprentissage, comme c'est le cas avec des couches entièrement connectées. C'est pourquoi nous allons maintenant nous concentrer sur l'utilisation de convolutions. Dans la partie suivante, la première contribution de la thèse est présentée.

4.5 SAE

Jusqu'ici, nous avons montré comment le traitement spectral de maillages triangulaires pouvait être utilisé afin de transformer la géométrie d'une surface en coefficients spectraux, et comment ces derniers peuvent être donnés en entrée à un réseau de neurones de type autoencodeur. Une illustration du processus général est présentée dans la figure 4.10. Ce processus, nommé autoencodeur spectral (*Spectral Autoencoder* ou SAE), représente la première contribution de la thèse et est présenté plus en détail dans cette section.

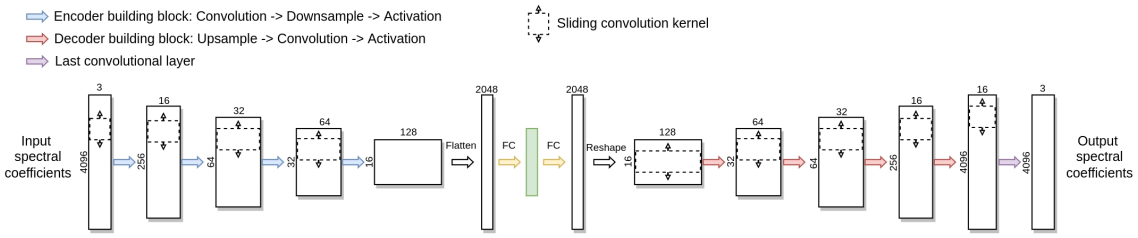


FIGURE 4.11 – Illustration de l'autoencodeur spectral avec agrégation apprise utilisant 4 096 fréquences (SAE-LP-4096).

Contributions principales

Le processus présenté permet d'utiliser des architectures traditionnelles sur des surfaces en utilisant une représentation spectrale des maillages. Les coefficients spectraux, calculés en transformant la géométrie depuis le Laplacien topologique (*Graph Laplacian*), sont ordonnés en fonction de la fréquence qui leur est associée. Ensuite, en utilisant une architecture de type autoencodeur, nous pouvons directement appliquer des convolutions et des opérations de sous et sur-échantillonnage à ces coefficients. Notre méthode appartient à la classe des modèles génératifs basés sur un autoencodeur destiné à traiter des formes 3D. Les principaux apports du réseau proposé sont :

- l'application de techniques d'apprentissage profond aux coefficients spectraux de maillages triangulaires sans revenir dans le domaine spatial,
- une architecture qui peut traiter les maillages de manière allégée en utilisant un nombre de fréquences inférieur au nombre de sommets,
- une architecture qui donne de meilleurs résultats de manière beaucoup plus rapide que les méthodes de l'état de l'art afin de pouvoir traiter de grands ensembles de données.

Le code et un modèle pré-entraîné sont disponibles à l'adresse suivante : https://github.com/MEPP-team/SAE.

Le modèle utilisé crée une représentation compacte de formes humaines déformables qui partagent une structure topologique commune. La figure 4.11 illustre ce modèle. La principale application recherchée utilisant cette représentation est la génération, mais d'autres sont possibles comme la classification, la segmentation, la correspondance ou la recherche de formes similaires.

Afin de pouvoir comparer notre processus à ceux de l'état de l'art, nous donnons ici plus de détails sur le prétraitement des données et sur l'application des convolutions et des opérations de sous et sur-échantillonnage aux coefficients spectraux.

4.5.1 Prétraitement

La première étape du prétraitement est le calcul des vecteurs propres. Ensuite, les sommets de tous les échantillons du jeu de données sont transformés en coefficients

spectraux et stockés afin d'éviter le calcul de la transformation durant l'entraînement. Dans la suite du processus, l'idée est de ne pas utiliser toutes les fréquences disponibles, mais plutôt une version tronquée de celles-ci. Dans la section 4.6, nous montrons l'impact de l'utilisation d'un nombre différent de fréquences. Pour rappel, la figure 3.9 exprime la compaction d'énergie du domaine spectral, et la section 4.3 donne des détails sur la différence de représentation entre des données spatiales et spectrales. Nous pouvons voir dans le tableau 4.1 les temps de prétraitement en fonction du nombre de vecteurs propres utilisés, raisonnables en comparaison avec le temps de calcul de spirales utilisées dans les méthodes de l'état de l'art [16, 53].

4.5.2 Convolutions dans l'état de l'art

Les convolutions sont les principaux éléments constitutifs des applications d'apprentissage profond. En 2D, elles permettent d'extraire des caractéristiques utiles des images de manière efficace car elles sont rapides à calculer et elles réduisent le nombre de paramètres d'un réseau de neurones. Il est donc naturel de chercher à étendre l'application des convolutions à d'autres domaines que les images. Les principales contributions concernant l'application de convolutions sur des maillages triangulaires se situent soit dans le domaine spectral, soit dans le domaine spatial. Ranjan et al. [114] ont construit un autoencodeur convolutif (CoMA) en utilisant le réseau ChebNet avec des filtres convolutifs spectraux, donnant des noyaux isotropes avec une expressivité limitée. Bouritsas et al. [16] ont amélioré ces résultats avec un opérateur de convolution en spirale (Neural3DMM [16]) qui définit un ordre explicite des voisins et crée des filtres anisotropes. Cependant, cette méthode nécessite de définir des sommets de départ pour les ordres en spirale, empêchant d'exploiter efficacement la structure irrégulière de la connectivité. De plus, des baisses de performance sont constatées puisque du remplissage par zéros est nécessaire pour avoir des spirales de taille fixe. Gong et al. [53] ont ensuite introduit SpiralNet++, une version améliorée de Neural3DMM, rendant les convolutions plus rapides en évitant les remplissages.

SpiralNet++, qui est le moyen le plus efficace et le plus rapide de faire des convolutions sur des maillages, doit tout de même définir un ordre sur les sommets, ce qui rend l'implémentation complexe. Dans notre travail, nous simplifions ce processus en faisant des convolutions sur la matrice des coefficients spectraux de manière naturelle puisque les fréquences sont déjà ordonnées. La figure 4.11 montre une illustration de l'autoencodeur spectral présenté avec agrégation apprise utilisant 4 096 fréquences (SAE-LP-4096, plus de détails sur cette architecture sont donnés ensuite). En faisant simplement glisser un noyau de convolution sur les coefficients, nous montrons que le réseau est capable d'apprendre des caractéristiques intéressantes.

4.5.3 Sous/sur-échantillonnage

Le comportement d'un réseau de neurones est étroitement lié à la procédure de sous/sur-échantillonnage. Les travaux classiques pour les signaux 1D ou les images

2D utilisent une fenêtre glissante afin de ne conserver que les valeurs maximales locales pour le sous-échantillonnage. Pour le sur-échantillonnage, des valeurs sont ajoutées afin de ramener une résolution plus élevée pour la couche suivante. Pour les maillages, Ranjan et al. [114] ont introduit une méthode de sous/sur-échantillonnage dans le domaine spatial. Les maillages sous-échantillonnés sont calculés en supprimant des arêtes tout en maintenant une erreur minimale entre les surfaces, et les maillages sur-échantillonnés sont calculés en créant des sommets à partir des coordonnées barycentriques des triangles des maillages sous-échantillonnés. Ces opérations sont représentées sous forme de matrices de transformation. Certains travaux ont proposé d'apprendre ces poids de sous/sur-échantillonnage avec une cartographie dense [150, 12] ou des couches entièrement connectées [43]. Chen et al. [30] ont introduit une méthode où ils sont appris via un module d'attention afin d'éviter une sur-paramétrisation.

Concernant notre processus, nous avons choisi d'échantillonner les coefficients spectraux avec deux méthodes. La première consiste à appliquer les opérations classiques de sous/sur-échantillonnage. Nous appellerons cette méthode **Spectral Autoencoder - Classic Pooling** (**SAE-CP**). La seconde consiste à apprendre les matrices de réduction et d'augmentation, comme dans les travaux de Chen et al. [30]. Mais nous n'avons pas utilisé de module d'attention car l'apprentissage direct des paramètres est plus simple, et comme nous n'utilisons pas toutes les fréquences disponibles, cela ne conduit pas à un sur-paramétrage. Nous appellerons cette méthode **Spectral Autoencoder - Learned Pooling** (**SAE-LP**). Dans ce cas, les matrices de sous/sur-échantillonnage sont simplement remplies de paramètres apprenables. Ensuite, lors de l'apprentissage, ces paramètres sont appris avec les autres paramètres du modèle. La figure 4.12 illustre les deux méthodes de sous/sur-échantillonnage.

Nous montrons dans la section suivante la comparaison entre l'utilisation du sous/sur-échantillonnage classique ou appris, ainsi qu'une comparaison entre plusieurs choix de nombre de fréquences utilisées.

4.6 Évaluations

Dans cette section, nous évaluons les modèles présentés. Les ensembles d'entraînement utilisés sont ceux présentés dans la section 4.3. Notre meilleur modèle est évalué par rapport à deux modèles de référence : Neural3DMM [16] et SpiralNet++ [53]. Nous présentons des résultats quantitatifs et qualitatifs et nous comparons les vitesses de calcul. Ensuite, une étude d'ablation montre qu'utiliser moins de fréquences donne toujours de bons résultats tout en réduisant le nombre de paramètres des modèles, puis nous montrons qu'il est possible d'interpoler dans l'espace latent afin de générer des maillages réalistes.

La métrique utilisée pour les expériences est une mesure de la qualité des maillages reconstruits à partir de l'espace latent. Elle est calculée comme la distance moyenne

Méthode	Vecteurs propres	DFaust	Total DFaust	AMASS	Total AMASS
Neural3DMM	-	-	$\sim$ 30s	-	$\sim$ 30s
SpiralNet++	-	-	$\sim$ 30s	-	$\sim$ 30s
SAE-*-6890	$\sim$40s	$\sim$3s	$\sim$ 43s	$\sim$16s	$\sim$ 56s
SAE-*-4096	$\sim$28s	$\sim$2s	$\sim$ 30s	$\sim$12s	$\sim$ 40s
SAE-*-2048	$\sim$36s	$\sim$1s	$\sim$ 37s	$\sim$8s	$\sim$ 44s
SAE-*-1024	$\sim$9s	$\sim$0.5s	$\sim$ 9.5s	$\sim$7s	$\sim$ 16s
SAE-*-512	$\sim$3.5s	$\sim$0.3s	$\sim$ 3.8s	$\sim$3.2s	$\sim$ 6.7s

TABLE 4.1 – Comparaison des temps de prétraitement. Pour Neural3DMM et SpiralNet++, les spirales ne sont calculées qu'une seule fois sur un maillage de base, les temps ne dépendent donc pas de la taille de la base de données. SAE-*-k représente notre autoencodeur spectral avec agrégation classique ou apprise utilisant k fréquences. Pour notre méthode, ce temps dépend du calcul des vecteurs propres (colonne vecteurs propres) et de la transformation de tous les maillages de la base de données en coefficients spectraux (colonnes DFaust et AMASS). Même avec 4 096 fréquences, le temps de prétraitement est raisonnable comparé à la méthode utilisant des spirales. Le fait que le calcul des vecteurs propres pour 2 048 fréquences soit plus long que pour 4 096 vient du solveur utilisé.

en millimètres entre les sommets correspondants des maillages en entrée et en sortie. Cela mesure la capacité du modèle à obtenir une représentation compacte et à généraliser à de nouvelles surfaces à partir de la distribution sur laquelle il a été entraîné. Les maillages ne sont pas normalisés et ont la taille réelle de la personne.

4.6.1 Implémentation

Nous suivons le processus des travaux précédents pour les architectures de l'autoencodeur.

Neural3DMM et **SpiralNet++** : les filtres convolutifs de l'encodeur ont les tailles [3, 16, 32, 64, 128]. Une couche entièrement connectée transforme ensuite les données à la taille latente souhaitée. Après une autre couche entièrement connectée, les filtres convolutifs du décodeur ont les tailles [128, 64, 32, 16, 16]. Une dernière couche convolutive transforme les données et adapte leur nombre de dimensions à celui de la géométrie, soit 3. Des convolutions dilatées avec h = 2 sauts et un rapport de dilatation r = 2 sont utilisées pour la première et les deux dernières couches de l'encodeur et du décodeur respectivement. Les tailles des spirales sont [12, 14, 9, 9] pour l'encodeur et [9, 9, 14, 12, 12] pour le décodeur.

Rappelons que pour les deux modèles de référence, les entrées sont les coordonnées cartésiennes standardisées des maillages dans le domaine spatial pour obtenir une moyenne égale à zéro et un écart type égal à un. Pour nos modèles, les entrées sont les coefficients spectraux.

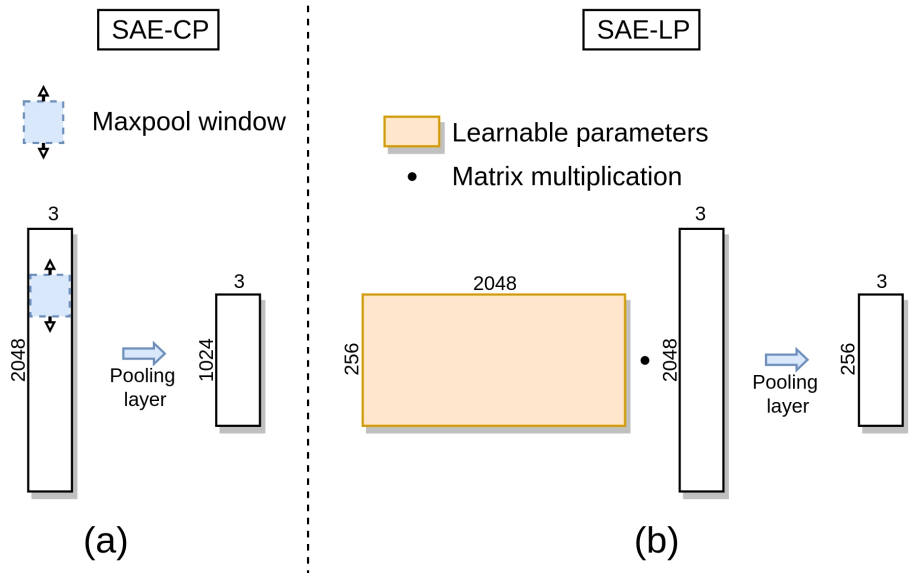


FIGURE 4.12 – Illustration des deux méthodes d'agrégation utilisées lors du sous-échantillonnage de 2 048 fréquences. (a) - Une fenêtre de sous-échantillonnage classique de taille 2 avec un pas de 2 permet de réduire la dimension des coeffcients spectraux d'un facteur 2. Cette méthode est nommée *Spectral Autoencoder with classic pooling* (SAE-CP). (b) - les coefficients spectraux sous-échantillonnés sont calculés en les multipliant par une matrice contenant des paramètres apprenables. Le facteur de sous-échantillonnage est plus important qu'avec l'agrégation classique (voir section 4.6.1 pour plus d'informations). Cette méthode est appelée *Spectral Autoencoder with Learned Pooling* (SAE-LP).

SAE-CP-k : la construction du réseau ***S**pectral **A**utoe**n**coder with **c**lassic **p**ooling* utilisant k fréquences suit les architectures courantes d'autoencodeur utilisant des convolutions et des opérations de sous/sur-échantillonnage. Le nombre k de fréquences des coefficients spectraux d'entrée est une puissance de deux. Le nombre de couches de l'encodeur dépend du nombre de fois qu'il faut diviser par deux pour avoir 32 fréquences restantes après la dernière étape de sous-échantillonnage, soit 4 étapes pour 512 fréquences, 5 pour 1 024 et ainsi de suite. Pour 512 fréquences, les filtres convolutifs de l'encodeur ont les tailles [3, 32, 64, 64, 128]. Le décodeur possède des filtres de tailles [128, 64, 64, 32, 32, 3]. Si plus de couches sont nécessaires car plus de fréquences sont utilisées, nous dupliquons les filtres de taille 64. Les tailles des noyaux de convolutions sont de 3 pour toutes les couches avec un remplissage de 1 afin que la longueur de l'entrée ne soit pas modifiée après l'application de la convolution. De plus, la taille de la fenêtre pour le sous-échantillonnage et le facteur du sur-échantillonnage est de 2 pour toutes les couches.

SAE-LP-k : la construction du réseau ***S**pectral **A**utoe**n**coder with **l**earned **p**ooling* utilisant k fréquences est similaire à la précédente à l'exception des couches de sous/sur-échantillonnage. Les filtres convolutifs de l'encodeur ont des tailles [3, 16, 32, 64, 128]. Au lieu de faire un sous/sur-échantillonnage classique, des matrices

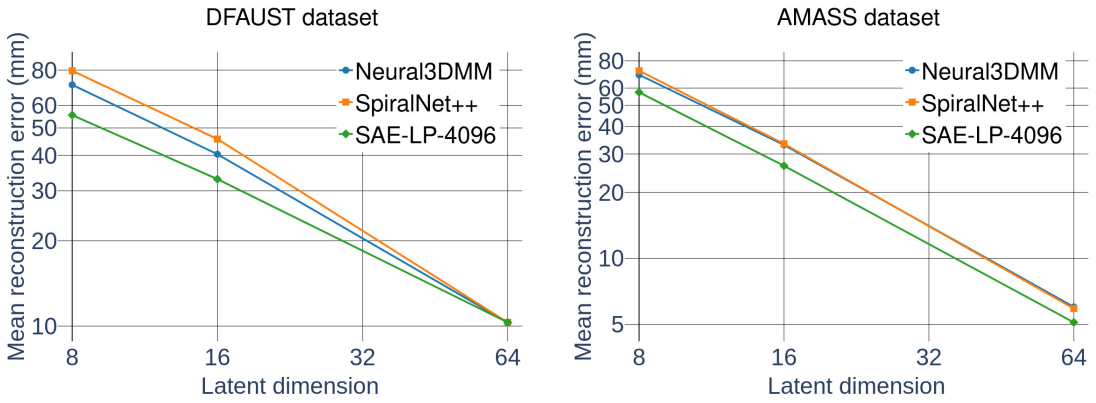


FIGURE 4.13 – Comparaison de la reconstruction sur les jeux de données DFaust et AMASS entre Neural3DMM, SpiralNet++ et notre meilleur modèle. Pour toutes les tailles latentes, notre méthode surpasse les deux modèles de référence.

de transformation contenant des paramètres apprenables sont créées. Ensuite, ces paramètres sont appris avec les autres composants du modèle. Les tailles des matrices pour l'encodeur sont $[k, 256, 64, 32, 16]$, k étant le nombre de fréquences choisi. Pour le décodeur, les filtres convolutifs ont les tailles [128, 64, 32, 16, 16, 3]. Les matrices de sur-échantillonnage sont de la même taille que celles de l'encodeur dans l'ordre inverse. Les tailles des noyaux de convolutions sont également de 3 pour ce modèle avec un remplissage de 1.

Tous les modèles sont entraînés sur le même matériel. Les données sont fournies par paquets (*batch*) de 16, le taux d'apprentissage est de 10^{-4} et un ordonnanceur est utilisé pour que le taux d'apprentissage soit réduit d'un facteur de 0.1 lorsque la reconstruction a cessé de s'améliorer (avec un seuil de 10^{-4}) pendant 3 époques. Les modèles sont entraînés pendant un maximum de 20 heures.

4.6.2 Comparaison avec les modèles de référence

Nous évaluons d'abord notre modèle donnant les meilleurs résultats : le SAE-LP-4096. Les différences entre les reconstructions moyennes sur les bases de données de test, la qualité visuelle des maillages reconstruits et les temps par époque sont comparés aux deux modèles de référence : Neural3DMM [16] et SpiralNet++ [53].

Résultats quantitatifs de la reconstruction

Nous suivons [16] pour le choix des tailles latentes, sur la base de la variance expliquée par PCA d'environ 85%, 95% et 99% de la variance totale. La figure 4.13 montre les résultats de la moyenne de reconstruction sur les bases de données DFaust et AMASS. Ici, le SAE-LP-4096 est un modèle qui prend en entrée 4 096 fréquences. Nous pouvons voir que pour toutes les tailles latentes, notre modèle surpasse les deux modèles de base. Le tableau 4.2 montre le nombre de paramètres des trois

Méthode	Latent size 8	16	64
Neural3DMM	274K	331K	675K
SpiralNet++	415K	471K	802K
SAE-LP-4096	2.23M	2.26M	2.46M

TABLE 4.2 – Comparaison du nombre de paramètres en fonction de la taille de l'espace latent. Dans ce cas, notre modèle utilise 4 096 fréquences. Plus de 90% des paramètres du SAE-LP-4096 sont situés dans la première et la dernière matrice de sous/sur-échantillonnage.

réseaux de neurones. Le SAE-LP-4096 en a plus, mais 90% de ces paramètres sont concentrés dans la première matrice de sous-échantillonnage et la dernière matrice de sur-échantillonnage (voir section 4.6.1). De plus, nous montrons dans la section 4.6.3 que des modèles ayant la même architecture mais sans autant de paramètres parviennent toujours à obtenir des résultats compétitifs.

Nous pouvons comparer la capacité de compression de l'espace latent du modèle avec la capacité de compression du domaine spectral. Cela peut être fait en mesurant simplement l'erreur de reconstruction après l'application d'une transformée spectrale directe puis d'une transformée spectrale inverse sur tous les maillages de l'ensemble de données de test lors de l'utilisation d'un nombre de fréquences similaire au nombre de dimensions latentes. Sur l'ensemble de données DFaust, les erreurs de reconstruction moyennes lors de l'utilisation de 3, 6 et 22 fréquences (résultant en 9-18-66 dimensions respectivement puisqu'il y a 3 coordonnées u, v, w) sont respectivement de 368,1±42,7, 96,5±10,7 et 54,7±3,6 millimètres. La figure 4.13 présente des erreurs de reconstruction pour les dimensions latentes de 8-16-64 de 55,5±16,9, 33,0±10,7, 10,3±2,7 millimètres respectivement. Ceci montre clairement que les espaces latents construits par le modèle ont une meilleure capacité de compression.

Résultats qualitatifs de la reconstruction

Des reconstructions visuelles sont présentées dans la figure 4.14. Tous les modèles comparés ont une dimension latente de 64. Le constat principal est que lorsque les modèles de référence doivent manipuler des parties du corps dans une position peu souvent vue lors de l'entraînement (notamment les bras et les mains), les détails sont plus dégénérés. En revanche, l'autoencodeur spectral est capable de reconstruire des surfaces plus lisses, ce qui donne de meilleurs résultats visuels. Cela peut s'expliquer par le fait que, lors de la première phase d'apprentissage, notre modèle apprend d'abord à mieux reconstruire l'information contenue dans les basses fréquences puisque les coefficients leur correspondant ont des amplitudes plus élevées que ceux de hautes fréquences. Cela conduit à des parties du corps dans la bonne position mais sans suffisamment de détails. Puis, en fin d'entraînement, le modèle apprend à reconstituer les détails. Au contraire, les modèles de référence ont du mal à reconstruire ces parties du corps.

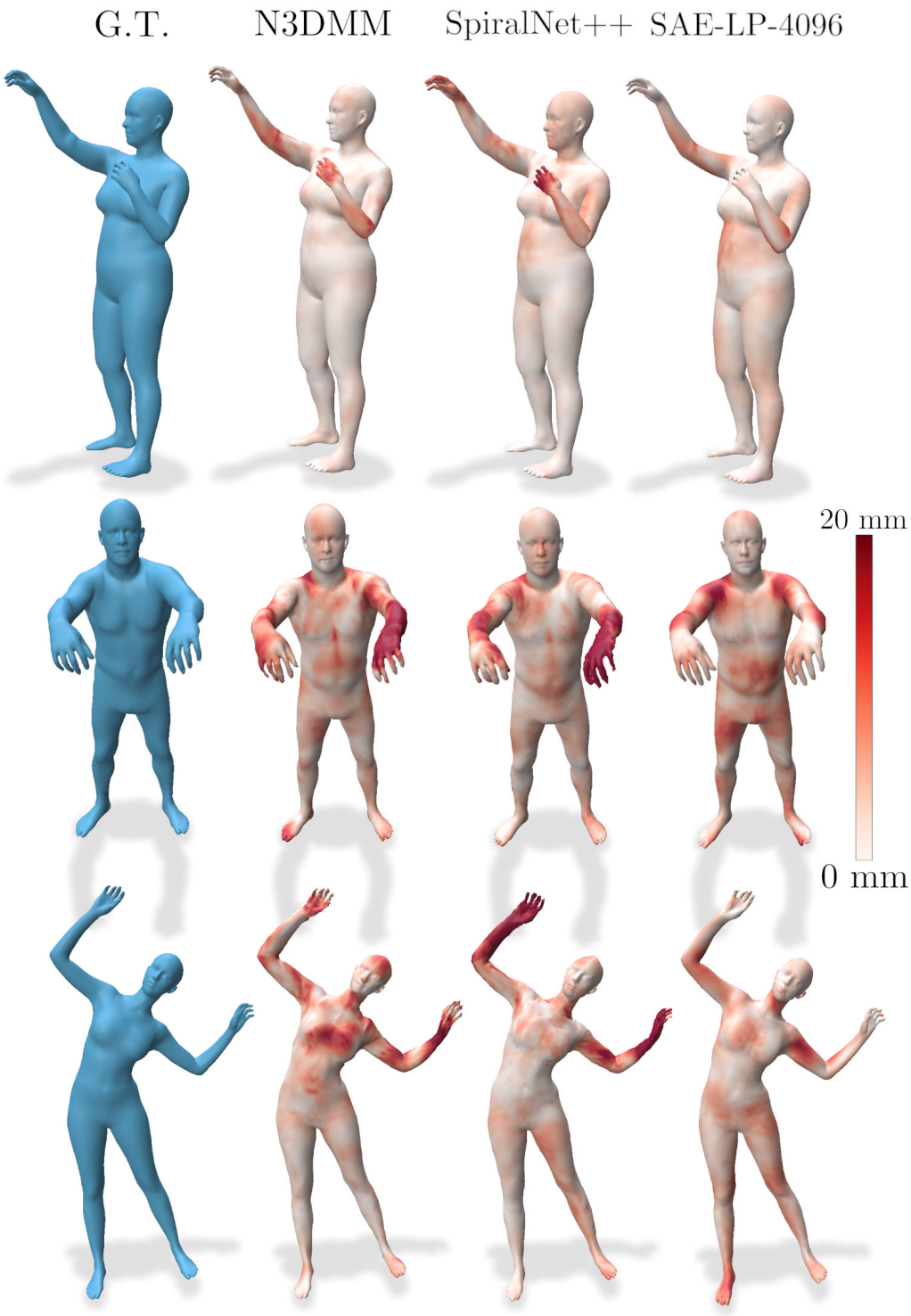


FIGURE 4.14 – Comparaison de reconstructions entre Neural3DMM (à gauche), SpiralNet++ (au milieu) et notre modèle utilisant 4 096 fréquences, le SAE-LP-4096 (à droite). La dimension latente est de 64. Lors de la reconstruction de parties du corps dans une position qui n'est pas souvent vue dans l'ensemble de données, notre modèle produit des surfaces plus détaillées et plus lisses, en particulier sur les mains.

Méthode	DFaust	AMASS
Neural3DMM	$\sim$ 156s	$\sim$ 472s
SpiralNet++	$\sim$ 68s	$\sim$ 200s
SAE-LP-4096	$\sim$ **28s**	$\sim$ **83s**

TABLE 4.3 – Comparaison du temps par époque sur les ensembles de données DFaust et AMASS entre le SAE-LP-4096 et les lignes de base. Notre modèle est plus rapide d'un facteur d'environ 2.4 par rapport à SpiralNet++ et d'environ 5.5 par rapport à Neural3DMM.

Temps par époque

Le principal avantage de notre méthode est la rapidité de calcul. Nous pouvons voir dans le tableau 4.3 la différence de temps par époque pour Neural3DMM, SpiralNet++ et le SAE-LP-4096. Comme nous ne prenons pas en entrée toutes les fréquences disponibles, le processus est beaucoup plus rapide, tout en ayant toujours accès aux fréquences importantes. De plus, Neural3DMM et SpiralNet++ doivent réorganiser les tableaux de sommets afin de faire des convolutions spécifiées par les spirales précalculées, contrairement à notre méthode où les convolutions sont effectuées sur des matrices de manière classique. Ensuite, même si SpiralNet++ a réussi à réduire le temps de calcul par époque par rapport à Neural3DMM, notre réseau reste plus rapide.

Nous avons montré que le SAE-LP-4096 est capable d'apprendre des caractéristiques importantes sur des maillages triangulaires et de construire un espace latent où la reconstruction est possible, donnant de meilleurs résultats que les méthodes de l'état de l'art. Nous montrons maintenant l'impact de l'utilisation de différentes configurations pour notre architecture.

4.6.3 Étude d'ablation

Dans cette section, nous évaluons le comportement de notre architecture en utilisant l'agrégation apprise, l'agrégation classique et un nombre différent de fréquences. Nous présentons d'abord des résultats quantitatifs afin de comparer l'erreur moyenne de reconstruction avec différentes configurations. Ensuite, des résultats sont présentés montrant la capacité d'un modèle entraîné sur une base de données à généraliser sur l'autre jeu de données. Enfin, les résultats qualitatifs montrent que les maillages reconstruits sont toujours visuellement acceptables même en utilisant moins de fréquences.

Résultats quantitatifs

Les tableaux 4.4 et 4.5 montrent le nombre de paramètres en fonction du nombre de fréquences utilisées et de la taille de l'espace latent. Les résultats de la reconstruction sur la base de données AMASS en utilisant différentes configurations sont présentés dans la figure 4.15. La ligne rouge indique le meilleur score atteignable

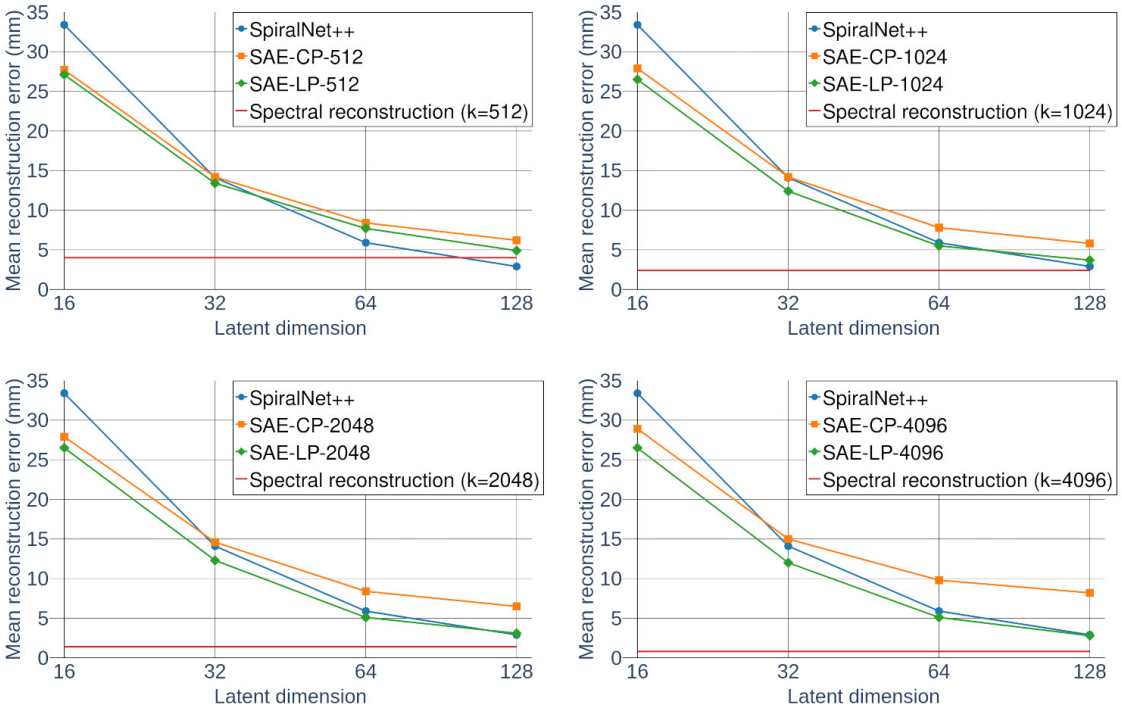


FIGURE 4.15 – Comparaison des résultats de reconstruction entre le modèle de référence SpiralNet++, le SAE-CP et le SAE-LP en utilisant un nombre différent de fréquences et de dimensions latentes sur l'ensemble de données AMASS. Le minimum atteignable, représenté par "Spectral reconstruction", est calculé pour chaque nombre de fréquences et correspond à l'erreur moyenne sur l'ensemble de données de test entre les maillages d'origine et les maillages reconstruits après une transformée spectrale inverse. Même en utilisant une méthode d'agrégation plus simple ou moins de fréquences, le SAE est capable de donner des résultats compétitifs.

Méthode	Latent size 8	16	32	64	128
SAE-LP-512	400K	433K	499KK	630K	892K
SAE-LP-1024	662K	695K	761K	892K	1.15M
SAE-LP-2048	1.18M	1.22M	1.28M	1.41M	1.67M
SAE-LP-4096	2.23M	2.26M	2.33M	2.46M	2.72M

TABLE 4.4 – Nombre de paramètres pour les autoencodeurs spectraux utilisant l'agrégation apprise. La plupart des paramètres sont contenus dans la première matrice de sous-échantillonnage et la dernière de sur-échantillonnage, surtout lorsque le nombre de fréquences utilisées est élevé.

Method	Latent size 8	16	32	64	128
SAE-CP-512	159K	225K	356K	618K	1.14M
SAE-CP-1024	184K	250K	381K	643K	1.16M
SAE-CP-2048	209K	274K	405K	668K	1.19M
SAE-CP-4096	233K	299K	430K	692K	1.21M

TABLE 4.5 – Nombre de paramètres pour les autoencodeurs spectraux utilisant l'agrégation classique.

pour chaque nombre de fréquences utilisées, correspondant à l'erreur moyenne sur l'ensemble de données de test après une transformation spectrale inverse (voir figure 3.9). En utilisant seulement 512 fréquences (figure 4.15 en haut à gauche), nos modèles ne peuvent pas avoir un meilleur score que celui du modèle comparé pour une dimension latente de 128. Pour une dimension latente de 16, nos deux architectures donnent de meilleurs résultats. Pour une dimension latente de 32, le SAE-CP donne des résultats similaires aux modèles comparés, tandis que le SAE-LP a toujours un meilleur score de reconstruction. Pour des dimensions latentes plus élevées, le SAE-CP produit des maillages moins bien reconstruits tandis que le SAE-LP donne des résultats similaires à ceux des modèles de référence lors de l'utilisation d'un faible nombre de fréquences et donne de meilleurs scores lors de l'utilisation d'un plus grand nombre de fréquences. Le pire comportement de SAE-CP provient de la méthode d'agrégation et est corrigée avec le SAE-LP. De plus, lorsqu'ils utilisent trop peu de fréquences, les réseaux n'ont pas accès aux informations détaillées, ce qui conduit à des maillages insuffisamment précis, comme illustré dans la suite.

Résultats qualitatifs

Différents niveaux de détails sur une tête et un pied de maillages reconstruits avec un nombre différent de fréquences utilisées sont présentés dans la figure 4.16. Nous pouvons voir qu'en donnant suffisamment de fréquences au modèle, il est capable de reconstruire des maillages avec autant de précision que Neural3DMM. La figure 4.17 montre des résultats visuels par rapport à SpiralNet++. Pour les maillages reconstruits avec 1 024 fréquences, nous pouvons voir qu'il manque des détails contenus dans des coefficients spectraux de fréquences élevées : cela se traduit par des parties symétriques sur le corps avec de faibles erreurs (voir figure 3.9 pour une comparaison). Néanmoins, le modèle n'utilisant que 1 024 fréquences parvient tout de même à reconstruire certaines parties du corps avec plus de précision par rapport à SpiralNet++. Ensuite, les modèles utilisant plus de fréquences donnent le même genre de résultats que celui n'ayant accès qu'à 1 024 fréquences mais avec plus de détails.

Temps par époque

Les temps par époque pour le SAE-CP sont présentés dans le tableau 4.6. Nous pouvons voir qu'en traitant plus de fréquences, le réseau devient plus lent mais reste

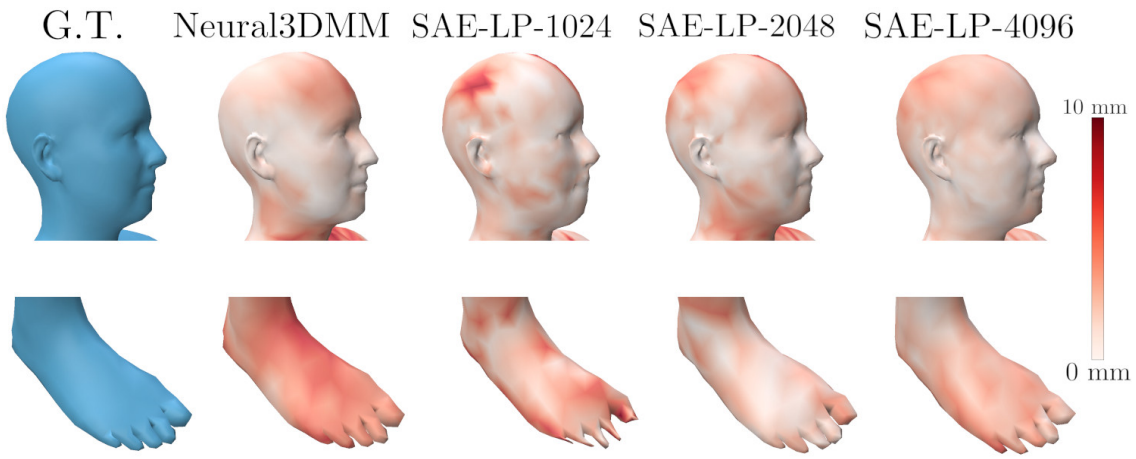


FIGURE 4.16 – Exemples de détails sur des maillages reconstruits avec des modèles utilisant différents nombres de fréquences (1 024, 2 048 et 4 096), zoomés sur la tête et le pied. Donner au modèle l'accès à des fréquences plus élevées conduit à des maillages reconstruits avec plus de détails.

Méthode	AMASS
SAE-CP-512	$\sim$ **84s**
SAE-CP-1024	$\sim$ 89s
SAE-CP-2048	$\sim$ 93s
SAE-CP-4096	$\sim$ 115s

TABLE 4.6 – Temps par époque des modèles avec agrégation classique. Pour le SAE-CP, l'augmentation du nombre de fréquences données conduit à des temps de calcul plus élevés car le facteur de sous-échantillonnage est de 2, conduisant à un nombre de couches plus grand. Voir figure 4.12.

.

plus rapide que SpiralNet++. En effet, le nombre de couches du réseau SAE-CP dépend du nombre de fréquences en entrée, ce qui conduit à davantage de convolutions appliquées sur plus de coefficients spectraux. Ensuite, le tableau 4.7 montre le temps de calcul par époque de SAE-LP. Prendre en entrée plus de coefficients spectraux ne détériore pas la vitesse et donne toujours des temps d'entraînement beaucoup plus courts que les modèles de référence. C'est probablement le principal avantage de notre méthode puisque l'entraînement sur des bases de données avec beaucoup plus d'échantillons, comme la base de données AMASS échantillonnée avec plus de poses, est maintenant réalisable en un temps réduit.

Bases de données croisées

Nous essayons ensuite d'observer la capacité d'un modèle entraîné sur un jeu de données à généraliser sur un autre. Cela permet d'évaluer la capacité de généralisation du réseau. C'est une tâche difficile à réaliser puisque pour pouvoir généraliser, la

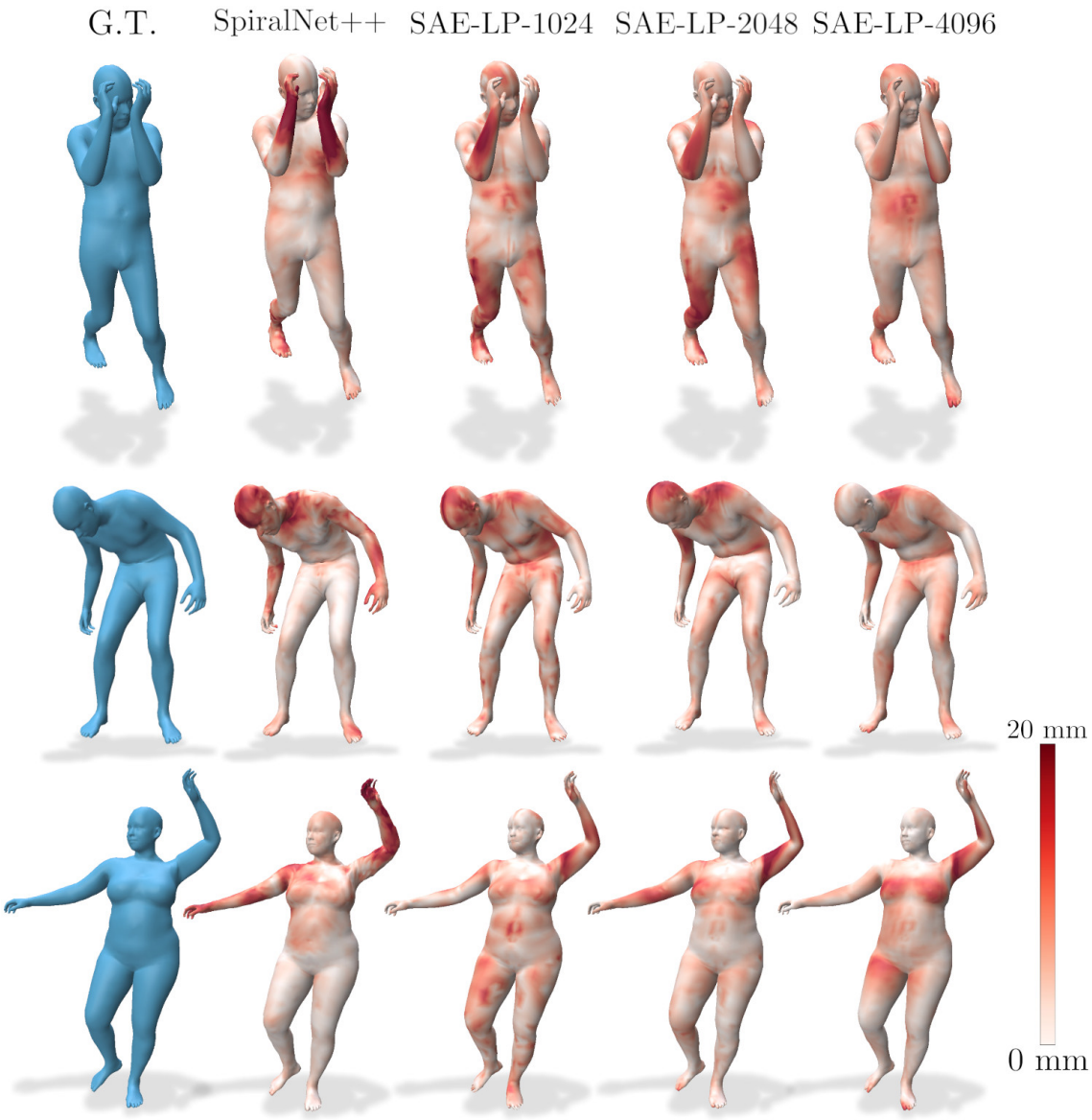


FIGURE 4.17 – Comparaison de reconstructions entre SpiralNet++ et l'autoencodeur spectral avec agrégation apprise utilisant un nombre de fréquences de 1 024, 2 048 et 4 096. La dimension latente est de 64 pour tous les modèles. Tout en montrant des erreurs correspondant au manque d'information contenue dans les hautes fréquences, le modèle utilisant 1 024 fréquences parvient tout de même à reconstruire certaines parties du corps d'une meilleure manière que SpiralNet++.

Méthode	AMASS
SAE-LP-512	$\sim$ **80s**
SAE-LP-1024	$\sim$ 83s
SAE-LP-2048	$\sim$ 83s
SAE-LP-4096	$\sim$ 83s

TABLE 4.7 – Temps par époque des modèles avec agrégation apprise en fonction des fréquences utilisées. Même avec plus de fréquences, le temps de calcul reste constant.

Latent size	Test dataset	Train dataset	Mean (mm)	Cross error (mm)
8	DFaust	DFaust	55.5	-
		AMASS	64.0	+**8.5**
	AMASS	AMASS	57.4	-
		DFaust	119.2	+61.8
16	DFaust	DFaust	33.0	-
		AMASS	35.9	+**2.9**
	AMASS	AMASS	26.5	-
		DFaust	73.6	+47.1
64	DFaust	DFaust	10.3	-
		AMASS	17.9	+**7.6**
	AMASS	AMASS	5.1	-
		DFaust	27.9	+22.8

TABLE 4.8 – Comparaison des erreurs de reconstruction lors du croisement de jeux de données à l'aide du modèle SAE-LP-4096. L'erreur croisée est la différence de reconstruction moyenne entre des modèles entraînés sur une base de données différente mais évalués sur la même. Un modèle entraîné sur AMASS, un grand jeu de données, est capable de reconstruire correctement la base de données DFaust qui est moins dense.

base de données doit être très dense, et le processus utilisé doit être assez rapide pour pouvoir entraîner le réseau sur cette dernière. Le tableau 4.8 présente les résultats lors du croisement des bases de données. Naturellement, les modèles entraînés sur le jeu de données DFaust ont du mal à reconstruire les maillages du jeu de données AMASS car ce dernier est plus volumineux. Inversement, les modèles entraînés sur le jeu de données AMASS parviennent à obtenir un score proche de celui obtenu par les modèles entraînés sur le jeu de données DFaust. Cela montre l'avantage pour un modèle d'être rapide, et donc capable d'apprendre sur un grand ensemble.

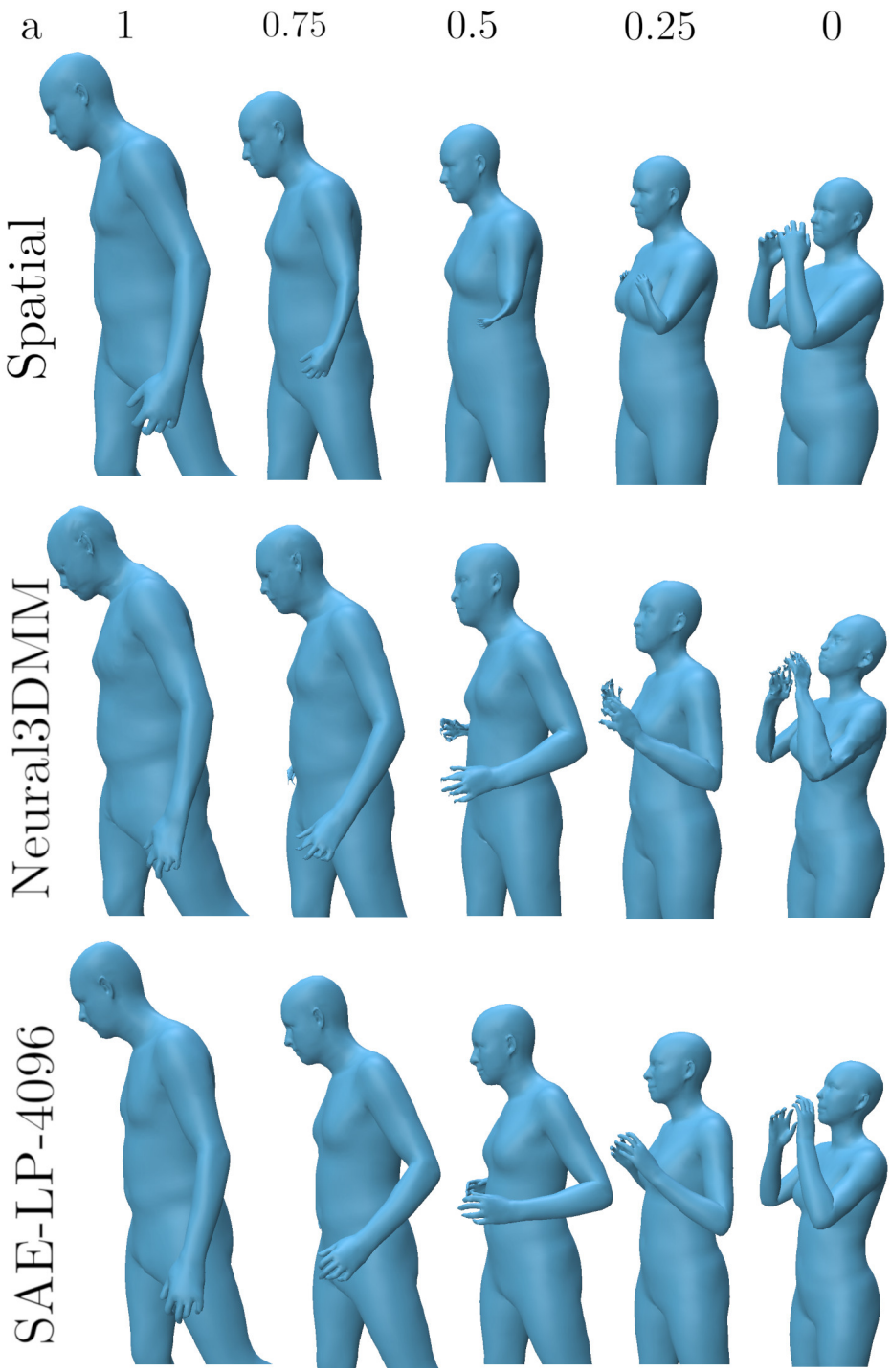


FIGURE 4.18 – La première ligne montre une interpolation linéaire des coordonnées cartésiennes des deux maillages à droite et à gauche. La deuxième et la troisième ligne montrent la génération de maillage en interpolant les codes latents de deux échantillons avec Neural3DMM et notre méthode. La dimension latente est 16 pour les deux modèles.

4.6.4 Interpolation

Enfin, nous montrons la capacité d'interpolation de notre modèle par rapport à Neural3DMM. En effectuant une interpolation linéaire dans l'espace latent, il est possible avec une architecture de type autoencodeur de générer de nouveaux échantillons, comme illustré dans la figure 4.5. Nous sélectionnons deux échantillons différents dans l'ensemble de test, les encodons dans leurs représentations latentes z_1 et z_2, et produisons de nouveaux maillages en échantillonnant le long de la ligne : $z = a * z_1 + (1 - a) * z_2$, $a \in [0, 1]$. La figure 4.18 compare les interpolations entre Neural3DMM et le SAE-LP-4096 pour diverses valeurs de a. Pour la première ligne, les maillages aux extrémités sont ceux sélectionnés dans la base de données de test, et les trois maillages du milieu correspondent à une interpolation linéaire des coordonnées cartésiennes. Lors de l'utilisation de toutes les fréquences et de l'interpolation des coefficients spectraux, le résultat est le même qu'une interpolation dans le domaine spatial puisque la transformée laplacienne est une fonction linéaire. Pour les deux autres lignes, les maillages aux extrémités sont les reconstructions des deux maillages sélectionnés et les codes latents interpolés et décodés sont présentés dans les trois colonnes du milieu. Pour l'interpolation dans le domaine spatial, la longueur des bras du personnage est considérablement réduite, ce qui est généralement un comportement indésirable lors de l'interpolation de corps humains. Les deux modèles parviennent à surmonter ce problème, c'est-à-dire qu'ils construisent un espace latent représentant l'ensemble des poses possibles pour un corps humain. Néanmoins, comme vu précédemment, la qualité de reconstruction de notre modèle est meilleure que celle de Neural3DMM, notamment pour une dimension latente de 16 utilisée ici, conduisant à des maillages générés plus propres. Il est important de noter que pour tous les modèles, si la taille de l'espace latent est plus grande, l'interpolation est similaire à celle illustrée dans la figure 4.6. La capacité d'interpolation n'est bien représentée que pour une taille d'espace latent faible.

4.7 Expériences supplémentaires

Jusqu'ici, les bases de données utilisées étaient composées de maillages centrés à l'origine et orientés dans la même direction afin de simplifier le traitement par les modèles. Cette caractéristique n'est pas représentative de la manière dont peut être capturé le mouvement de corps humains puisqu'un sujet est susceptible de se déplacer et de tourner dans un espace en 3 dimensions. Incorporer cette information de translation et de rotation des maillages dans le processus peut représenter une difficulté : cette problématique sera abordée dans cette section. Ensuite, nous verrons que la manière de se déplacer pour un corps humain, par exemple en marchant, est une action qui nécessite une architecture plus complexe afin d'être comprise par le modèle. Finalement, nous aborderons le sujet de la séparation de l'information de style et de pose qui peut poser des problèmes propres au traitement de corps humains.

4.7.1 Inclure l'information de translation et de rotation

Les bases de données DFaust et AMASS introduites dans la section 4.3 ont été générées sans utiliser les informations de translation et de rotation disponibles. Leurs moyennes et écarts types sont illustrés dans les figures 4.1 et 4.2. Ici, nous allons observer comment l'introduction de maillages correctement positionnés dans la scène peut perturber l'apprentissage du meilleur modèle introduit précédemment (SAE-LP) et comment il est possible de résoudre cette problématique.

Les figures 4.19 et 4.20 illustrent les moyennes et écarts types des bases de données DFaust et AMASS en incluant ces informations précédemment omises.

Si l'on compare les figures 4.1 et 4.19, on observe peu de différences : cela est dû au fait que les mouvements composant la base DFaust sont peu variés, la plupart représentant des actions réalisées sans déplacement et rotation. En revanche, si l'on compare les figures 4.2 et 4.20, on observe une réelle différence. Dans le domaine spatial ou spectral, cette différence se reflète par de plus grandes variations sur les axes X et Z, tandis que sur l'axe vertical Y les variations sont similaires. La dispersion des données étant plus grande dans la base AMASS, nous pouvons nous concentrer sur cette dernière afin de réaliser des tests.

Afin de pouvoir traiter séparémment les informations de translation et de rotation, il faut pouvoir les isoler. Il serait possible de les récupérer directement depuis le processus SMPL, mais l'idée est de s'affranchir de ce formalisme et d'utiliser plutôt une méthode indépendante en les récupérant depuis le domaine spectral. La figure 4.21 illustre le 24e vecteur propre sur un maillage quelconque. Contrairement aux autres vecteurs propres (voir figure 3.7), les zones affectées par ce dernier sont bien réparties entre l'avant et l'arrière de la forme. Cela signifie qu'il est possible de calculer l'angle entre le 24e vecteur 3D des coefficients spectraux et l'axe avec lequel on veut aligner le maillage et enfin d'appliquer à tous les coefficients spectraux une rotation de cet angle. En restant dans l'espace spectral, il est donc possible d'aligner un maillage avec un axe choisi. De même, comme visualisé dans la figure 3.7 en haut à gauche, le premier vecteur propre a une valeur constante sur toute la surface. Cela signifie qu'il est possible de translater le maillage en manipulant simplement le premier coefficient spectral.

Le principe est donc le suivant : au lieu d'encoder directement les coefficients spectraux (voir figure 4.8), le vecteur de translation T (correspondant au premier coefficient spectral) et l'angle Θ entre le vecteur propre 24 et un axe choisi sont conservés. Ensuite, le premier coefficient spectral est mis à zéro (le maillage résultant se retrouve à l'origine) et une rotation de Θ est appliquée à tous les coefficients spectraux (le maillage résultant se retrouve orienté). Les coefficients spectraux centrés et orientés sont ensuite encodés comme dans la figure 4.8, et la variable latente résultante est concaténée aux deux valeurs isolées précédemment (T et Θ). Similairement, avant de décoder, ces deux valeurs sont d'abord isolées, la variable latente est décodée, et les coefficients spectraux en sortie sont translatés de T et orientés.

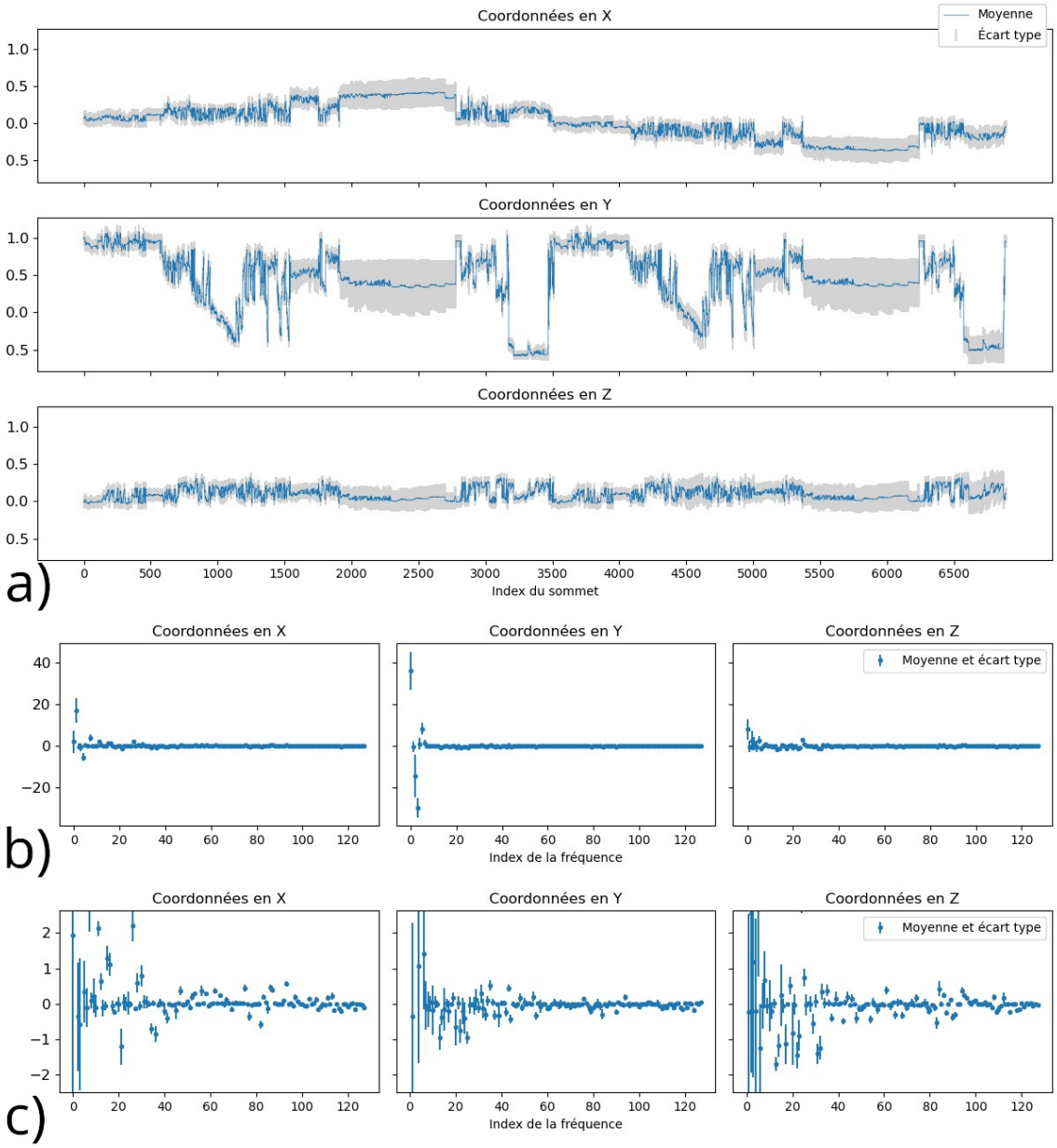


FIGURE 4.19 – Illustration des moyennes et des écarts types de la base de données DFaust. Les maillages sont correctement positionnés et orientés dans la scène. a) axes X, Y et Z représentés dans le domaine spatial. b) axes représentés dans le domaine spectral. c) axes représentés dans le domaine spectral en utilisant un zoom. Pour le domaine spectral, seulement 128 fréquences sont représentées.

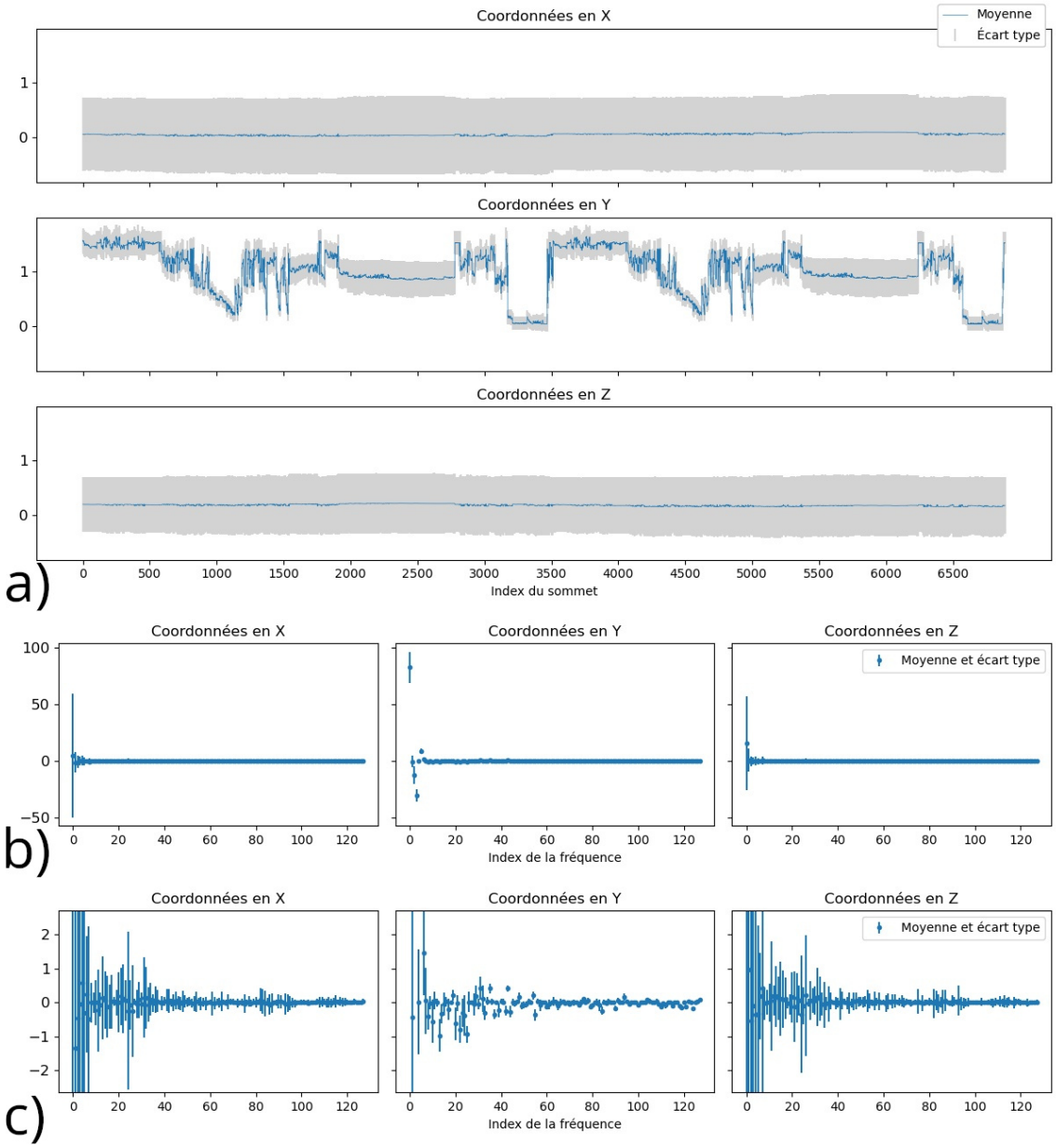


FIGURE 4.20 – Illustration des moyennes et des écarts types de la base de données AMASS. Les maillages sont correctement positionnés et orientés dans la scène. a) axes X, Y et Z représentés dans le domaine spatial. b) axes représentés dans le domaine spectral. c) axes représentés dans le domaine spectral en utilisant un zoom. Pour le domaine spectral, seulement 128 fréquences sont représentées.

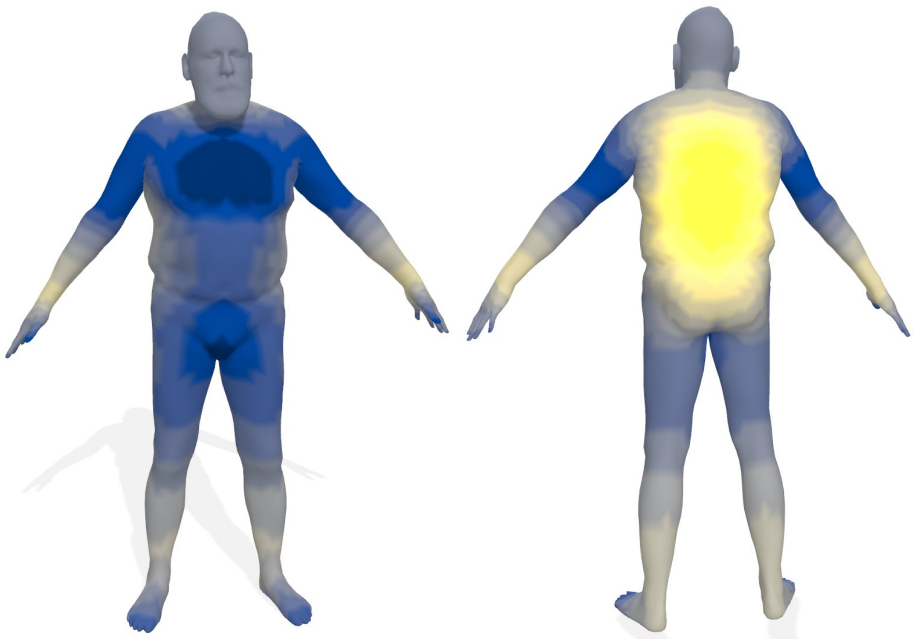


FIGURE 4.21 – Visualisation du 24e vecteur propre.

Ce processus représente un contournement, ou un court-circuit, de l'information : les valeurs de translation et de rotation ne sont pas encodées par le modèle mais plutôt directement données à l'espace latent.

La figure 4.22 présente l'évolution des courbes d'apprentissage et de test du modèle précédemment présenté, le SAE-LP, et d'un modèle SAE-LP-CR (SAE-LP *Centered Rotated*) présenté ici. Le modèle SAE-LP-CR a plus de facilités pour reconstruire les échantillons vus durant l'entraînement ainsi que des échantillons non vus.

Les figures 4.23 et 4.24 présentent des exemples visuels de reconstruction et d'interpolation depuis des modèles avec et sans utilisation du centrage et rotation. Dans les deux exemples, l'interpolation spatiale des deux maillages aux extremités (première ligne et colonne du milieu) génère une surface non réaliste et aplatie en raison de la différence d'orientation des références. Le modèle SAE-LP ne reconstruit pas correctement les maillages de référence et interpole dans l'espace latent de manière non réaliste. En revanche, le modèle SAE-LP-CR parvient à mieux reconstruire les surfaces. Du fait du traitement indépendant de rotation, l'interpolation génère une surface qui n'est pas aplatie. Enfin, la compression réalisée par l'architecture autoencodeur permet de construire un espace latent qui respecte les dimensions naturelles d'un corps humain : la longueur des bras du maillage interpolé sur la figure 4.24 est réaliste.

Une vidéo disponible en ligne (`https://youtu.be/OdGaT4Vxi9A`) montre un exemple d'interpolation entre deux surfaces éloignées dans la scène. L'effet de glissement des pieds sur le sol est causé par la méthode d'apprentissage : bien que le

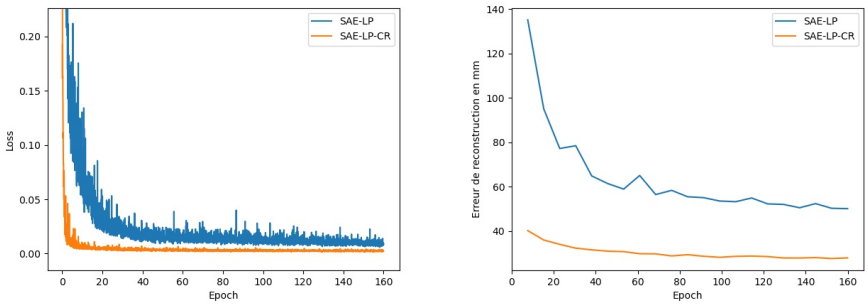


FIGURE 4.22 – Différentes valeurs d'entraînement sur la base de données AMASS dont les échantillons ne sont ni centrés ni orientés. À gauche, l'évolution de la fonction de coût, et à droite l'évolution du score de validation. Deux modèles sont évalués : le SAE-LP présenté précédemment, et le SAE-LP-CR dans lequel l'information de translation et de rotation est traitée différemment.

modèle réussisse à générer des surfaces qui ne sont ni aplaties ni mal dimensionnées, le réseau ne peut pas comprendre par lui-même que la manière de se déplacer dans l'espace doit se faire de manière réaliste. Ce mouvement dans la scène peut être représenté par de la marche ou de la course, de manière lente, rapide, ou avec des styles différents. Similairement à des mouvements de bras entre deux poses, le corps humain peut se déformer de plusieurs façons, et l'utilisation d'un autoencodeur n'est pas apte à représenter cette variété de mouvements. Pour pouvoir donner cette information de déplacement au réseau, plusieurs méthodes sont possibles.

Si l'on veut conserver l'utilisation d'un unique autoencodeur, on peut par exemple essayer de modifier la structure de l'espace latent en forçant l'interpolation à ressembler à une vérité terrain, disponible dans les bases de données. Nous avons testé cette solution : un sous-ensemble d'AMASS composé de mini-séquences a d'abord été construit. Ensuite, ce sous-ensemble a été utilisé pour essayer de modifier l'interpolation entre les poses de départ et de fin de ces mini-séquences afin qu'elle soit réaliste. Mais cette méthode fait intervenir trop de choix de l'utilisateur lors de la création des mini-séquences, étant donné qu'il ne faut pas donner des exemples non-apprenables : si le base de données de mini-séquences est composées de mouvements trop variés (par exemple, un mouvement représentant un pas peut être réalisé avec la jambe droite ou gauche), alors le réseau ne parviendra pas à apprendre.

Autrement, on peut penser encoder directement des animations au lieu de poses statiques à l'aide d'architectures comprenant le contexte d'une séquence de données. C'est plutôt cette solution qui a été adoptée, pour laquelle plus de détails sont donnés dans le chapitre 5.

4.7.2 Séparer l'information d'identité et de pose

Une propriété désirable d'un modèle produisant des surfaces est de pouvoir contrôler la capacité de génération. La séparation entre l'information d'identité et de

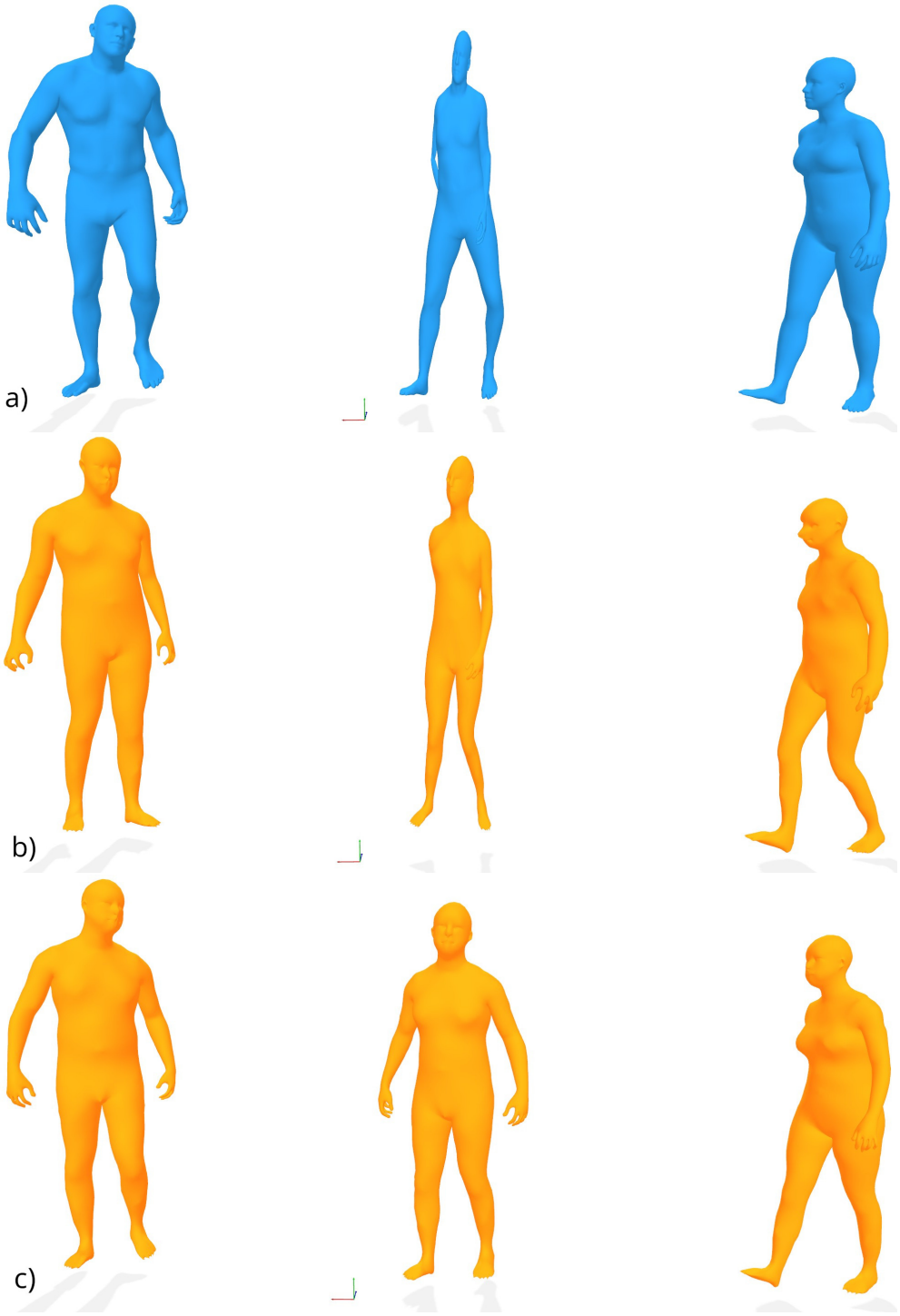


FIGURE 4.23 – Différentes reconstructions et interpolations avec et sans utilisation du centrage et de la rotation. a) : interpolation dans le domaine spatial. b) : la méthode SAE-LP ne fonctionne plus en raison de l'information supplémentaire de déplacement et de rotation à apprendre. c) : la méthode SAE-LP-CR fonctionne puisque l'interpolation est réalisée directement en tenant compte des valeurs brutes de translation et de rotation. Le repère indiquant l'origine de la scène est indiqué pour montrer la différence de translation et d'orientation des maillages interpolés.

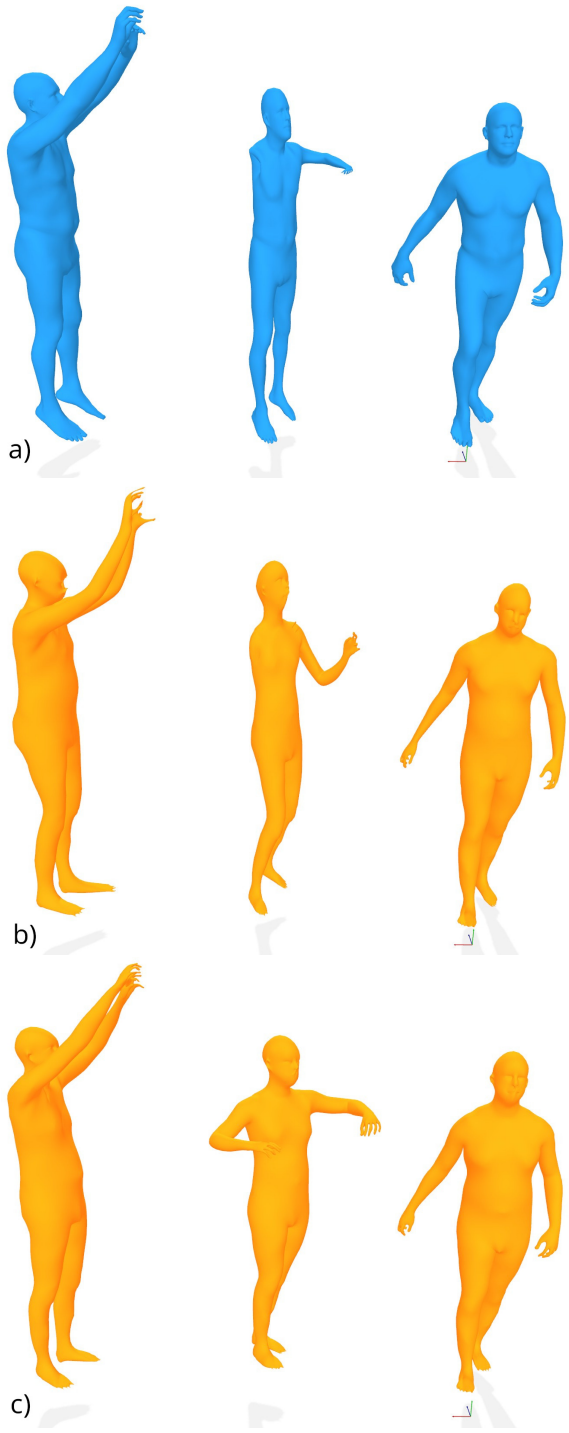


FIGURE 4.24 – Différentes reconstructions et interpolations avec et sans utilisation du centrage et de la rotation. a) : interpolation dans le domaine spatial. b) : la méthode SAE-LP ne fonctionne plus en raison de l'information supplémentaire de déplacement et de rotation à apprendre. c) : la méthode SAE-LP-CR fonctionne puisque l'interpolation est réalisée directement en tenant compte des valeurs brutes de translation et de rotation. Le repère indiquant l'origine de la scène est indiqué pour montrer la différence de translation et d'orientation des maillages interpolés.

pose (*disentanglement*) des sujets étudiés est une manière de faire cela : la création d'un espace latent "démêlé" permet de pouvoir changer une partie de ce dernier afin de ne modifier qu'un facteur de variation en étant invariant aux autres (modification de l'identité seulement sans changer la pose ou inversement). Cette problématique représente aujourd'hui toujours un challenge. Aumentado-Armstrong et al. [8] exploitent l'invariance aux isométries de l'opérateur de Laplace-Beltrami couplée à une double architecture autoencodeur/autoencodeur variationnel afin d'obtenir un modèle pouvant réaliser cette tâche. Foti et al. [51] introduisent une fonction de coût basée sur la projection de distances avec un maillage de référence sur les vecteurs propres du Laplacien topologique. Ces deux travaux récents comprennent un état de l'art concernant le démêlement de l'information appliqué à des surfaces.

Dans nos travaux, l'objectif est de réussir à séparer cette information de manière semi-supervisée. Les bases de données de corps humain en mouvement contiennent des annotations d'identité : chaque animation correspond à un sujet. En revanche, l'information de pose n'est pas annotée et nécessiterait une trop grande quantité de travail de la part d'utilisateurs pour que ce soit le cas. Les méthodes de l'état de l'art reposent donc sur le calcul de propriétés géométriques dans le domaine spatial ou spectral combiné à des méthodes d'apprentissage profond. Dans le chapitre suivant, nous montrons qu'il est possible d'apprendre à un réseau comment modifier des coefficients spectraux afin que les séquences de surfaces générées présentent un mouvement réaliste tout en gardant l'identité du sujet traité, ce qui représente une forme de séparation de l'information d'identité et de pose.

4.8 Conclusion

Le problème avec les méthodes actuelles de l'état de l'art est qu'elles sont limitées par le nombre élevé de sommets, la grille non structurée et le temps de calcul. Nous avons montré dans ce chapitre qu'en utilisant des méthodes de traitement spectral sur des surfaces triangulées, nous sommes capables de résoudre plusieurs de ces problèmes. Premièrement, traiter directement les coefficients spectraux au lieu des coordonnées cartésiennes permet l'application directe de convolutions puisque les coefficients spectraux sont ordonnés en fonction de leur valeur propre associée. Cette application directe permet également d'augmenter la vitesse des réseaux puisque les convolutions se font sur des tableaux sans réarrangement contrairement aux méthodes comparées. De plus, il est possible d'augmenter le nombre de sommets des maillages de la base de données tout en gardant la même architecture de réseau de neurones puisqu'il n'y a pas de calcul dans le domaine spatial. Enfin, nous avons montré que nos modèles donnent de meilleurs résultats que les méthodes de l'état de l'art en termes de reconstruction et d'interpolation.

Bien que notre méthode nécessite toujours une connectivité constante, il est possible d'envisager dans le futur de pouvoir synchroniser des bases calculées à partir de différentes triangulations en utilisant des cartes fonctionnelles (voir section 3.4). Des travaux additionnels pourraient s'attacher à essayer de généraliser ce processus

à des formes de topologie arbitraire, permettant de travailler sur des maillages issus directement de scans bruts avec un nombre élevé de sommets et une connectivité changeante tout en gardant le même nombre de paramètres et la même vitesse de calcul.

En résumé, notre approche repose sur l'utilisation de l'analyse spectrale pour extraire les informations cruciales contenues dans les maillages triangulaires tout en réduisant la complexité des données en entrée. Cette méthode pourrait permettre d'ouvrir la voie à des améliorations significatives dans le domaine du traitement de maillages, avec des applications potentielles dans de nombreux domaines comme la modélisation 3D, la reconnaissance d'objets et bien d'autres.

Chapitre 5

Apprentissage et génération de séquences de maillages

5.1 Introduction

Dans le chapitre précédent, un modèle permettant la création d'une représentation adaptée au corps humain a été introduit. Mais ce modèle ne traite l'information que de manière statique et n'est pas capable de correctement prendre en compte la dimension temporelle. Les progrès récents concernant la technologie de capture de formes en mouvement ont rendu l'acquisition moins chère et plus efficace tout en rendant les bases de données de maillages dynamiques plus disponibles et plus détaillées. Le coût et la difficulté d'obtenir ces scans restent tout de même non négligeables et développer des processus capables de comprendre ou de générer des données dynamiques et réalistes est aujourd'hui un besoin nécessaire. Cet aspect est abordé dans ce chapitre.

Les travaux récents concernant la modélisation du mouvement humain traitent ce problème en prenant en entrée l'information concernant des séquences de squelettes ([78, 91, 90, 92, 57, 122, 154]), réduisant d'un facteur important la dimension du problème. L'inconvénient de ces méthodes est qu'elles n'ont pas accès à la géométrie et nécessitent des étapes supplémentaires de squelettage (*rigging*) et de mise en correspondance des surfaces avec les squelettes (*skinning*) afin de générer des maillages. D'autres méthodes [97] prennent en entrée une représentation simplifiée telle que les paramètres SMPL [87] qui contiennent cette information géométrique mais elles sont spécifiques à certains jeux de données et nécessitent encore un processus supplémentaire pour générer des maillages. Il existe aussi des méthodes capables de traiter des formes mais elles ne reposent que sur les poses statiques, sans tenir compte du mouvement comme nous l'avons vu dans le chapitre précédent. Dans ce travail, notre objectif est de montrer qu'en utilisant des méthodes d'analyse spectrale, il est possible de traiter efficacement des surfaces en prenant également en compte la dynamique.

Le processus présenté ici est basé sur l'utilisation de ce domaine spectral. Il consiste en un autoencodeur convolutif capable de représenter un maillage statique

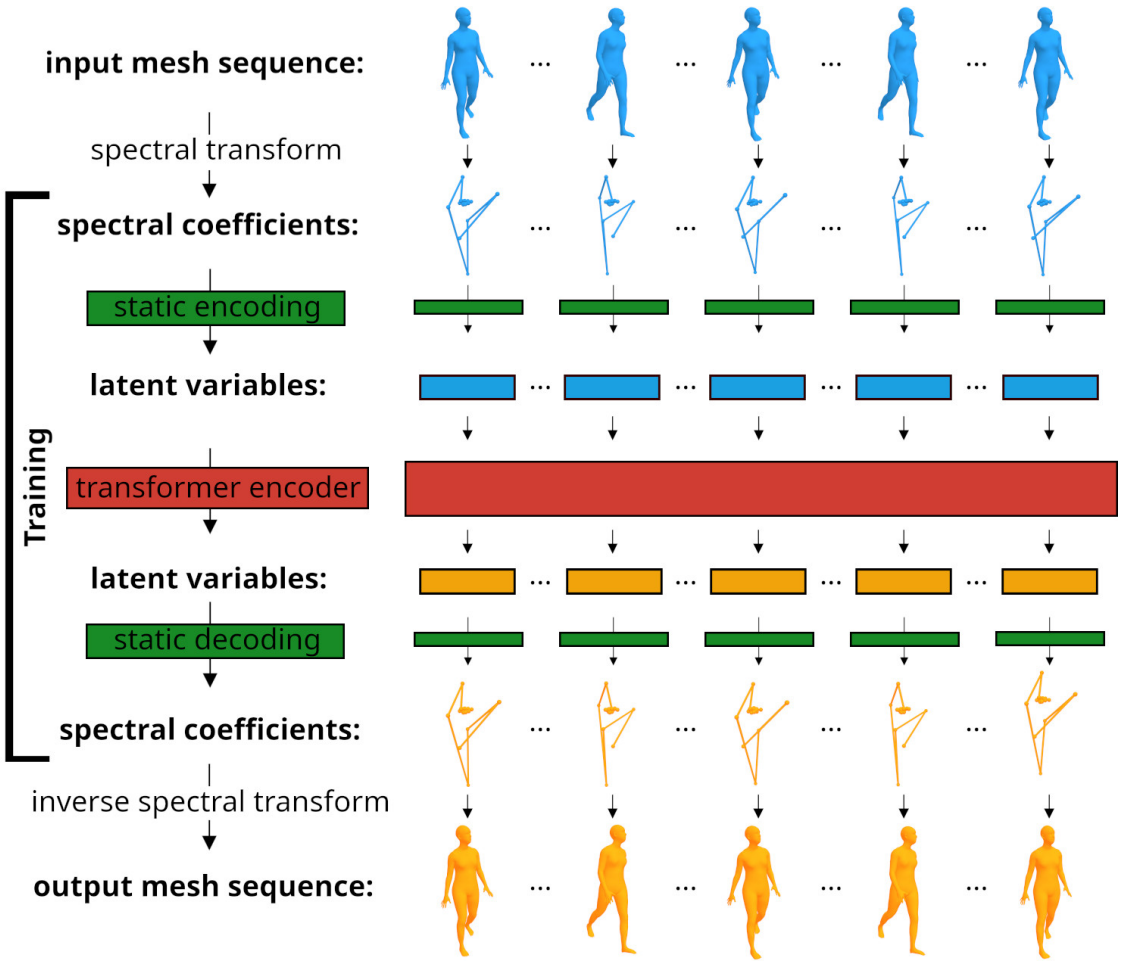


FIGURE 5.1 – Illustration du processus général proposé. Les sommets de chaque pose d'une séquence de maillage sont d'abord transformés en coefficients spectraux. Ils sont ensuite passés à travers un encodeur convolutif afin d'obtenir des variables latentes, qui sont l'entrée d'un transformeur. Les variables latentes générées par le transformeur sont ensuite décodées par un décodeur convolutif, et finalement transformées dans le domaine spatial. L'apprentissage se fait uniquement dans le domaine spectral. Les données en entrée sont en bleu, les données en sortie sont en orange, les paramètres apprenables pour le processus statique sont en vert et les paramètres apprenables pour le processus dynamique sont en rouge.

en une variable latente en prenant en entrée des coefficients spectraux au lieu de sommets dans le domaine spatial dont le fonctionnement est expliqué en détail dans le chapitre précédent. Cette architecture est couplée à un transformeur [135], réseau récemment introduit capable de comprendre le contexte de séquences de données. Il est basé sur BERT [41] (utilisation de la partie encodeur du transformeur uniquement) et est principalement évalué sur une tâche de prédiction de la fin d'une séquence. Des expériences supplémentaires montrent la possibilité d'étendre facilement ses applications (prédiction à long terme, complétion de parties manquantes (*in-betweening*) et généralisation à d'autres ensembles de données). L'utilisation du domaine spectral permet un traitement rapide et le transformeur permet un processus parallélisable qui peut générer un mouvement humain de manière non autorégressive (dans ce contexte, une génération non autorégressive signifie que plusieurs poses sont générées en une seule passe grâce au transformeur, contrairement aux réseaux de neurones récurrents qui génèrent des échantillons en se basant sur les générations précédentes, pas à pas). Le fonctionnement du modèle est représenté dans la figure 5.1.

Les contributions se résument comme suit :

1. nous utilisons les coefficients spectraux obtenus depuis la géométrie de maillages triangulaires humains à connectivité constante comme entrée du réseau.
2. nous n'entraînons le réseau que dans le domaine spectral sans revenir dans le domaine spatial, ce qui accélère le processus.
3. nous utilisons une double architecture composée d'un autoencodeur convolutif et de l'encodeur d'un transformeur capable de prédire les poses futures (à court ou long terme), ou de compléter les poses manquantes d'une séquence de maillages.
4. l'espace latent de l'autoencodeur convolutif est utilisé comme entrée et sortie du transformeur, lui donnant accès à l'information contenue dans les surfaces.
5. le processus fonctionne de manière directe (non autorégressive), permettant la génération efficace et rapide de mouvements humains.

Les modèles pré-entraînés, le code pour les entraîner et le code pour créer des ensembles de données sont disponibles à l'adresse suivante : https://github.com/MEPP-team/SpecTrHuMS.

Des travaux similaires seront premièrement introduits, puis le fonctionnement de notre modèle sera présenté, et enfin, des résultats d'expériences montreront la capacité de génération du réseau.

5.2 État de l'art

Pour un état de l'art concernant le traitement statique de maillages, le lecteur est renvoyé au chapitre précédent.

La génération de mouvement humain a récemment suscité beaucoup d'intérêt dans la littérature. Puisque notre application principale est la prédiction, nous nous concentrerons sur les articles récents concernant cet objectif. La prédiction du mouvement humain est généralement exprimée sous la forme d'un processus séquence à séquence où le mouvement observé est représenté comme l'entrée du modèle. Les premières méthodes utilisaient un processus gaussien [138], la machine de Boltzmann restreinte [130] ou des modèles de Markov [72]. Des méthodes plus récentes utilisent des réseaux de neurones récurrents (RNNs) pour des prédictions plus longues et plus précises [52, 64] mais souffrent toujours de discontinuités et ne sont entraînées que sur des actions spécifiques. Ces inconvénients ont été corrigés en travaillant sur des vitesses de jointures de squelettes [98] ou avec des variantes de RNNs [31, 86]. Mais l'entraînement et l'inférence sont difficiles avec les RNNs et ont des contraintes de mémoire, donc la prédiction a été encore améliorée en utilisant des fenêtres glissantes [21, 22], des modèles convolutifs [59, 78] ou des modèles similaires à des GANs [56, 59]. Ensuite, d'autres travaux utilisent des réseaux convolutifs sur des graphes (*Graph Convolutional Networks* ou GCNs) [66]) pour mieux traiter l'information spatiale des jointures de squelettes. Le premier utilise des coefficients DCT pour coder les informations temporelles dans l'espace fréquentiel [91]. Différentes tailles de couches convolutives couplées à un GCN sont utilisées dans [71], et des variantes de GCN sont utilisées dans [122, 154, 88]. Un réseau entièrement connecté est couplé à des convolutions pour coder les informations temporelles et spatiales dans [21]. Certains travaux utilisent l'architecture transformeur [135] couplée à un RNN [128], en combinant l'attention pour corréler les informations temporelles et spatiales [2], ou en combinant l'attention avec les coefficients DCT [23]. D'autres utilisent l'attention avec des GCN et des coefficients DCT [90] ou en fusionnant les prédictions de trois modules d'attention qui traitent le mouvement à différents niveaux : corps entier, parties du corps et articulations individuelles [92]. De plus, de la complétion de mouvement est réalisée dans [45] en utilisant une architecture transformeur. Guo et al. [57] utilisent un réseau entièrement connecté à plusieurs couches (*Multilayer Perceptron* ou MLP) et prouvent qu'un modèle simple est capable de donner les meilleurs résultats. Une autre ligne de travail génère des mouvements humains conditionnés depuis du texte en utilisant un autoencodeur variationnel (VAE) [108] ou en utilisant un modèle de diffusion [131]. Bien que le conditionnement depuis du texte sorte du cadre de ce document, toutes les méthodes citées ont en commun le fait que seule l'information des squelettes est exploitée, ce qui fait que les processus n'ont pas accès aux détails contenus dans les surfaces.

Marsot et al. [97] utilisent un autoencodeur variationnel conditionnel (*Conditional Variational Autoencoder* ou CVAE) [123] afin de créer un espace latent permettant de représenter et de générer des mouvements humains en prenant en entrée des paramètres SMPL, prenant ainsi en compte l'information géométrique. Ceci est similaire à notre cas puisque ces paramètres représentent un espace compact comme celui que nous utilisons et à partir duquel un maillage peut être récupéré. Néanmoins, des processus comme SMPL sont dédiés aux corps humains, ou du moins à des surfaces qui peuvent être approximées par des squelettes comme des animaux

[157]. Contrairement à cela, notre modèle peut être facilement généralisé à d'autres types de surfaces dynamiques qui ne peuvent pas être approximées avec des squelettes (voir la section 5.4.6). De plus, le modèle de Marsot et al. [97] a besoin d'un espace en entrée dans lequel les informations de pose et d'identité ont été séparées comme SMPL, tandis que le nôtre a l'espace des coefficients spectraux en entrée dans lequel ces informations sont intriquées, le rendant applicable à une plus large gamme de bases de données.

Fernandez-Abrevaya et al. [48] ont introduit un processus qui permet d'étendre l'opérateur de Laplace-Beltrami à des séquences de maillages temporellement cohérentes, incluant ainsi l'information de surface et temporelle et permettant d'éditer une séquence de maillages de manière simple. S'il serait intéressant d'observer comment des modèles d'apprentissage profond pourraient utiliser l'information spectrale provenant de cet opérateur, il est peu pratique de l'introduire dans notre situation : similairement à l'opérateur de Laplace-Beltrami dans un cadre statique, chaque séquence nécessiterait un nouveau calcul et une nouvelle décomposition en éléments propres, créant à chaque fois une nouvelle base. Dans notre cas, en utilisant le Laplacien topologique de manière statique, un seul calcul de vecteurs propres est nécessaire.

Dans ce chapitre, nous montrons qu'il est possible de développer des applications telles que la prédiction de mouvements humains tout en donnant en entrée au réseau l'information contenue dans les surfaces en utilisant un traitement spectral des maillages couplé à un autoencodeur convolutif et une architecture transformeur. Dans la section suivante, le processus du modèle est présenté.

5.3 SpecTrHuMS

Dans ce chapitre est présenté un transformeur spectral pour apprendre depuis des séquences de maillages humains (*Spectral Transformer for Human Mesh Sequence learning* ou SpecTrHuMS). Le procédé que nous proposons est composé d'une double architecture : la première est l'autoencodeur spectral (*Spectral Autoencoder* ou SAE) présenté dans le chapitre précédent, et le second est un encodeur de type transformeur [135]. Le traitement spectral de maillages a été abordé dans le chapitre 3. Ici, nous rappelons d'abord le fonctionnement du SAE, et expliquons comment nous le couplons avec la deuxième architecture transformeur.

Nous utilisons l'architecture SAE-LP du chapitre précédent qui consiste en des couches successives de convolutions, de sous/sur-échantillonnage (*pooling/upsampling*) utilisant des multiplications matricielles et des couches d'activation, créant un espace latent à partir duquel la reconstruction est possible. Pour rappel, le processus statique est le suivant : un maillage composé de n sommets est représenté par une matrice $V \in \mathbb{R}^{n\times 3}$. Les coordonnées cartésiennes des sommets sont transformées en coefficients spectraux $C \in \mathbb{R}^{k\times 3}$ avec une transformée spectrale : $C = \Phi^T \cdot V$, où $k <= n$ est le nombre de vecteurs propres du Laplacien topologique utilisé et Φ est la

matrice les représentant. Ces coefficients spectraux C sont passés dans l'encodeur du SAE afin d'obtenir une variable latente $X \in \mathbb{R}^l$, l étant la taille de l'espace latent représentant un espace plus compact que le domaine spectral. Cet encodeur est fait de blocs composés de couches de convolution, de sous-échantillonnage et d'activation, suivis d'une couche finale entièrement connectée. Dans le paragraphe suivant, nous montrerons comment les variables latentes d'une séquence de maillages sont passées à travers l'architecture du transformeur. La variable latente en sortie du transformeur est ensuite passée dans le décodeur du SAE afin d'obtenir les coefficients spectraux reconstruits $\hat{C} \in \mathbb{R}^{k\times 3}$. Le décodeur est composé d'une première couche entièrement connectée suivie de blocs composés de couches de sur-échantillonnage, de convolutions et d'activation. Pour la visualisation, les coefficients spectraux reconstruits sont transformés dans le domaine spatial pour obtenir les sommets reconstruits avec une multiplication matricielle $\hat{V} = \Phi \cdot \hat{C}$ avec $\hat{V} \in \mathbb{R}^{n\times 3}$. Le choix des valeurs k et l est précisé dans la section 5.4.

Notre objectif est de transmettre un mouvement humain à un réseau de neurones afin qu'il comprenne le contexte du mouvement qui est représenté comme une séquence de sommets spatiaux $V^{1:t} \in \mathbb{R}^{t\times n\times 3}$ avec t le nombre de poses de la séquence. Nous exploitons l'architecture transformeur largement utilisée dans la littérature pour la deuxième partie de notre réseau. Afin de pouvoir donner l'information contenue dans la séquence de sommets spatiaux, nous transformons d'abord chaque pose dans le domaine spectral, donnant une séquence de coefficients spectraux $C^{1:t} \in \mathbb{R}^{t\times k\times 3}$, puis encodons chaque pose de la séquence de coefficients spectraux avec l'encodeur SAE introduit précédemment, donnant une séquence de variables latentes $X^{1:t} \in \mathbb{R}^{t\times l}$.

La génération est formulée comme un problème séquence à séquence. Pour les variables latentes en entrée qui doivent être prédites, avant de les donner au modèle transformeur, soit nous répétons la dernière pose de la séquence si l'objectif est la prédiction, qui est la tâche principale sur laquelle le modèle est évalué, soit nous utilisons une interpolation linéaire pour remplir les valeurs manquantes si l'objectif est la complétion, ce qui est une tâche supplémentaire. Ensuite, la fonction de coût est calculée comme l'erreur entre la sortie du transformeur et les vérités terrain afin d'entraîner le modèle. La figure 5.2 montre comment les séquences sont données au modèle dans les deux cas. Nous sous-échantillonnons chaque séquence de 3 secondes à 25Hz (plus d'informations sur les bases de données utilisées sont données dans la section 5.4), ce qui signifie que $t = 75$. Pour la prédiction, les variables $X^{1:50}$ sont des vérités terrain, et les variables $X^{51:75}$ sont représentées par la variable X^{50} répétée. Pour la complétion, les variables $X^{1:50}$ et X^{75} sont des vérités terrain, et les variables $X^{51:74}$ sont des interpolations entre les variables X^{50} et X^{75}.

Le processus dans la partie transformeur est le suivant. Tout d'abord, la séquence en entrée de variables latentes $X^{1:t}$ est projetée sur la dimension du transformeur avec une couche entièrement connectée. Cela donne un ensemble de variables latentes projetées $X_p^{1:t} \in \mathbb{R}^{t\times d}$, d étant la dimension de sortie de la couche entièrement connectée et la dimension du transformeur. La conception de l'architecture du

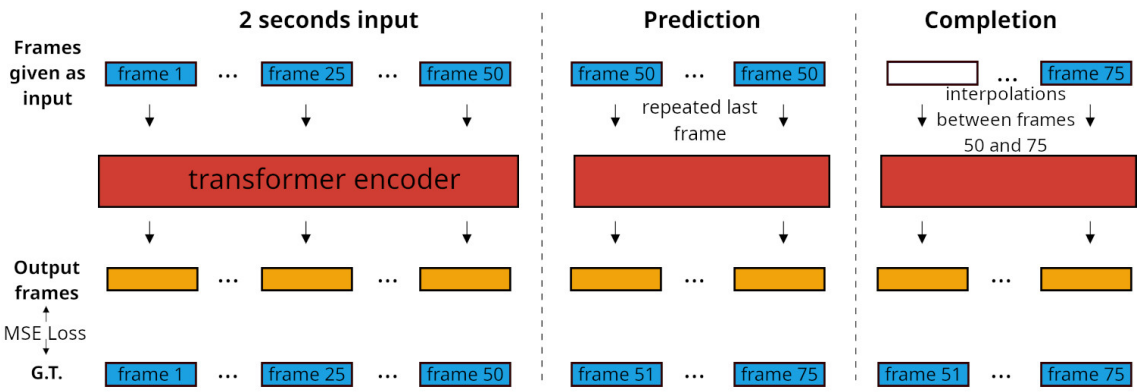


FIGURE 5.2 – Processus du transformeur. Pour la prédiction et la complétion, 2 secondes, ou 50 poses à 25 Hz, sont données en entrée au transformeur (à gauche). Pour la prédiction (au milieu), la dernière pose de la séquence en entrée est répétée pour la partie finale de la séquence. Pour la complétion (à droite), la partie finale est remplacée par une interpolation entre la dernière pose connue et la dernière pose de la séquence. Ensuite, une erreur quadratique moyenne (*Mean Squared Error* ou MSE) est calculée entre la sortie du transformeur et la vérité terrain en tant que fonction de coût.

transformeur l'empêche de connaître l'ordre des poses en entrée, donc les variables latentes sont additionnées avec un codage positionnel standard $P_e \in \mathbb{R}^{t \times d}$. Le codage positionnel est une matrice apprenable de fonctions sinusoïdales. Le résultat des variables latentes projetées est $X_e^{1:t} = X_p^{1:t} + P_e$, avec $X_e^{1:t} \in \mathbb{R}^{t \times d}$.

Nous utilisons l'architecture encodeur d'un transformeur standard pour traiter les variables latentes projetées. Il est composé de plusieurs couches d'encodeur, chacune constituée de couches d'auto-attention multi-têtes (*multi-head self-attention layers*) et de réseaux de neurones à propagation avant (*feed-forward networks*), avec une connexion résiduelle entre les deux couches et une couche de norme (*norm layer*) finale. Les couches d'attention multi-têtes permettent de réaliser un calcul dense entre chaque paire de poses en entrée et de capturer des relations à longue portée. Nous renvoyons le lecteur à la littérature [135] pour plus de détails sur le processus général de cette architecture. L'encodeur du transformeur produit une séquence de variables latentes reconstruites, qui est ensuite projetée dans la dimension de l'espace latent du réseau statique à l'aide de couches entièrement connectées, donnant une séquence $\hat{X}^{1:t} \in \mathbb{R}^{t \times l}$. Chaque pose de ces variables latentes reconstruites peut finalement être décodée par le décodeur convolutif du processus statique afin d'obtenir des coefficients spectraux reconstruits, qui peuvent ensuite être transformés dans le domaine spatial pour la visualisation.

Les deux réseaux (SAE et transformeur) sont entraînés simultanément. Le premier, représenté comme un encodeur et décodeur statique, est entraîné pour reconstruire les coefficients spectraux en entrée sans utiliser le réseau dynamique. La fonction de coût pour entraîner ce réseau statique est donc une erreur quadratique

moyenne entre les coefficients spectraux C et $\hat{C}$ sans utiliser le transformeur. Le deuxième réseau dynamique, le transformeur, est entraîné afin de reconstruire la séquence des variables latentes. La fonction de coût pour entraîner ce réseau est donc une erreur quadratique moyenne entre les séquences de variables latentes $X^{1:t}$ et $\hat{X}^{1:t}$.

5.4 Évaluations

Dans cette section, nous présentons les modèles auxquels notre réseau est comparé, les bases de données utilisées, les métriques d'évaluation et des détails concernant l'implémentation. Ensuite, des tâches de prédiction à court et long terme, de complétion et de généralisation à d'autres bases de données sont exposées.

5.4.1 Modèles de référence

Nous évaluons notre modèle par rapport à plusieurs modèles de référence. convSeq2Seq [78] utilise un modèle convolutif, mais comme pour les images, les convolutions ne prennent pas en compte les relations entre les éléments distants, ce qui est important pour la compréhension du comportement humain. LTD [91] utilise un GCN pour les informations spatiales en construisant des graphes depuis les articulations des squelettes et des coefficients DCT pour les informations temporelles, conduisant à un modèle réalisant déjà une bonne prédiction. Cependant, cette méthode était encore insuffisante et a été dépassé par Pose Motion Att [90] qui utilise de l'attention couplée à un GCN et des coefficients DCT, qui a été mis à jour avec Motion Att. + Post-fusion [92] en ajoutant les prédictions de trois autres modules fonctionnant au niveau du corps entier, des parties du corps et des articulations individuelles. Ce modèle a obtenu des résultats représentant l'état de l'art jusqu'à ce que siMLPe [57] introduise un perceptron multicouches simple (MLP) opérant sur les positions articulaires des squelettes codées avec une DCT. STS-GCN [122] et STG-GCN [154] utilisent des variantes de GCN, mais ont également été dépassés par siMLPe. Tous ces modèles de référence utilisent l'information contenue dans les articulations des squelettes, ce qui signifie que leurs architectures ne sont pas directement applicables aux sommets de surface en raison de leur grande dimensionnalité et de leur non-ordre. De plus, ces architectures ne sont destinées qu'à la tâche de prédiction. Notre architecture est quant à elle applicable à des surfaces même avec un nombre élevé de sommets, et l'utilisation de l'architecture du transformeur rend notre processus facilement généralisable à d'autres tâches telles que la complétion ou à l'entraînement sur des séquences de différentes longueurs. Enfin, l'utilisation d'une architecture transformeur pourrait être utile, à l'avenir, pour des étapes d'apprentissage plus complexes telles que le préentraînement avec masques combiné à un affinage des paramètres avec des fonctions de coût spécifiques à d'autres objectifs.

5.4.2 Bases de données

AMASS [89] est une unification de plusieurs autres bases de données utilisant la paramétrisation SMPL. Nous suivons [90, 92, 57] et utilisons plusieurs bases telles que CMU, KIT et d'autres (toutes faisant partie d'AMASS) comme ensemble d'entraînement et AMASS-BMLrub comme ensemble de test. Cet ensemble de données est différent de celui introduit dans le chapitre précédent : plutôt que de contenir des poses individuelles (1 sur 100, voir section 4.3), elle contient des actions dynamiques. Pour la première partie des expériences, nous reprenons les conditions de l'état de l'art et utilisons une identité unique sans mouvement de la main pour que nos résultats soient comparables. Les modèles entraînés sur cette base auront le suffixe OI pour *One Identity*. Pour la deuxième partie des expériences, puisque nous travaillons avec des surfaces et que l'identité d'un maillage a du sens, nous générons le même jeu de données en utilisant différentes identités issues des vérités terrain contenues dans AMASS afin de montrer que notre réseau est capable de comprendre la conservation de l'apparence. Les modèles entraînés sur cette base de données auront le suffixe MI pour *Multiple Identities*. Nous rappelons que les deux bases de données sont composées du même nombre de poses, mais l'une est composée d'une identité unique tandis que l'autre est composée d'identités multiples. De plus, comme elles utilisent la même discrétisation SMPL, les vecteurs propres ne sont calculés qu'une seule fois et sont utilisables pour les deux. De la même manière que dans les travaux comparés, nous ignorons translation et rotation globales des poses et sous-échantillonnons chaque séquence à 25Hz. Après traitement et pour la visualisation uniquement, les séquences sont suréchantillonnées à 60Hz.

Nous reprenons la méthode des travaux antérieurs pour sélectionner les séquences en entrée : nous fixons la longueur à 50 poses (2 secondes) et la longueur de sortie à 25 poses (1 seconde). Ces 75 poses sont sélectionnées à l'aide d'une fenêtre glissante sur les séquences AMASS avec un décalage de 5. Le tout est composé de 7 799 animations et donne un ensemble de 7 475 animations après filtrage puisque certaines animations sont plus courtes que la fenêtre nécessaire, soit 458 104 fenêtres, ce qui équivaut à environ 380 heures de vidéo. Le jeu de test est composé de 3 061 animations et donne 2 640 animations après filtrage et 145 443 fenêtres, soit environ 120 heures de vidéo.

5.4.3 Métriques d'évaluation

Comme nous nous concentrons sur des surfaces alors que les travaux précédents n'examinent que les squelettes, une comparaison directe n'est pas possible. Pour nous aligner sur l'état de l'art, nous commençons par présenter les résultats à l'aide de l'erreur de position moyenne par articulation (*Mean Per Joint Position Error* ou MPJPE) [63] calculée sur les coordonnées des jointures de squelettes 3D. Nous obtenons ces positions d'articulations en utilisant une matrice de régression disponible dans la paramétrisation SMPL qui permet de transformer les sommets du maillage en positions d'articulations 3D. Mais cette MPJPE ne compare que les mouvements

générés avec une vérité terrain, et un mouvement généré pourrait toujours être réaliste sans avoir un bon score en suivant cette métrique. Ainsi, puisque la MPJPE ne reflète pas parfaitement la qualité de la génération des mouvements, nous introduisons également une mesure de variation de longueur d'arête qui exprime la conservation de l'identité. Cette mesure est calculée comme la différence moyenne absolue entre les paires de longueurs d'arêtes correspondantes de la dernière pose connue du mouvement en entrée avec les poses prédites de sorte qu'elle ne dépende pas du mouvement exprimé par la vérité terrain mais plutôt de la conservation de l'identité. Cette mesure sera nommée *Mean Edge Length Variation* (MELV).

5.4.4 Implémentation

Le nombre de sommets des maillages provenant de SMPL est $n = 6\ 890$. Nous fixons la valeur des vecteurs propres utilisés à $k = 512$ sur les 6 890 disponibles puisque c'est suffisant pour avoir des surfaces assez détaillées et que cela allège le travail du couple de réseaux. Le réseau statique est composé de 3 couches pour l'encodeur et le décodeur chacun, et utilise une fenêtre de convolution de taille 3. La taille latente du réseau statique est fixée à 256, ce qui est identique à la dimension du transformeur. Le transformeur est composé de 4 têtes, 6 couches, a une dimension de 1 024 et a la fonction GELU comme activation.

Les coefficients spectraux sont d'abord précalculés et stockés. La fonction de perte du réseau statique est calculée sur les coefficients spectraux bruts pour donner plus d'importance aux basses fréquences (on rappelle que les coefficients spectraux de basses fréquences ont des amplitudes plus élevées que ceux de hautes fréquences). Les variables latentes en entrée sont standardisées et la fonction de coût du transformeur est calculée sur ces variables latentes standardisées. Les réseaux statique et dynamique sont entraînés simultanément. Le réseau statique est d'abord entraîné seul pendant deux époques pour qu'il apprenne d'abord à reconstruire approximativement les maillages statiques. À la troisième époque, l'entraînement du réseau dynamique commence et les deux réseaux continuent de s'entraîner conjointement. Comme les poids du réseau statique sont mis à jour pendant l'apprentissage et que les valeurs des variables latentes sont modifiées, nous utilisons l'algorithme de Welford [143] pour mettre à jour les moyennes et les écarts types des variables latentes pendant l'apprentissage pour la standardisation de ces dernières.

Nous utilisons PyTorch pour entraîner les réseaux pendant 100 époques, avec l'optimiseur ADAM, une taille de paquet (*batch*) de 32 et un taux d'apprentissage de départ de 10^{-4} sans échauffement (*warm-up*) mais avec un ordonnanceur qui réduit le taux d'apprentissage à 10^{-6} à la fin de l'entraînement. L'entraînement se fait sur une carte graphique GPU NVIDIA V-100. Plus de détails sur le nombre d'époques sont donnés dans la section des résultats quantitatifs. L'entraînement dure environ 12 heures.

Nous présentons d'abord les résultats sur la tâche principale de prédiction en utilisant la base de données qui suit les travaux comparés composée d'une seule

Dataset	AMASS-BMLrub							
Time (ms)	80	160	320	400	560	720	880	1000
repeating last frame	24.0	45.0	77.5	89.0	103.7	107.7	101.5	95.3
convSeq2Seq [78]	20.6	36.9	59.7	67.6	79.0	87.0	91.5	93.5
LTD-10-10 [91]	10.3	19.3	36.6	44.6	61.5	75.9	86.2	91.2
LTD-10-25 [91]	11.0	20.7	37.8	45.3	57.2	65.7	71.3	75.2
Pose Motion Att [90]	11.3	20.7	35.7	42.0	51.7	58.6	63.4	67.2
Motion Att. + Post-fusion [92]	11.0	20.3	35.0	41.2	50.7	57.4	61.9	65.8
siMLPe [57]	**10.8**	**19.6**	**34.3**	40.5	50.5	57.3	62.4	65.7
SpecTrHuMS-OI-ES (ours)	15.4	22.3	34.8	**39.9**	**47.6**	**52.8**	**56.9**	**59.6**
SpecTrHuMS-OI (ours)	11.9	21.3	39.1	46.2	56.	63.1	69.4	73.2

TABLE 5.1 – Nous comparons les scores MPJPE de deux de nos modèles sur la base de données utilisant une identité (OI) pour la prédiction à chaque pas de temps (sans tenir compte des poses prédites précédentes), l'un avec arrêt précoce (ES) et le second sans arrêt précoce. Lors d'un arrêt précoce, notre modèle est capable de donner de meilleurs résultats que l'état de l'art pour la prédiction à long terme.

identité et en utilisant celle composée de plusieurs identités. Ensuite, nous présentons d'autres applications facilement implémentées à partir de notre processus telles que la prédiction à long terme (en utilisant une génération autorégressive), la complétion (*in-betweening*) et nous montrons enfin que notre modèle peut être utilisé avec d'autres bases de données constituées de surfaces qui ne peuvent pas être approximées par des squelettes.

5.4.5 Application principale : prédiction

La tâche principale présentée dans ce travail est la prédiction : le modèle essaie de générer un mouvement de 1 seconde lorsqu'il a en entrée les 2 secondes précédentes (voir figure 5.2 au milieu). Afin d'être comparable avec la littérature, nous évaluons d'abord notre méthode sur une base de données composée d'une seule identité neutre. Dans ce cas, nous présentons deux modèles : un qui a été entraîné pendant quelques époques (SpecTrHuMS-OI-ES pour SpecTrHuMS utilisant une identité, *one identity*, avec arrêt précoce, *early-stopping*), et un autre qui a été entièrement entraîné (SpecTrHuMS-OI). Ensuite, les résultats sur la base de données composée de plusieurs identités sont mis en évidence avec un modèle nommé SpecTrHuMS-MI pour SpecTrHuMS utilisant plusieurs identités (*multiple identities*).

Résultats quantitatifs

Le tableau 5.1 présente les résultats de nos modèles par rapport à ceux des méthodes comparées en utilisant la méthodologie d'évaluation proposée dans [90, 92,

Dataset	AMASS-BMLrub							
Time (ms)	80	160	320	400	560	720	880	1000
repeating last frame	18.3	29.1	47.7	55.4	67.9	76.6	81.7	83.5
STS-GCN [122]	10.0	12.5	21.8	24.5	31.9	38.1	42.7	45.5
STG-GCN [154]	10.0	11.9	20.1	24.0	30.4	-	-	43.1
siMLPe [57]	**6.1**	**10.8**	**19.1**	**22.8**	**29.5**	**35.1**	**39.7**	42.7
SpecTrHuMS-OI-ES (ours)	13.9	17.2	23.8	26.8	32.	36.2	**39.7**	**42.**
SpecTrHuMS-OI (ours)	9.8	14.3	23.5	27.7	34.8	40.5	45.4	48.6

TABLE 5.2 – Nous comparons les scores MPJPE de deux de nos modèles sur une base de données composée d'une seule identité (OI) pour la prédiction moyenne à chaque pas de temps (en tenant compte des poses prédites précédentes), l'un avec arrêt précoce (ES) et le second sans arrêt précoce. Lors d'un arrêt précoce, notre modèle est capable de donner de meilleurs résultats que l'état de l'art pour la prédiction à long terme.

57]. Dans ce tableau, le score d'évaluation ne tient compte que des pas de temps prédits et ne prend pas en compte les prédictions des pas de temps précédents (la MPJPE est calculée entre la vérité terrain et la prédiction seulement au pas de temps correspondant). Le tableau 5.2 présente les résultats obtenus grâce au protocole utilisé dans [122, 154], où les évaluations sont calculées en prenant la moyenne sur les poses précédentes. Les valeurs des autres modèles sont tirées des articles comparés. Nous rapportons également dans les tableaux 5.1 et 5.2 les résultats de la MPJPE lors de la simple répétition de la dernière pose connue comme référence. Pour rappel, nous nous concentrons sur les surfaces, alors que les méthodes comparées opèrent sur les positions des jointures de squelettes. Par conséquent, notre réseau traite des informations supplémentaires qui intègrent les éléments d'identité des surfaces. Bien que nos résultats soient moins précis pour des pas de temps plus courts, notre réseau, lorsqu'il est arrêté tôt, est capable de mieux prédire le mouvement à des pas de temps plus longs en utilisant la métrique MPJPE. Cependant, nous allons voir qu'après un entraînement complet, le modèle ne produit pas de meilleurs résultats en utilisant cette métrique mais plutôt en utilisant la seconde métrique introduite, celle évaluant la conservation des longueurs d'arêtes. Cela indique que le modèle apprend de manière non supervisée d'autres caractéristiques contenues dans les données pendant le processus d'entraînement.

La figure 5.3 illustre la progression des métriques d'évaluation. La MPJPE commence à augmenter à l'époque 5, tandis que la MELV continue de diminuer. Ceci est attribué au fait que le réseau génère des mouvements qui ne reproduisent pas précisément la vérité terrain, mais qui conservent mieux la longueur des arêtes. Ce phénomène met en évidence l'ambiguïté de l'évaluation de la génération du mouvement qui va au-delà d'une simple comparaison entre les positions prédites et les positions réelles des articulations de squelettes. Le tableau 5.3 présente les valeurs

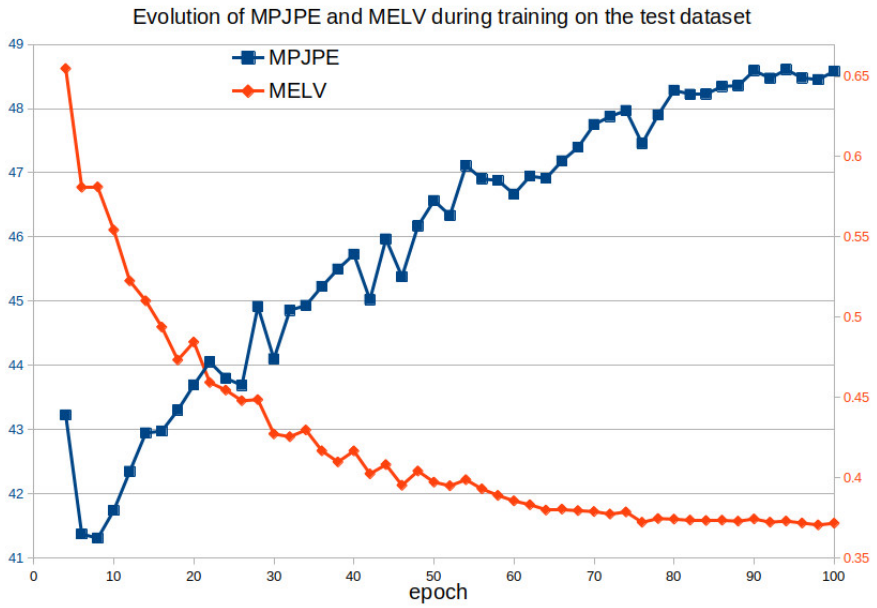


FIGURE 5.3 – Courbes d'évolution de la MPJPE moyenne et de la MELV moyenne sur tous les pas de temps lors de l'apprentissage sur l'ensemble de test. Nous pouvons voir que le modèle devient progressivement moins précis en termes de MPJPE mais devient meilleur pour préserver les longueurs d'arête.

Time (ms)	80	160	320	400	560	720	880	1000
Dataset	AMASS-BMLrub-OI (one identity)							
ground truth	0.13 ± 0.08	0.23 ± 0.13	0.38 ± 0.2	0.43 ± 0.22	0.49 ± 0.25	0.51 ± 0.26	0.49 ± 0.28	0.47 ± 0.28
SpecTrHuMS-OI-ES	0.4 ± 0.17	0.48 ± 0.21	0.63 ± 0.31	0.7 ± 0.35	0.81 ± 0.43	0.87 ± 0.47	0.89 ± 0.5	0.88 ± 0.52
SpecTrHuMS-OI	0.2 ± 0.09	0.29 ± 0.13	0.42 ± 0.2	0.47 ± 0.21	0.54 ± 0.23	0.57 ± 0.24	0.57 ± 0.26	0.57 ± 0.27
Dataset	AMASS-BMLrub-MI (multiple identities)							
ground truth	0.13 ± 0.08	0.24 ± 0.14	0.39 ± 0.21	0.44 ± 0.23	0.5 ± 0.26	0.52 ± 0.27	0.5 ± 0.29	0.48 ± 0.29
SpecTrHuMS-MI	0.37 ± 0.14	0.43 ± 0.15	0.55 ± 0.2	0.59 ± 0.21	0.65 ± 0.23	0.68 ± 0.24	0.68 ± 0.26	0.67 ± 0.27

TABLE 5.3 – En haut : comparaison de la MELV sur la base de données avec une identité entre la vérité terrain, un modèle arrêté prématurément (SpecTrHuMS-OI-ES) et un modèle entièrement entraîné (SpecTrHuMS-OI). Le modèle entièrement entraîné est capable de mieux préserver les longueurs d'arête que le modèle arrêté prématurément. En bas : comparaison de la MELV sur la base de données avec plusieurs identités entre la vérité terrain et un modèle entièrement entraîné (SpecTrHuMS-MI). La conservation de la longueur des arêtes entre les deux vérités terrain pour les deux ensembles de données a approximativement les mêmes valeurs, montrant que l'évaluation sur les deux ensembles de données est comparable. Le modèle entraîné sur plusieurs identités est capable de préserver correctement la longueur des arêtes par rapport à SpecTrHuMS-OI.

MELV et leurs écarts types correspondants pour le même modèle arrêté à l'époque donnant les meilleurs résultats de MPJPE et complètement entraîné par rapport aux valeurs de vérités terrain. Les résultats démontrent qu'avec un entraînement plus long, le modèle entièrement entraîné est capable de mieux préserver les longueurs d'arêtes que le modèle arrêté prématurément.

Nous illustrons maintenant comment notre modèle apprend correctement à préserver l'identité d'un maillage tout en le faisant bouger. Nous utilisons pour cela une base de données similaire à la précédente mais en introduisant la variation de l'identité (avec le suffixe MI pour *Multiple Identities*). À notre connaissance, il n'existe aucune recherche qui se concentre sur une tâche de prédiction et qui génère directement des surfaces, ce qui rend impossible la comparaison de la qualité de nos surfaces générées avec celles de travaux antérieurs. Cependant, nous montrons dans le tableau 5.3 que notre modèle, lorsqu'il utilise plusieurs identités, obtient des résultats similaires en terme de préservation des longueurs d'arête par rapport aux modèles qui n'emploient qu'une seule identité. La conservation des longueurs d'arête dans les deux bases de données est approximativement la même, comme l'indiquent les lignes de vérité terrain, ce qui signifie que les valeurs sont comparables (la ligne de vérité terrain dans le tableau 5.3 indique la MELV sur les données contenues dans la base d'entraînement). Plus précisément, on peut observer que l'évolution de la MELV ralentit significativement à 560ms à environ 0.5mm, indiquant que l'évaluation devrait converger vers cette valeur. Nous notons que la longueur moyenne des arêtes de la base de données avec plusieurs identités est d'environ 16 mm.

Résultats qualitatifs

Dans les figures 5.4 et 5.5, nous comparons des résultats visuels de prédictions sur l'ensemble de test avec une identité lors de l'utilisation d'un modèle arrêté tôt et donnant les meilleurs résultats en termes de MPJPE et d'un modèle entièrement entraîné. Les MELV sont indiquées dans les légendes de la figure pour les vérités terrain et les prédictions du modèle. En observant ces valeurs et les représentations visuelles sur les deux figures, il est évident que le modèle entraîné pendant une durée plus longue présente une capacité supérieure à maintenir les longueurs des bras.

Dans les figures 5.6 et 5.7, nous montrons des résultats visuels obtenus depuis le modèle utilisant l'ensemble de données à identités multiples (SpecTrHuMS-MI). Les deux figures montrent la capacité de notre modèle à préserver correctement l'identité. De même que pour le modèle entraîné en utilisant une seule identité, le mouvement généré n'est pas nécessairement proche de la vérité terrain mais reste réaliste. Néanmoins, le modèle est capable de comprendre comment déplacer les coefficients spectraux, et donc les sommets des maillages en entrée, tout en conservant l'apparence du sujet. Cela est une qualité cruciale pour de nombreuses applications du monde réel.

Pour résumer, le modèle présentant les meilleurs résultats est celui entièrement entraîné même si son score MPJPE est moins bon. Ceci est visuellement mis en

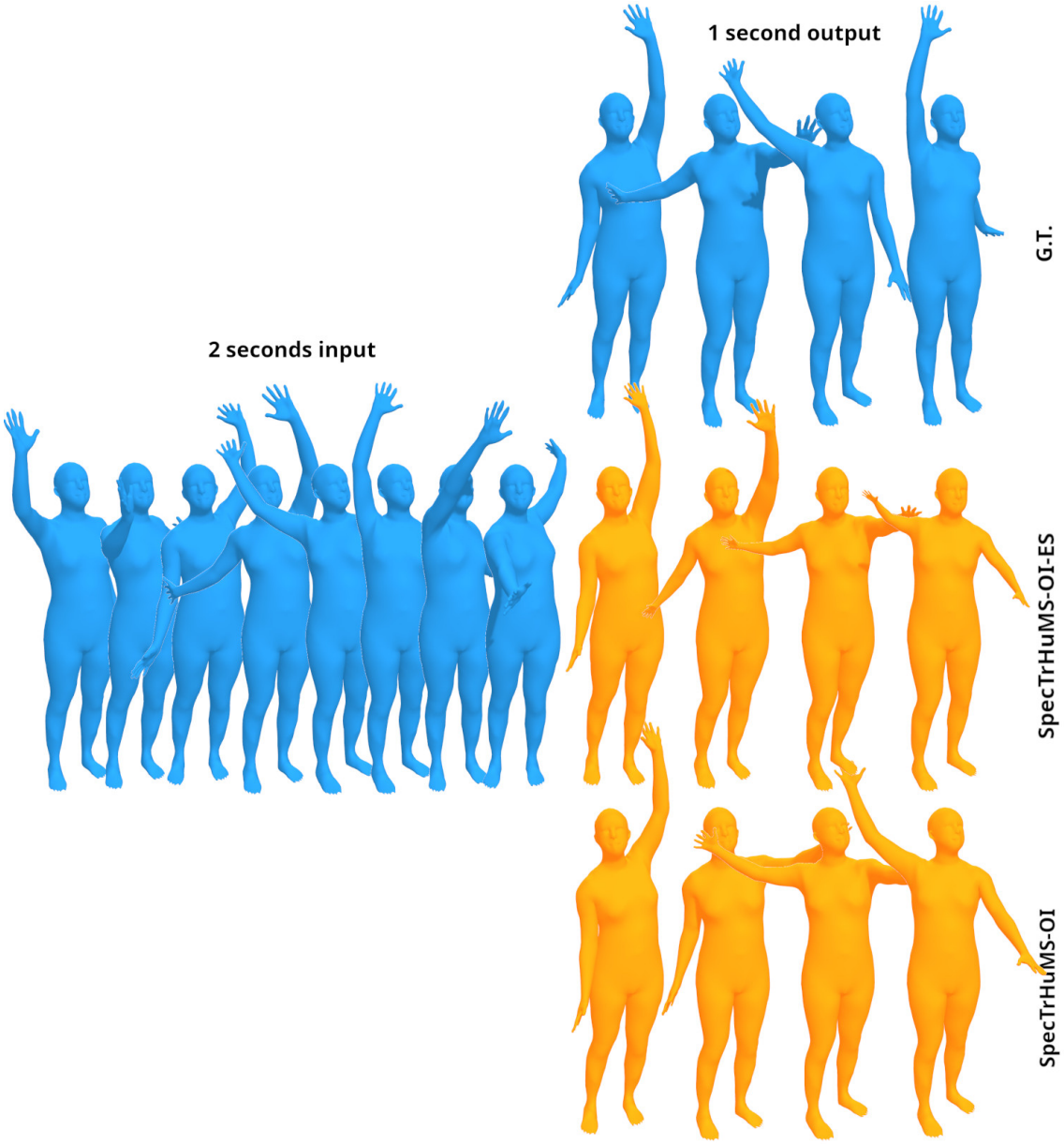


FIGURE 5.4 – Comparaison visuelle de notre modèle utilisant une identité dont l'entraînement a été arrêté prématurément (SpecTrHuMS-OI-ES) avec notre modèle entièrement entraîné (SpecTrHuMS-OI) sur la base de données de test. Alors que le modèle arrêté prématurément donne de meilleurs scores MPJPE, le modèle entièrement entraîné est capable de mieux préserver les longueurs des arêtes. Les maillages bleus représentent les vérités terrain avec une MELV de 0,72 mm. Les maillages oranges représentent la génération de nos modèles, avec une MELV de 2,03 mm pour le SpecTrHuMS-OI-ES et de 0,59 mm pour le SpecTrHuMS-OI. Toutes les MELV indiquées dans cette légende concernent le dernier pas de temps.

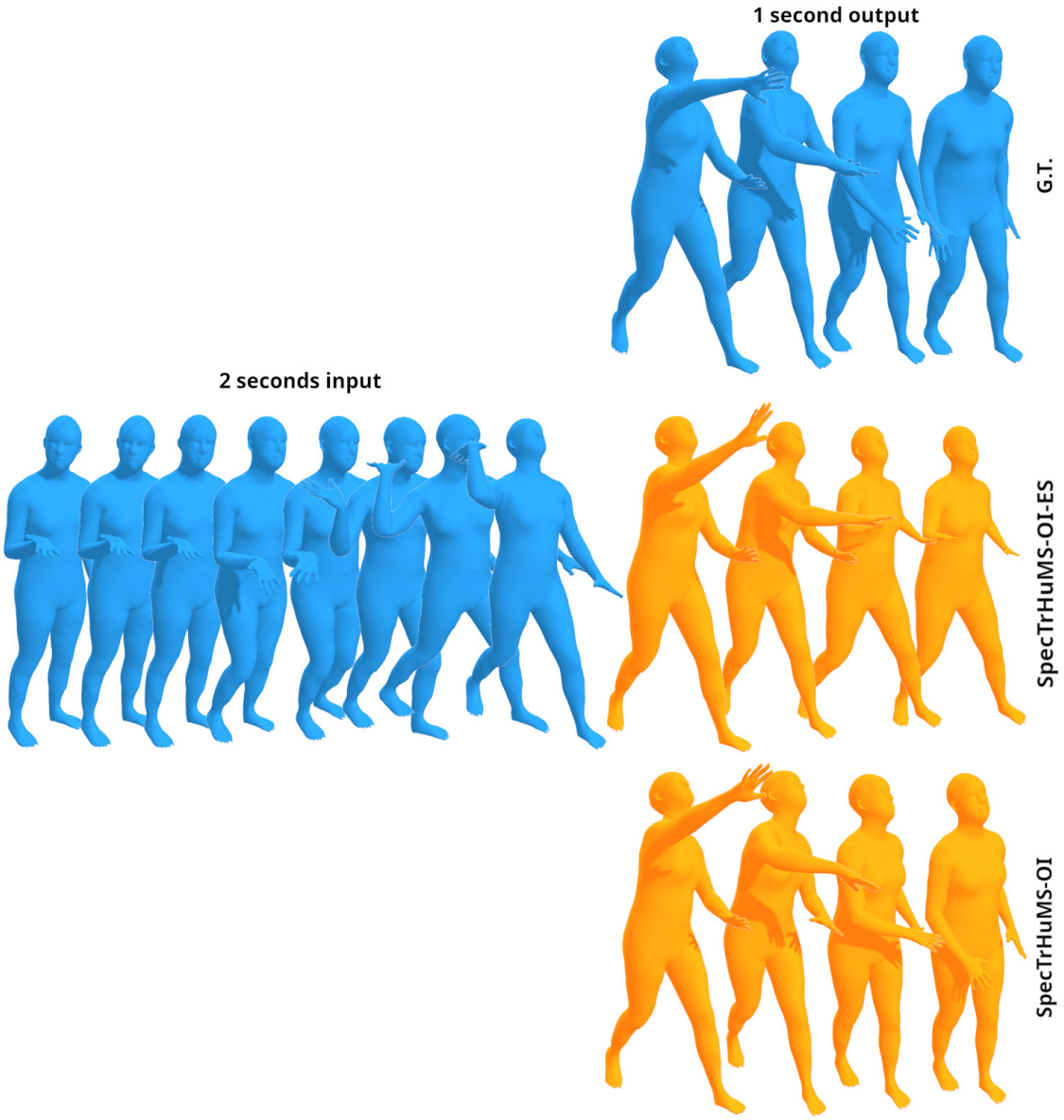


FIGURE 5.5 – Comparaison visuelle de notre modèle utilisant une identité dont l'entraînement a été arrêté prématurément (SpecTrHuMS-OI-ES) avec notre modèle entièrement entraîné (SpecTrHuMS-OI) sur la base de données de test. Alors que le modèle arrêté prématurément donne de meilleurs scores MPJPE, le modèle entièrement entraîné est capable de mieux préserver les longueurs des arêtes. Les maillages bleus représentent les vérités terrain avec une MELV de 0,96 mm. Les maillages oranges représentent la génération de nos modèles, avec une MELV de 2,31 pour le SpecTrHuMS-OI-ES et de 0,64 pour le SpecTrHuMS-OI. Toutes les MELV indiquées dans cette légende concernent le dernier pas de temps.

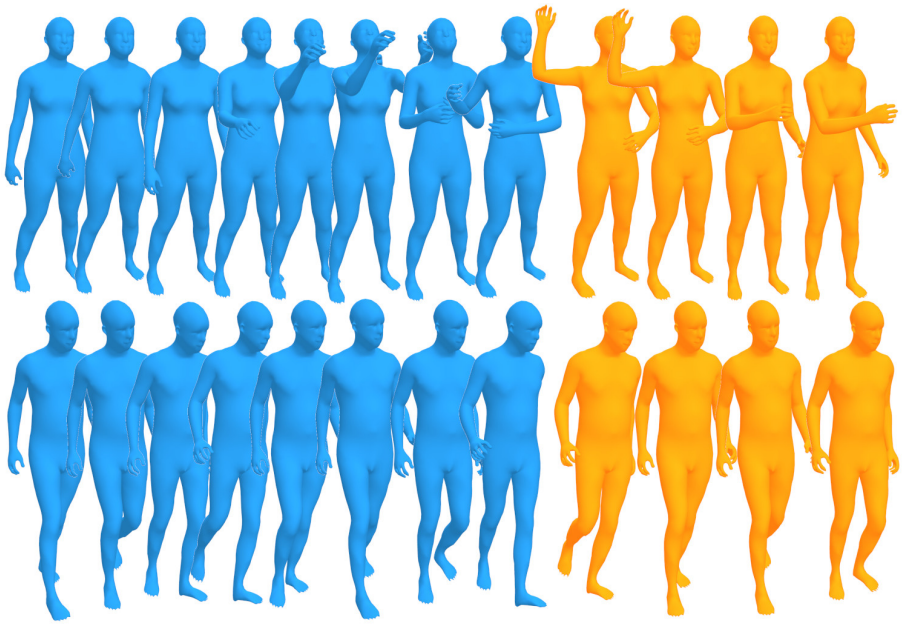


FIGURE 5.6 – Résultats visuels en utilisant un modèle entraîné sur une base de données à identités multiples (SpecTrHuMS-MI) : l'identité des formes est bien préservée tout en générant un mouvement réaliste.

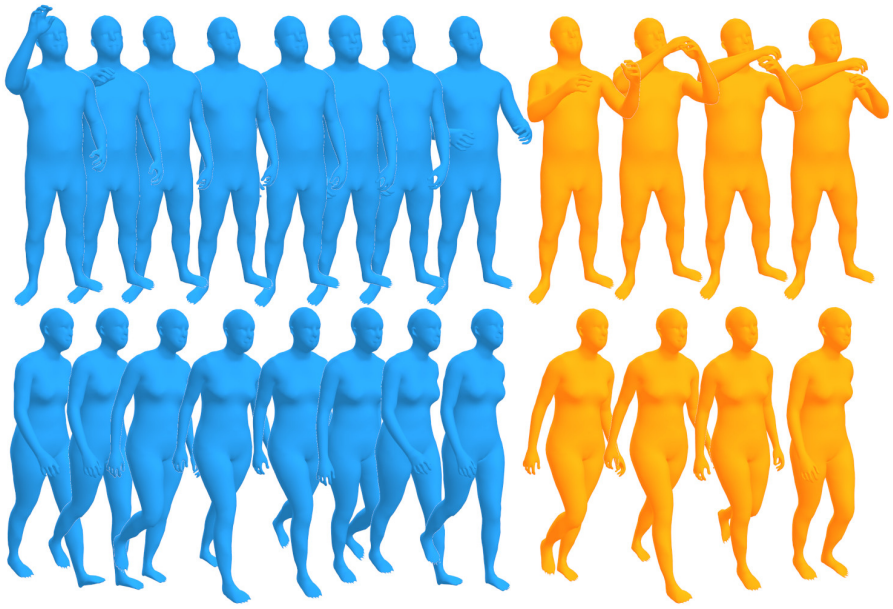


FIGURE 5.7 – Résultats visuels en utilisant un modèle entraîné sur une base de données à identités multiples (SpecTrHuMS-MI) : l'identité des formes est bien préservée tout en générant un mouvement réaliste.

évidence dans les figures 5.4 et 5.5 : lorsque les longueurs de bras ne sont pas conservées, la MPJPE est meilleure, ce qui signifie que cette métrique n'est pas adaptée à notre problématique. C'est pourquoi la métrique MELV est plus pertinente (voir figure 5.3). Il en résulte des prédictions qui peuvent s'écarter des vérités de terrain mais qui sont plus réalistes. À l'avenir, des mesures concernant le réalisme d'un mouvement devraient être introduites.

Nous présentons également des comparaisons visuelles avec siMLPe [57] dans les figures 5.8 et 5.9. Nous rappelons que leur architecture prend en entrée des articulations de squelettes alors que notre modèle prend en entrée des coefficients spectraux contenant l'information géométrique (nous obtenons les positions des articulations à partir des sommets du maillage en utilisant la matrice de régression disponible dans le processus SMPL). Dans la figure 5.8, pour le premier exemple, le mouvement prédit par siMLPe est plus proche de la vérité terrain, tandis que le mouvement prédit par notre modèle s'en écarte tout en restant réaliste, reflétant l'ambiguïté de l'évaluation MPJPE abordée précédemment. Dans les autres exemples, les deux travaux ont tendance à produire des mouvements moins dynamiques que les vérités terrain, mais notre méthode est capable de générer plus de mouvements que siMLPe (voir les figures 5.8 et 5.9 en bas). De plus, dans le cas de siMLPe, les squelettes générés peuvent avoir des longueurs de bras réduites (voir figure 5.9 à la deuxième ligne).

Une vidéo disponible en ligne (`https://youtu.be/sIw0vJCzfcg`) est fournie afin de mieux visualiser les résultats.

5.4.6 Autres applications

Nous montrons dans cette section des résultats supplémentaires pour d'autres applications, d'abord sur une tâche de prédiction à long terme qui ne nécessite pas un autre entraînement, puis sur une tâche de complétion (*in-betweening*) qui nécessite une autre méthode d'apprentissage, et enfin sur une tâche de prédiction lors de l'utilisation d'une base de données composée de surfaces qui ne peuvent pas être approximées par des squelettes.

Prédiction à long terme

Afin de prédire à plus long terme un mouvement donné, nous adoptons une méthode autorégressive. Cette approche consiste à donner un ensemble initial de 50 poses d'un mouvement au modèle et à prédire les 25 suivantes. Les poses prédites sont ensuite ajoutées à l'ensemble initial, et le processus est répété avec la dernière seconde du mouvement en entrée et les poses nouvellement prédites. En répétant ce processus de manière itérative, nous sommes en mesure de générer une séquence de poses qui extrapole le mouvement d'origine sur une longue période de temps. La figure 5.10 montre des exemples de cette application. La première ligne montre la prédiction à long terme d'un mouvement de bras. Sur la deuxième ligne, une

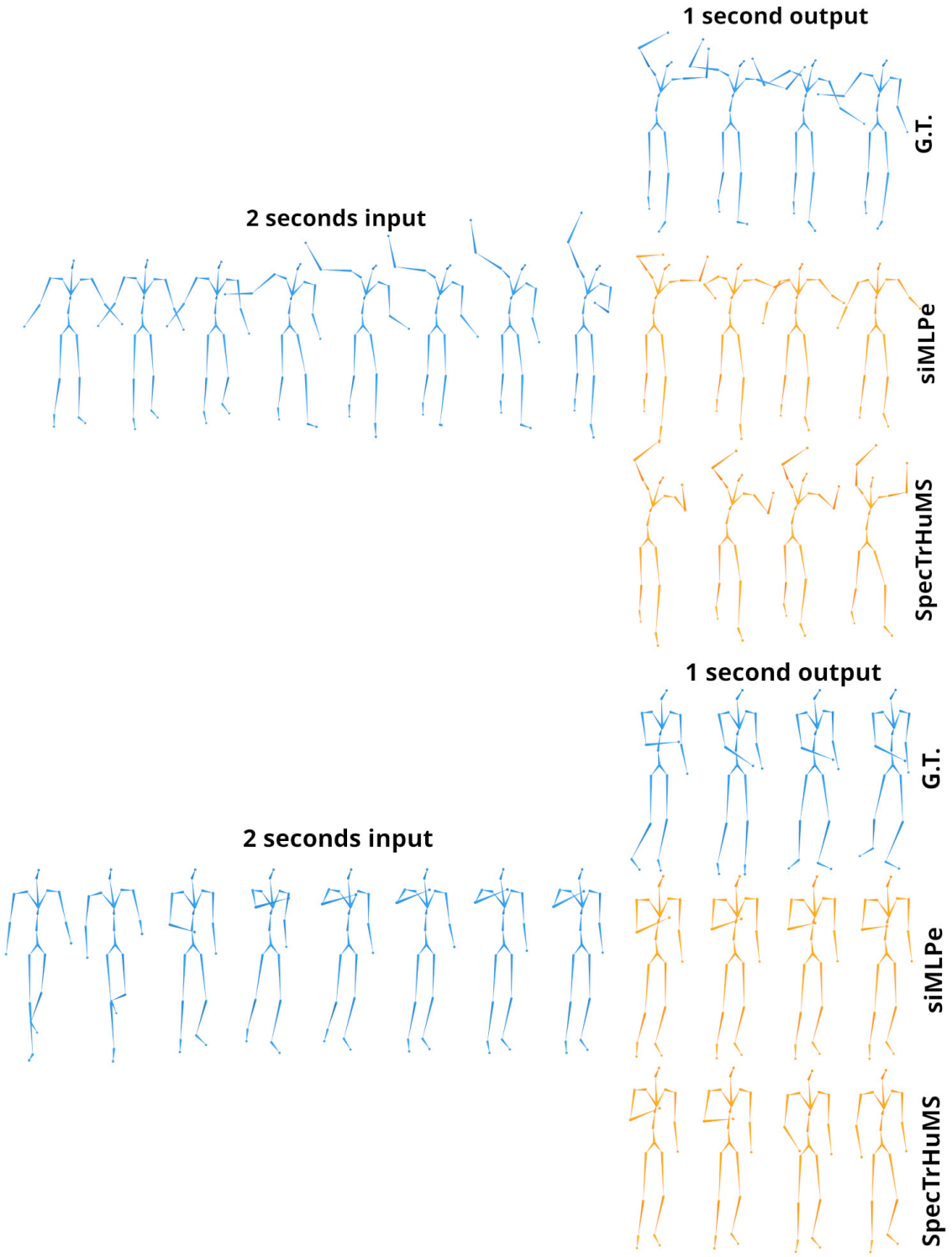


FIGURE 5.8 – Comparaison visuelle avec siMLPe [57]. Dans notre cas, nous obtenons les positions des articulations à partir des sommets du maillage en utilisant la matrice de régression disponible dans le processus SMPL, tandis que siMLPe fonctionne directement avec les jointures des squelettes.

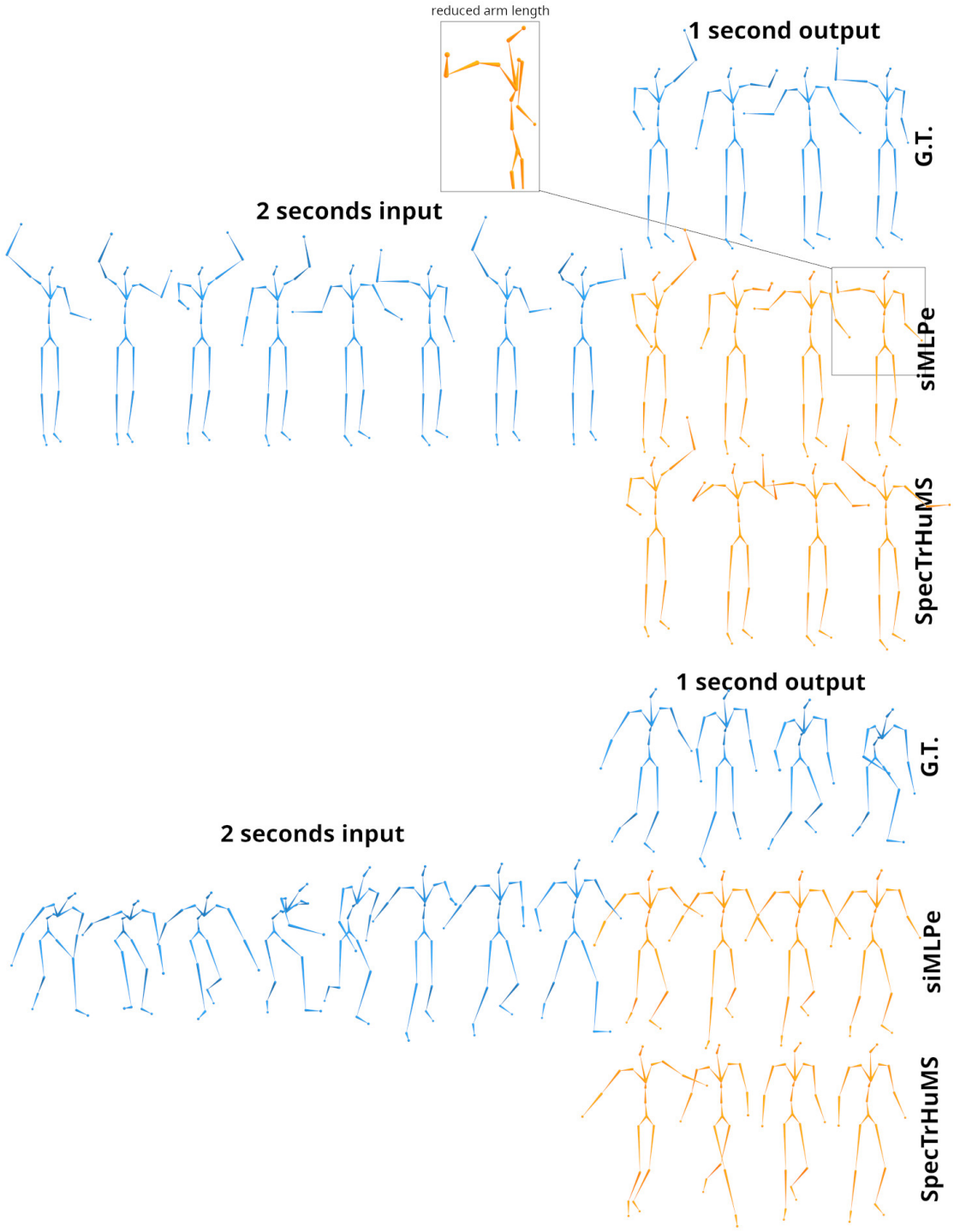


FIGURE 5.9 – Comparaison visuelle avec siMLPe [57]. Dans notre cas, nous obtenons les positions des articulations à partir des sommets du maillage en utilisant la matrice de régression disponible dans le processus SMPL, tandis que siMLPe fonctionne directement avec les jointures des squelettes.

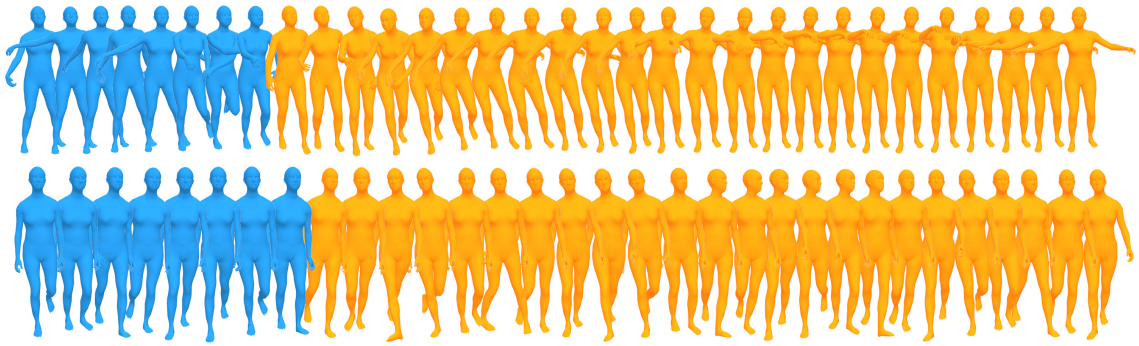


FIGURE 5.10 – Exemples d'extrapolation. Les maillages bleus sont donnés en entrée au modèle et les maillages oranges sont générés de manière autorégressive : 2 secondes sont données en entrée, 1 seconde est générée, et les secondes suivantes sont générées en donnant en entrée les générations précédentes. Le modèle utilisé est SHTMS-MI, et un total de 6 secondes sont produites. Le modèle est capable d'extrapoler un mouvement tout en conservant l'information d'identité et de mouvement.

animation de marche est générée, similaire à la séquence de base. Le modèle est capable de générer correctement un mouvement tout en conservant les informations de mouvement et d'identité. Il est important de noter que la génération de poses supplémentaires avec cette méthode autorégressive peut être réalisée en temps réel.

Complétion (*in-betweening*)

Parce que notre approche utilise une architecture transformeur, elle peut être facilement appliquée à d'autres tâches telles que la complétion. Dans cette application supplémentaire, qui nécessite un nouvel apprentissage, le modèle apprend à prédire le mouvement entre deux poses connues, ce qui nécessite de donner deux secondes de mouvement et une dernière pose supplémentaire. L'adaptation de notre approche à la complétion nécessite uniquement l'ajout d'une pose supplémentaire aux données en entrée et le remplacement des variables latentes inconnues par une interpolation entre la dernière pose des données en entrée et la pose supplémentaire (voir figure 5.2). La figure 5.11 montre un exemple de cette application. Le modèle est capable d'interpoler correctement entre deux poses tout en tenant compte du mouvement précédent et en préservant les longueurs d'arête.

Application à d'autres bases de données

Afin de prouver la généralisabilité de notre méthode, nous présentons des expériences sur la prédiction de simulations d'un morceau de tissu. A l'aide de Blender [34], nous générons un maillage triangulaire représentant un morceau de tissu (voir figure 5.12) composé de 484 sommets, tout en lui attribuant un modificateur permettant la simulation physique et réaliste de la matière. À l'aide de ce maillage, 20 000 simulations de deux secondes sont créées dans lesquelles deux sommets aléatoires différents sont épinglés et dans lesquelles la surface subit la gravité et des

FIGURE 5.11 – Exemples de complétion. Le modèle est capable d'interpoler correctement entre deux poses tout en tenant compte du mouvement précédent.

auto-collisions (16 000 sont utilisées pour l'entraînement et 4 000 pour les tests). Le maillage est géométriquement asymétrique de sorte que les simulations générées soient toujours différentes. Le Laplacien topologique avec ses vecteurs propres sont ensuite calculés à partir de la connectivité, permettant de créer des séquences de coefficients spectraux qui sont données en entrée à notre modèle de la même manière que dans les figures 5.1 et 5.2 dans le cas de la prédiction, mais avec une seconde connue et une seconde de prédiction.

Dans le tableau 5.4, des évaluations quantitatives présentent les racines de l'erreur quadratique moyenne (*root-mean-square error* ou RMSE) entre les vérités terrain et les surfaces générées à chaque pas de temps sur l'ensemble de données de

Time (ms)	80	160	320	400	560	720	880	1000
Dataset	AMASS-BMLrub-OI							
SpecTrHuMS-OI	0.00590	0.01123	0.02166	0.02575	0.03111	0.03476	0.03782	0.03959
Dataset	Cloth dataset							
SpecTrHuMS-cloth	0.00625	0.00672	0.00769	0.00819	0.00935	0.01069	0.01228	0.01369

TABLE 5.4 – Comparaison des racines de l'erreur quadratique moyenne (*root-mean-square error* ou RMSE) sur les sommets des maillages pour les modèles entraînés sur l'ensemble représentant des humains et utilisant une seule identité et sur l'ensemble de données de tissu. Les RMSEs sont normalisées par la plus grande boîte englobante de chaque maillage afin que les valeurs soient comparables. Les valeurs sont plus faibles pour l'ensemble de données de tissu, ce qui montre que notre modèle est généralisable.

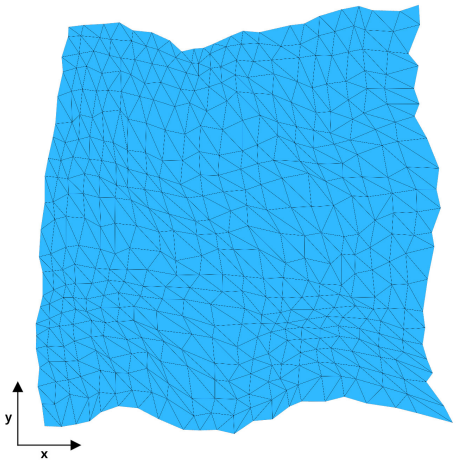


FIGURE 5.12 – Le maillage triangulaire utilisé pour créer une base de données de simulations d'une pièce de tissu. Il est plat (uniquement sur les axes x et y) et composé de 484 sommets. Pour chaque animation, deux sommets aléatoires sont épinglés et le tissu subit la gravité et des auto-collisions. Le maillage est géométriquement asymétrique de sorte que les simulations soient toujours différentes.

test. Les scores sont exprimés en pourcentage de la boîte englobante du maillage correspondant. Les valeurs correspondant à la base de données de tissu sont comparables à celles de l'ensemble représentant des humains, montrant que notre modèle n'a pas à s'appuyer sur des paramétrisations telles que SMPL et est généralisable à des surfaces dynamiques qui ne peuvent pas être approximées avec des squelettes.

Des exemples visuels sont également présentés dans la figure 5.13. Nous pouvons voir que le modèle parvient à prédire des mouvements réalistes d'une simulation de tissu compte tenu du début de la séquence. Les exemples sont présentés sous forme d'images donc les résultats sont difficilement visualisables, mais les auto-collisions sont bien reproduites dans les animations générées. De plus, les plis créés ne sont pas exactement identiques à ceux des vérités terrain, ce qui montre que le modèle n'a pas complètement appris l'ensemble de données mais est plutôt capable de générer un comportement réaliste du tissu sous la gravité. Une vidéo disponible en ligne (`https://youtu.be/sIw0vJCzfcg`) est fournie afin de mieux visualiser les résultats.

5.5 Discussion

Nous avons montré que notre modèle est capable de générer un mouvement humain à court et à long terme en fonction des poses données en entrée, de compléter une partie manquante dans un mouvement, et qu'il est généralisable à d'autres types de surfaces sans avoir à s'appuyer sur des paramétrisations. Néanmoins, l'architecture proposée présente certaines limitations. Premièrement, puisque nous utilisons le Laplacien topologique, seuls les jeux de données constitués de maillages avec une connectivité constante peuvent être fournis en entrée. Dans le cas d'un ensemble de

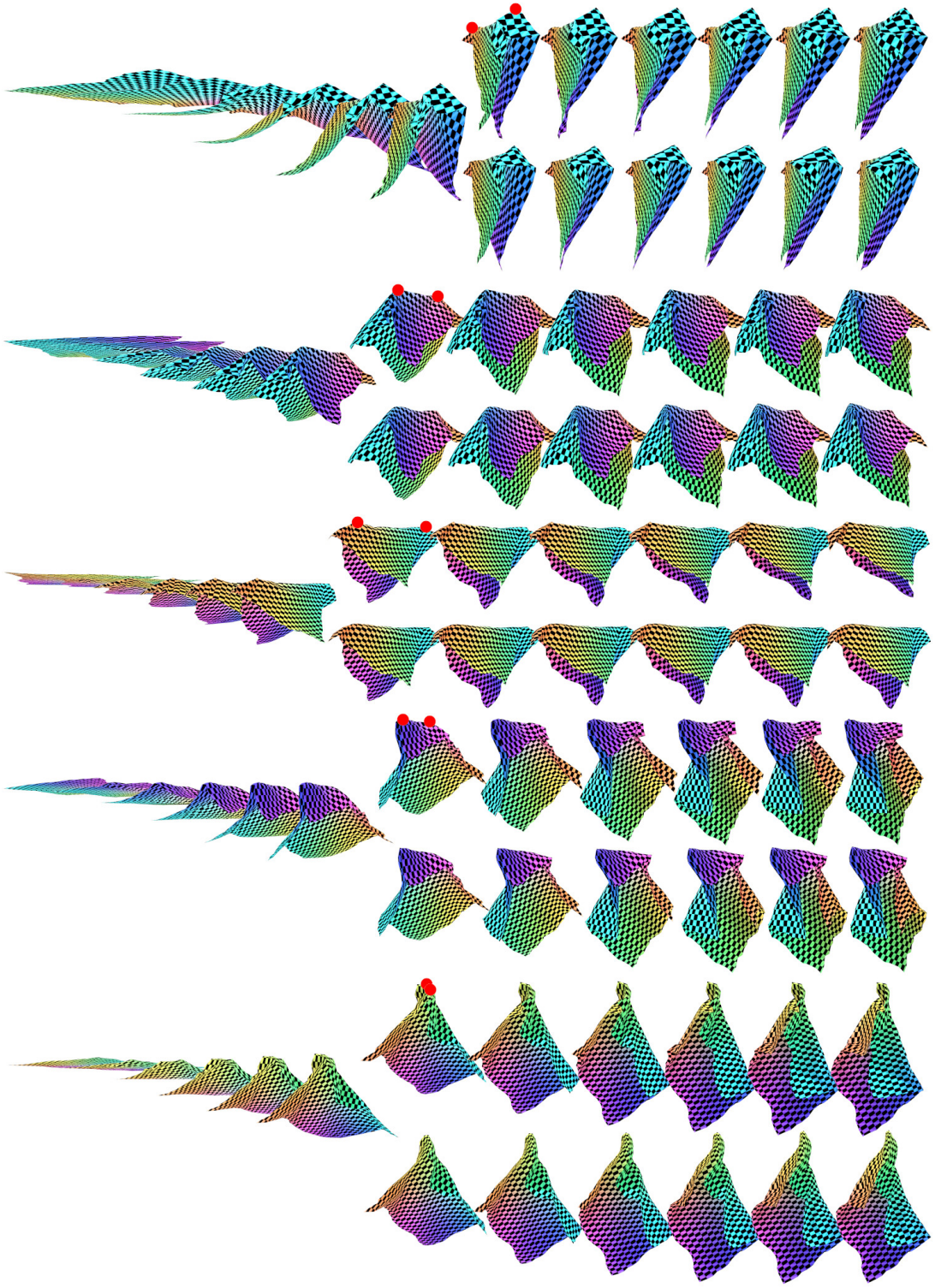


FIGURE 5.13 – Exemples visuels de prédictions de simulations de tissu. Pour chaque ligne, une seconde est donnée en entrée (à gauche), en haut se trouvent les vérités terrain et en bas les maillages générés. Pour chaque animation, deux sommets aléatoires sont épinglés (marqués avec des points rouges) et le tissu subit la gravité et des auto-collisions. Notre modèle est capable de reproduire des mouvements, des auto-collisions et des plis réalistes.

données avec un nombre de sommets et une connectivité variables, nous pourrions prétraiter et remailler tous les échantillons avec une connectivité commune afin qu'ils soient compatibles avec notre processus. De plus, il est possible d'envisager dans le futur de pouvoir synchroniser des bases spectrales calculées depuis différentes triangulations en utilisant des cartes fonctionnelles (voir section 3.4). Deuxièmement, nous n'avons utilisé que 512 fréquences sur les 6 890 disponibles depuis la discrétisation SMPL. Cela conduit à des maillages filtrés qui ne montrent pas les détails des hautes fréquences, en particulier sur le visage, les mains et les pieds. C'est une limitation, mais aussi un avantage car lors de l'analyse du mouvement d'un corps humain, les détails contenus dans les hautes fréquences ne sont pas essentiels, et notre méthode permet un contrôle sur la quantité d'information à laquelle le réseau a accès. Donner des hautes fréquences à notre modèle est simple, tandis que des méthodes telles que SMPL nécessitent une mise à niveau pour augmenter la qualité des maillages [106]. À l'avenir, cet aspect pourrait être amélioré soit en travaillant avec plus de fréquences, soit en introduisant un réseau de neurones supplémentaire dont la tâche est de compléter les détails contenus dans les hautes fréquences en fonction de l'information disponible dans les basses.

5.6 Conclusion

Dans ce chapitre, nous avons présenté SpecTrHuMS, un transformeur spectral qui peut traiter efficacement des séquences de maillages triangulaires 3D. Notre modèle est capable de capturer à la fois les dépendances spatiales et temporelles des formes en travaillant directement avec une représentation compressée de la géométrie. Contrairement à la plupart des travaux antérieurs dans ce domaine, notre modèle ne repose pas uniquement sur des articulations de squelettes et est capable de conserver l'information d'identité des formes. De plus, notre modèle est capable de contrôler le nombre de fréquences données en entrée et donc la quantité d'information à laquelle il a accès, et est généralisable à d'autres bases de maillages qui ne peuvent pas être modélisés à l'aide de squelettes.

Nous avons évalué notre réseau sur une tâche de prédiction principale sur AMASS, une base de données de séquences de surfaces humaines. De plus, grâce à l'architecture utilisée, nous avons proposé des applications supplémentaires qui peuvent être facilement implémentées. Nos expériences montrent que notre modèle est capable de générer correctement une séquence en préservant la longueur des arêtes, produisant des mouvements réalistes tout en préservant l'identité d'un sujet. Ces résultats démontrent l'efficacité de notre approche pour capturer à la fois les attributs physiques d'un objet et les variations des caractéristiques de maillages au fil d'une séquence. Cela suggère que ce travail représente une avancée dans la représentation efficace de séquences de maillages triangulaires et ouvre de nouvelles possibilités d'applications dans les domaines de l'animation, de la réalité virtuelle et de l'infographie. Dans l'ensemble, notre approche consistant à combiner l'information spectrale avec un autoencodeur convolutif et un transformeur fournit une direction potentiellement intéressante pour des futurs travaux dans le domaine de l'analyse et du traitement de

formes 4D. À l'avenir, nos efforts seront dirigés vers des entraînements sur des bases de données plus étendues et vers la modification du processus pour permettre aux utilisateurs d'influencer la génération. Cela comprendra la modification du nombre de coefficients spectraux utilisés pour générer des surfaces avec différents degrés de détail, ainsi que la possibilité de saisir un texte spécifique pour générer des actions ciblées.

Chapitre 6

Conclusion et perspectives

6.1 Conclusion

Ce travail s'inscrit dans le projet national ANR Human4D qui souhaite relever de nouveaux défis en proposant une nouvelle modélisation 4D efficiente du corps humain en mouvement et en s'intéressant à la problématique de la génération de séquences de données. Les plateformes permettant la capture de formes dynamiques deviennent de plus en plus accessibles et vont permettre un accès facilité à des bases de données composées de surfaces évoluant dans le temps et contenant l'information détaillée de l'apparence et du mouvement. Le coût de ces acquisitions reste tout de même non négligeable en terme de sujets sollicités pour effectuer la capture ou encore de matériel invoqué pour la numérisation et le nettoyage des données. Plus particulièrement, nous nous sommes intéressés à la création réaliste de nouvelles données en s'appuyant sur l'utilisation de techniques d'analyse spectrale appliquées à des maillages triangulaires et de techniques d'apprentissage profond génératif. Nous avons abordé les différents sujets qui concernent le développement d'un tel processus et introduit deux contributions qui permettent de créer une représentation adaptée au corps humain puis de comprendre et générer des données incluant une dimension temporelle.

Le chapitre 1 introduit le contexte dans lequel se sont déroulés les travaux en présentant les principales méthodes d'apprentissage profond dont s'inspirent les contributions de ce document, en quoi les appliquer aux données traitées est encore aujourd'hui un défi et enfin en quoi traiter la dynamique représente un défi supplémentaire.

Le chapitre 2 présente le formalisme SMPL utilisé durant la thèse pour générer des données d'entraînement ainsi que des maillages issus de la plateforme KINOVIS représentant des objets auxquels notre méthode peut être appliquée dans de futurs travaux. Aussi, les outils permettant l'entraînement et la visualisation des structures générées sont présentés.

Le chapitre 3 aborde la notion d'analyse spectrale appliquée à des maillages triangulaires. La transformée de Fourier permet la mise en œuvre de nombreuses techniques de traitement de signaux et d'images et trouve de nombreuses applications

dans des domaines tels que la reconnaissance vocale, les transmissions numériques ou encore le milieu biomédical. Son lien avec l'opérateur Laplacien est d'abord montré, ce qui permet d'introduire la transformée dans le domaine des fréquences de signaux définis sur des surfaces. Ce document met l'accent sur la projection des signaux définissant la géométrie des objets étudiés, permettant d'obtenir les coefficients spectraux qui en sont une version compactée et ordonnée. L'analyse spectrale de données 3D a largement été utilisée dans la littérature et est aujourd'hui combinée à des réseaux de neurones en exploitant l'information provenant des valeurs propres, des vecteurs propres ou de descripteurs découlant de l'opérateur utilisé. Cependant, les travaux de la littérature ne tirent pas d'avantages directs de ces coefficients spectraux et reposent toujours sur le domaine spatial dans lequel les contraintes sont toujours présentes. Ce travail permet de montrer qu'il est possible de développer des applications d'apprentissage profond génératif en ne se reposant que sur le domaine spectral.

Le chapitre 4 s'intéresse premièrement à un traitement statique de l'information et représente la première contribution. En utilisant une architecture autoencodeur simple prenant en entrée les coefficients spectraux correspondant à des poses, il est possible d'obtenir une représentation simplifiée et adaptée au corps humain. Le fait de donner l'information contenue dans ces coefficients à un réseau de neurones permet de s'affranchir de plusieurs problématiques telles que la non-régularité des sommets dans le domaine spatial ou encore leur grand nombre. En comparaison aux travaux de l'état de l'art, notre méthode permet d'obtenir des résultats compétitifs en terme de reconstruction, de génération et surtout en terme de temps de calcul, permettant de s'entraîner sur des bases de données denses sans temps d'entraînement trop long en raison de la capacité de contrôle sur la quantité d'information (le nombre de fréquences) donnée au réseau. Ce traitement statique permet une compréhension pseudo-dynamique des données en entrée : une interpolation dans l'espace latent construit par le réseau entre deux poses génère un mouvement réaliste. Cependant, le corps humain est capable de se transformer de manière presque isométrique et complexe, et la variété de mouvements possibles entre deux poses ne peut pas être correctement représentée en utilisant de simples interpolations, ce qui implique d'utiliser une architecture qui comprenne le contexte d'une séquence de données. Aussi, l'information d'identité et de pose n'est pas séparée dans l'espace initial des données, et une tâche importante est de développer un algorithme capable de séparer cette information en utilisant les annotations disponibles et sans ajouter de règles géométriques à respecter qui compliqueraient le processus d'apprentissage.

Le chapitre 5 s'intéresse ensuite aux deux problématiques induites par un simple traitement statique. Comprendre le contexte d'une séquence temporelle de données à l'aide de réseaux de neurones est une tâche largement étudiée dans la littérature. L'architecture transformeur récemment introduite permet de traiter ce genre de données efficacement en créant des liens entre des échantillons d'une séquence même s'ils sont temporellement éloignés. Néanmoins, l'entraînement de ce genre d'architectures est déjà coûteux en temps de calcul sur des données à une dimension rendant leur application directe à des données 3D impossible. La représentation créée par le

réseau introduit dans le chapitre précédent permet de s'affranchir de cette problématique. L'espace latent construit par le réseau de la première contribution est une représentation encore plus compacte que le domaine fréquentiel de surfaces et, en raison de sa faible dimension, peut être donné en entrée à une architecture transformeur. Ce processus représente la seconde contribution de la thèse dont le principal apport est, en comparaison à l'état de l'art concernant la prédiction, l'inclusion de l'information surfacique dans un traitement dynamique de corps humain en mouvement. Aussi, notre processus peut être employé pour traiter des surfaces qui ne sont pas approximables par des squelettes. Le modèle est principalement évalué sur une tâche de prédiction de mouvement, mais la structure de l'architecture utilisée le rend facilement applicable à d'autres tâches comme de la génération à long terme. Le fait que le modèle réussisse à générer des mouvements réalistes tout en conservant l'identité du sujet montre enfin que la séparation de l'information d'identité et de pose est partiellement réalisée de manière semi-supervisée (en utilisant l'annotation d'identité) et sans ajouter de propriétés géométriques à respecter.

Les travaux présentés contribuent au domaine de l'informatique graphique en permettant de générer des séquences de surfaces représentant des corps humains en mouvement. La méthode proposée permet de résoudre des problèmes liés au domaine d'étude de l'apprentissage profond géométrique (*Geometric Deep Learning*) récemment introduit et présente des avantages qui introduisent une nouvelle manière de traiter des données 3D et 4D. Mais la manière de traiter l'information présente aussi des limitations, ouvrant des perspectives pour de futurs travaux.

6.2 Perspectives

La littérature concernant le développement d'architectures d'apprentissage profond est en expansion constante, avec de nombreuses contributions qui enrichissent notre compréhension générale de ce domaine. Les méthodes d'apprentissage et architectures exploitées ici sont relativement simples, et une première perspective est d'envisager l'utilisation de techniques plus poussées qui pourraient améliorer les résultats obtenus pour le traitement statique ou dynamique de l'information. Par exemple, l'autoencodeur pourrait être remplacé par un GAN, un VAE, un transformeur ou une de leurs variantes (voir section 1.1.1). Cela permettrait éventuellement de créer un espace latent de meilleure qualité, proposant une meilleure génération de poses possibles et améliorant les résultats obtenus avec le processus dynamique qui utilise ce dernier. Aussi, le transformeur utilisé pour les séquences pourrait être remplacé par une version créant lui aussi un espace latent dans lequel des interpolations entre animations seraient possibles. Le modèle de diffusion, largement utilisé ces dernières années dans la littérature, pourrait être utilisé afin de donner une réelle possibilité de générer du contenu. L'ajout de conditionnement pourrait enfin apporter un contrôle de l'utilisateur sur le processus de génération.

La base de données AMASS utilise le formalisme SMPL, ce qui génère des maillages ayant un faible nombre de sommets et possédant peu de détails concernant

la surface. L'accès à une base de données possédant plus de détails sur les surfaces (morphologiques ou textiles) et assez dense pour un apprentissage pourrait donner l'occasion de mettre plus en avant l'utilisation de la compaction de l'énergie du domaine spectral et le contrôle offert par notre méthode sur la quantité d'information donnée. Utiliser des objets comprenant plus de détails et une géométrie plus précise implique de traiter plus d'information. D'autres travaux pourraient concerner l'apport de plus de fréquences aux modèles afin d'augmenter la quantité d'information fournie aux réseaux sans détériorer la qualité des résultats et sans temps de calcul plus longs.

Une autre caractéristique importante du formalisme SMPL est la connectivité commune entre les échantillons, permettant la projection des géométries de tous les maillages sur une base commune obtenue depuis le Laplacien topologique. Dans le cas d'une base de données composée de plusieurs connectivités différentes, notre méthode n'est plus applicable. Une solution simple est de remailler tous les échantillons afin qu'ils aient la même discrétisation, permettant l'application directe de notre méthode à cette dernière. Autrement, une méthode consisterait à synchroniser les domaines spectraux des différentes connectivités. Pour l'instant, le calcul des cartes fonctionnelles (*Functional Maps*) ne permet pas de trouver facilement des synchronisations entre différents domaines spectraux, mais de futurs travaux concernant ce processus pourraient améliorer leur qualité. Cette méthode nécessite de calculer un opérateur pour chaque échantillon, impliquant un temps de calcul beaucoup plus long. Des recherches pour optimiser le calcul d'un opérateur seraient donc aussi nécessaires.

Notre méthode n'utilise que les coefficients spectraux en entrée du modèle. Il serait envisageable de donner plus d'information telle que des descripteurs calculés depuis les valeurs ou vecteurs propres. Cela permettrait au réseau de générer des surfaces qui respectent certaines contraintes géométriques telles que des distances géodésiques ou encore des quantités de diffusion de l'information calculées sur les surfaces.

Finalement, de manière plus générale, notre méthode pourrait être utilisée pour étudier tout type d'objets dynamiques : animaux, protéines, réseaux sociaux et financiers ou même des structures cosmologiques. L'analyse temporelle de surfaces dynamiques n'est pas seulement applicable à la compréhension du mouvement du corps humain, et les travaux réalisés dans cette thèse sont en lien avec un domaine d'étude bien plus large et précurseur qui prendra probablement de plus en plus de place dans le paysage de l'apprentissage profond génératif ces prochaines années : l'analyse et la création de données 3D/4D assistée par intelligence artificielle.

Bibliographie

[1] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Bjorn Ottersten. A survey on Deep Learning Advances on Different 3D Data Representations. 2018. Publisher : arXiv Version Number : 2.

[2] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A Spatio-temporal Transformer for 3D Human Motion Prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574, London, United Kingdom, December 2021. IEEE.

[3] Muhammad Shamsul Alam, Farhan Bin Mohamed, Ali Selamat, and Akm Bellal Hossain. A Review of Recurrent Neural Network Based Camera Localization for Indoor Environments. *IEEE Access*, 11 :43985–44009, 2023.

[4] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138 :14–32, June 2021.

[5] Andrea Asperti, Davide Evangelista, and Elena Loli Piccolomini. A Survey on Variational Autoencoders from a Green AI Perspective. *SN Computer Science*, 2(4) :301, July 2021.

[6] James Atwood and Don Towsley. Diffusion-Convolutional Neural Networks, July 2016. arXiv :1511.02136 [cs].

[7] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature : A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1626–1633, Barcelona, Spain, November 2011. IEEE.

[8] Tristan Aumentado-Armstrong, Stavros Tsogkas, Sven Dickinson, and Allan Jepson. Disentangling Geometric Deformation Spaces in Generative Latent Shape Models. *International Journal of Computer Vision*, 131(7) :1611–1641, July 2023.

[9] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric Disentanglement for Generative Latent Shape Models. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8180–8189, Seoul, Korea (South), October 2019. IEEE.

[10] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. In Lior Rokach, Oded Maimon, and Erez Shmueli, editors, *Machine Learning for Data Science Handbook*, pages 353–374. Springer International Publishing, Cham, 2023.

[11] Maysam Behmanesh, Maximilian Krahn, and Maks Ovsjanikov. TIDE : Time Derivative Diffusion for Deep Learning on Graphs, June 2023. arXiv :2212.02483 [cs].

[12] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral Clustering with Graph Neural Networks for Graph Pooling, December 2020. arXiv :1907.00481 [cs, stat].

[13] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST : Dataset and Evaluation for 3D Mesh Registration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, Columbus, OH, USA, June 2014. IEEE.

[14] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST : Registering Human Bodies in Motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5573–5582, Honolulu, HI, July 2017. IEEE.

[15] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[16] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Stefanos Zafeiriou, and Michael Bronstein. Neural 3D Morphable Models : Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7212–7221, Seoul, Korea (South), October 2019. IEEE.

[17] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric Deep Learning : Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4) :18–42, July 2017.

[18] Michael M. Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711, San Francisco, CA, USA, June 2010. IEEE.

[19] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. 2020. Publisher : arXiv Version Number : 4.

[20] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs, May 2014. arXiv :1312.6203 [cs].

[21] Judith Butepage, Michael J. Black, Danica Kragic, and Hedvig Kjellstrom. Deep Representation Learning for Human Motion Prediction and Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1591–1599, Honolulu, HI, July 2017. IEEE.

[22] Judith Butepage, Hedvig Kjellstrom, and Danica Kragic. Anticipating Many Futures : Online Human Motion Prediction and Generation for Human-Robot Interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4563–4570, Brisbane, QLD, May 2018. IEEE.

[23] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu, and Nadia Magnenat Thalmann. Learning Progressive Joint Propagation for Human Motion Prediction. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12352, pages 226–242. Springer International Publishing, Cham, 2020. Series Title : Lecture Notes in Computer Science.

[24] Wenming Cao, Canta Zheng, Zhiyue Yan, and Weixin Xie. Geometric deep learning : progress, applications and challenges. *Science China Information Sciences*, 65(2) :126101, February 2022.

[25] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. A Comprehensive Survey of AI-Generated Content (AIGC) : A History of Generative AI from GAN to ChatGPT, March 2023. arXiv :2303.04226 [cs].

[26] Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami Flow and Neural Diffusion on Graphs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1594–1609. Curran Associates, Inc., 2021.

[27] Benjamin Paul Chamberlain, James Rowbottom, Maria Gorinova, Stefan Webb, Emanuele Rossi, and Michael M. Bronstein. GRAND : Graph Neural Diffusion, September 2021. arXiv :2106.10934 [cs, stat].

[28] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, Honolulu, HI, July 2017. IEEE.

[29] Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu. A Survey on Graph Neural Networks and Graph Transformers in Computer Vision : A Task-Oriented Perspective, October 2022. arXiv :2209.13232 [cs].

[30] Zhixiang Chen and Tae-Kyun Kim. Learning Feature Aggregation for Deep 3D Morphable Models, May 2021. arXiv :2105.02173 [cs].

[31] Hsu-Kuang Chiu, Ehsan Adeli, Borui Wang, De-An Huang, and Juan Carlos Niebles. Action-Agnostic Human Pose Forecasting. In *2019 IEEE Winter*

Conference on Applications of Computer Vision (WACV), pages 1423–1432, Waikoloa Village, HI, USA, January 2019. IEEE.

[32] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics.

[33] Sakib H. Chowdhury, Md. Robius Sany, Md. Hafiz Ahamed, Sajal K. Das, Faisal Rahman Badal, Prangon Das, Zinat Tasneem, Md. Mehedi Hasan, Md. Robiul Islam, Md. Firoj Ali, Sarafat Hussain Abhi, Md. Manirul Islam, and Subrata Kumar Sarker. A State-of-the-Art Computer Vision Adopting Non-Euclidean Deep-Learning Models. *International Journal of Intelligent Systems*, 2023 :1–33, May 2023.

[34] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[35] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. LIMP : Learning Latent Shape Representations with Metric Preservation Priors. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12348, pages 19–35. Springer International Publishing, Cham, 2020. Series Title : Lecture Notes in Computer Science.

[36] Luca Cosmo, Mikhail Panine, Arianna Rampini, Maks Ovsjanikov, Michael M. Bronstein, and Emanuele Rodolà. Isospectralization, or how to hear shape, style, and correspondence. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7521–7530, June 2019. arXiv :1811.11465 [cs].

[37] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Spin transformations of discrete surfaces. *ACM Transactions on Graphics*, 30(4) :1–10, July 2011.

[38] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat : A new approach to computing distance based on heat flow. *ACM Transactions on Graphics*, 32(5) :1–11, September 2013.

[39] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion Models in Vision : A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9) :10850–10869, September 2023.

[40] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, February 2017. arXiv :1606.09375 [cs, stat].

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv :1810.04805 [cs].

[42] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep Geometric Functional Maps : Robust Feature Learning for Shape Correspondence. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[43] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. HOPE-Net : A Graph-Based Model for Hand-Object Pose Estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6607–6616, Seattle, WA, USA, June 2020. IEEE.

[44] Ke-Lin Du, Chi-Sing Leung, Wai Ho Mow, and M. N. S. Swamy. Perceptron : Learning, Generalization, Model Selection, Fault Tolerance, and Role in the Deep Learning Era. *Mathematics*, 10(24) :4730, December 2022.

[45] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. Single-Shot Motion Completion with Transformer, March 2021. arXiv :2103.00776 [cs].

[46] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints, November 2015. arXiv :1509.09292 [cs, stat].

[47] Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2) :179–211, March 1990.

[48] Victoria Fernández Abrevaya, Sandeep Manandhar, Franck Hétroy-Wheeler, and Stefanie Wuhrer. A 3D+t Laplace operator for temporal mesh sequences. *Computers & Graphics*, 58 :12–22, August 2016.

[49] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Muller. SplineCNN : Fast Geometric Deep Learning with Continuous B-Spline Kernels. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–877, Salt Lake City, UT, June 2018. IEEE.

[50] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2) :298–305, 1973. Publisher : Institute of Mathematics, Academy of Sciences of the Czech Republic.

[51] Simone Foti, Bongjin Koo, Danail Stoyanov, and Matthew J. Clarkson. 3D Generative Model Latent Disentanglement via Local Eigenprojection. *Computer Graphics Forum*, 42(6) :e14793, September 2023.

[52] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent Network Models for Human Dynamics. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4346–4354, Santiago, Chile, December 2015. IEEE.

[53] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. SpiralNet++ : A Fast and Highly Efficient Mesh Convolution Operator. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4141–4148, Seoul, Korea (South), October 2019. IEEE.

[54] Carolyn Gordon, David L. Webb, and Scott Wolpert. One cannot hear the shape of a drum, June 1992. arXiv :math/9207215.

[55] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734, Montreal, Que., Canada, 2005. IEEE.

[56] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José M. F. Moura. Adversarial Geometry-Aware Human Motion Prediction. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11208, pages 823–842. Springer International Publishing, Cham, 2018. Series Title : Lecture Notes in Computer Science.

[57] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to MLP : A Simple Baseline for Human Motion Prediction. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4798–4808, Waikoloa, HI, USA, January 2023. IEEE.

[58] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN : a network with an edge. *ACM Transactions on Graphics*, 38(4) :1–12, August 2019.

[59] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno. Human Motion Prediction via Spatio-Temporal Inpainting. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7133–7142, Seoul, Korea (South), October 2019. IEEE.

[60] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8) :1735–1780, November 1997.

[61] Tim Hoffmann and Zi Ye. A Discrete Extrinsic and Intrinsic Dirac Operator. *Experimental Mathematics*, 31(3) :920–935, July 2022.

[62] Qixing Huang, Xiangru Huang, Bo Sun, Zaiwei Zhang, Junfeng Jiang, and Chandrajit Bajaj. ARAPReg : An As-Rigid-As Possible Regularization Loss for Learning Deformable Shape Generators, September 2021. arXiv :2108.09432 [cs].

[63] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M : Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7) :1325–1339, July 2014.

[64] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN : Deep Learning on Spatio-Temporal Graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, Las Vegas, NV, USA, June 2016. IEEE.

[65] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. arXiv :1609.02907 [cs, stat].

[66] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. arXiv :1609.02907 [cs, stat].

[67] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1D convolutional neural networks and applications : A survey. *Mechanical Systems and Signal Processing*, 151 :107398, April 2021.

[68] Ilya Kostrikov, Zhongshi Jiang, Daniele Panozzo, Denis Zorin, and Joan Bruna. Surface Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, Salt Lake City, UT, June 2018. IEEE.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6) :84–90, May 2017.

[70] Lalit Kumar and Dushyant Kumar Singh. A comprehensive survey on generative adversarial networks used for synthesizing multimedia content. *Multimedia Tools and Applications*, March 2023.

[71] Tim Lebailly, Sena Kiciroglu, Mathieu Salzmann, Pascal Fua, and Wei Wang. Motion Prediction Using Temporal Inception Module. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Computer Vision – ACCV 2020*, volume 12623, pages 651–665. Springer International Publishing, Cham, 2021. Series Title : Lecture Notes in Computer Science.

[72] Andreas M. Lehrmann, Peter V. Gehler, and Sebastian Nowozin. Efficient Nonlinear Markov Models for Human Motion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1314–1321, Columbus, OH, USA, June 2014. IEEE.

[73] Na Lei, Zezeng Li, Zebin Xu, Ying Li, and Xianfeng Gu. What's the Situation With Intelligent Mesh Generation : A Survey and Perspectives. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–20, 2023.

[74] Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. Representation learning of 3D meshes using an Autoencoder in the spectral domain. *Computers & Graphics*, 107 :131–143, October 2022.

[75] Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. SpecTrHuMS : Spectral transformer for human mesh sequence learning. *Computers & Graphics*, 115 :191–203, October 2023.

[76] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. CayleyNets : Graph Convolutional Neural Networks With Complex Rational Spectral Filters. *IEEE Transactions on Signal Processing*, 67(1) :97–109, January 2019.

[77] B. Levy. Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13, Matsushima, Japan, 2006. IEEE.

[78] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional Sequence to Sequence Model for Human Dynamics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, Salt Lake City, UT, June 2018. IEEE.

[79] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks, September 2017. arXiv :1511.05493 [cs, stat].

[80] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A Survey of Convolutional Neural Networks : Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12) :6999–7019, December 2022.

[81] Fan Liang, Cheng Qian, Wei Yu, David Griffith, and Nada Golmie. Survey of Graph Neural Networks and Applications. *Wireless Communications and Mobile Computing*, 2022 :1–18, July 2022.

[82] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A Simple Approach to Intrinsic Correspondence Learning on Unstructured 3D Meshes. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, volume 11131, pages 349–362. Springer International Publishing, Cham, 2019. Series Title : Lecture Notes in Computer Science.

[83] Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion Models for Time Series Applications : A Survey, April 2023. arXiv :2305.00624 [cs].

[84] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep Functional Maps : Structured Prediction for Dense Shape Correspondence. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[85] Hsueh-Ti Derek Liu, Alec Jacobson, and Keenan Crane. A Dirac Operator for Extrinsic Shape Analysis. *Computer Graphics Forum*, 36(5) :139–149, August 2017.

[86] Zhenguang Liu, Shuang Wu, Shuyuan Jin, Qi Liu, Shijian Lu, Roger Zimmermann, and Li Cheng. Towards Natural and Accurate Future Motion Prediction of Humans and Animals. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9996–10004, Long Beach, CA, USA, June 2019. IEEE.

[87] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL : a skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6) :1–16, November 2015.

[88] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively Generating Better Initial Guesses Towards Next Stages for High-Quality Human Motion Prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6427–6436, New Orleans, LA, USA, June 2022. IEEE.

[89] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael Black. AMASS : Archive of Motion Capture As Surface Shapes. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5441–5450, Seoul, Korea (South), October 2019. IEEE.

[90] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History Repeats Itself : Human Motion Prediction via Motion Attention. In Andrea Vedaldi, Horst

Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12359, pages 474–489. Springer International Publishing, Cham, 2020. Series Title : Lecture Notes in Computer Science.

[91] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning Trajectory Dependencies for Human Motion Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9488–9496, Seoul, Korea (South), October 2019. IEEE.

[92] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Multi-level Motion Attention for Human Motion Prediction. *International Journal of Computer Vision*, 129(9) :2513–2535, September 2021.

[93] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodola, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections. In *2020 International Conference on 3D Vision (3DV)*, pages 120–129, Fukuoka, Japan, November 2020. IEEE.

[94] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections, November 2020. arXiv :2003.06523 [cs].

[95] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Spectral Shape Recovery and Analysis Via Data-driven Connections. *International Journal of Computer Vision*, 129(10) :2745–2760, October 2021.

[96] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Spectral Shape Recovery and Analysis Via Data-driven Connections. *International Journal of Computer Vision*, 129(10) :2745–2760, October 2021.

[97] Mathieu Marsot, Stefanie Wuhrer, Jean-Sebastien Franco, and Stephane Durocher. A structured latent space for human body motion generation, September 2022. arXiv :2106.04387 [cs].

[98] Julieta Martinez, Michael J. Black, and Javier Romero. On Human Motion Prediction Using Recurrent Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4674–4683, Honolulu, HI, July 2017. IEEE.

[99] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 832–840, Santiago, Chile, December 2015. IEEE.

[100] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In Gerald Farin, Hans-Christian Hege, David Hoffman, Christopher R. Johnson, Konrad Polthier, Hans-Christian Hege, and Konrad Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. Series Title : Mathematics and Visualization.

[101] Francesco Milano, Antonio Loquercio, Antoni Rosinol, Davide Scaramuzza, and Luca Carlone. Primal-Dual Mesh Convolutional Neural Networks, October 2020. arXiv :2010.12455 [cs].

[102] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5425–5434, Honolulu, HI, July 2017. IEEE.

[103] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning Convolutional Neural Networks for Graphs, June 2016. arXiv :1605.05273 [cs, stat].

[104] Rajendran Nirthika, Siyamalan Manivannan, Amirthalingam Ramanan, and Ruixuan Wang. Pooling in convolutional neural networks for medical image analysis : a survey and an empirical study. *Neural Computing and Applications*, 34(7) :5321–5347, April 2022.

[105] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps : a flexible representation of maps between shapes. *ACM Transactions on Graphics*, 31(4) :1–11, August 2012.

[106] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive Body Capture : 3D Hands, Face, and Body From a Single Image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10967–10977, Long Beach, CA, USA, June 2019. IEEE.

[107] M. Pegoraro, S. Melzi, U. Castellani, R. Marin, and E. Rodolà. Localized Shape Modelling with Global Coherence : An Inverse Spectral Approach. *Computer Graphics Forum*, 41(5) :13–24, August 2022.

[108] Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS : Generating Diverse Human Motions from Textual Descriptions. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13682, pages 480–497. Springer Nature Switzerland, Cham, 2022. Series Title : Lecture Notes in Computer Science.

[109] Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, 2(1) :15–36, January 1993.

[110] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL : Bodies, Action and Behavior with English Labels, June 2021. arXiv :2106.09696 [cs].

[111] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation, April 2017. arXiv :1612.00593 [cs].

[112] Marie-Julie Rakotosaona and Maks Ovsjanikov. Intrinsic Point Cloud Interpolation via Dual Latent Space Navigation. In Andrea Vedaldi, Horst Bischof,

Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12347, pages 655–672. Springer International Publishing, Cham, 2020. Series Title : Lecture Notes in Computer Science.

[113] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, July 2021.

[114] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D Faces Using Convolutional Mesh Autoencoders. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11207, pages 725–741. Springer International Publishing, Cham, 2018. Series Title : Lecture Notes in Computer Science.

[115] Dan Raviv, Michael M. Bronstein, Alexander M. Bronstein, and Ron Kimmel. Volumetric heat kernel signatures. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 39–44, Firenze Italy, October 2010. ACM.

[116] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands : modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6) :1–17, December 2017.

[117] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu. A Survey on Data-Driven 3D Shape Descriptors. *Computer Graphics Forum*, 38(1) :356–393, February 2019.

[118] Raif M. Rustamov. Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation. page 9 pages, 2007. Artwork Size : 9 pages ISBN : 9783905673463 Publisher : The Eurographics Association.

[119] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1) :61–80, January 2009.

[120] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet : Discretization Agnostic Learning on Surfaces. *ACM Transactions on Graphics*, 41(3) :1–16, June 2022.

[121] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet : Discretization Agnostic Learning on Surfaces. *ACM Transactions on Graphics*, 41(3) :1–16, June 2022.

[122] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-Time-Separable Graph Convolutional Network for Pose Forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11189–11198, Montreal, QC, Canada, October 2021. IEEE.

[123] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[124] Olga Sorkine. Laplacian Mesh Processing. *Eurographics 2005 - State of the Art Reports*, page 18 pages, 2005. Artwork Size : 18 pages Publisher : The Eurographics Association.

[125] Yuanhang Su and C.-C. Jay Kuo. Recurrent Neural Networks and Their Memory Behavior : A Survey. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[126] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*, 28(5) :1383–1392, July 2009.

[127] Zhiyu Sun, Ethan Rooke, Jerome Charton, Yusen He, Jia Lu, and Stephen Baek. ZerNet : Convolutional Neural Networks on Arbitrary Surfaces Via Zernike Local Tangent Space Estimation. *Computer Graphics Forum*, 39(6) :204–216, September 2020.

[128] Yongyi Tang, Lin Ma, Wei Liu, and Weishi Zheng. Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamic, May 2018. arXiv :1805.02513 [cs].

[129] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, pages 351–358, Not Known, 1995. ACM Press.

[130] Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. Modeling Human Motion Using Binary Latent Variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[131] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human Motion Diffusion Model, October 2022. arXiv :2209.14916 [cs].

[132] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv : Flexible and Deformable Convolution for Point Clouds, August 2019. arXiv :1904.08889 [cs].

[133] Matthew Thorpe, Tan Minh Nguyen, Hedi Xia, Thomas Strohmer, Andrea Bertozzi, Stanley Osher, and Bao Wang. GRAND++ : Graph Neural Diffusion with A Source Term. In *International Conference on Learning Representations*, 2022.

[134] B. Vallet and B. Lévy. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum*, 27(2) :251–260, April 2008.

[135] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[136] Nitika Verma, Edmond Boyer, and Jakob Verbeek. FeaStNet : Feature-Steered Graph Convolutions for 3D Shape Analysis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2598–2606, Salt Lake City, UT, June 2018. IEEE.

[137] He Wang and Juyong Zhang. A Survey of Deep Learning-Based Mesh Processing. *Communications in Mathematics and Statistics*, 10(1) :163–194, March 2022.

[138] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2) :283–298, February 2008.

[139] Yiqun Wang, Jianwei Guo, Dong-Ming Yan, Kai Wang, and Xiaopeng Zhang. A Robust Local Spectral Descriptor for Matching Non-Rigid Shapes With Incompatible Shape Structures. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6224–6233, Long Beach, CA, USA, June 2019. IEEE.

[140] Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. MGCN : descriptor learning using multiscale GCNs. *ACM Transactions on Graphics*, 39(4), August 2020.

[141] Yu Wang, Mirela Ben-Chen, Iosif Polterovich, and Justin Solomon. Steklov Spectral Geometry for Extrinsic Shape Analysis. *ACM Transactions on Graphics*, 38(1) :1–21, February 2019.

[142] Yu Wang and Justin Solomon. Intrinsic and extrinsic operators for shape analysis. In *Handbook of Numerical Analysis*, volume 20, pages 41–115. Elsevier, 2019.

[143] B. P. Welford. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 4(3) :419–420, August 1962.

[144] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1) :4–24, January 2021.

[145] Yun-Peng Xiao, Yu-Kun Lai, Fang-Lue Zhang, Chunpeng Li, and Lin Gao. A survey on deep geometry learning : From a representation perspective. *Computational Visual Media*, 6(2) :113–133, June 2020.

[146] Kai Xu, Vladimir G. Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, pages 1–38, Macau, November 2016. ACM.

[147] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models : A Comprehensive Survey of Methods and Applications, March 2023. arXiv :2209.00796 [cs].

[148] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A Survey on Deep Semi-supervised Learning. 2021. Publisher : arXiv Version Number : 2.

[149] Zi Ye, Olga Diamanti, Chengcheng Tang, Leonidas Guibas, and Tim Hoffmann. A unified discrete framework for intrinsic and extrinsic Dirac operators for geometry processing. *Computer Graphics Forum*, 37(5) :93–106, August 2018.

[150] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling, February 2019. arXiv :1806.08804 [cs, stat].

[151] Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences*, 12(17) :8643, August 2022.

[152] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image Diffusion Models in Generative AI : A Survey, April 2023. arXiv :2303.07909 [cs].

[153] H. Zhang, O. Van Kaick, and R. Dyer. Spectral Mesh Processing. *Computer Graphics Forum*, 29(6) :1865–1894, September 2010.

[154] Chongyang Zhong, Lei Hu, Zihao Zhang, Yongjing Ye, and Shihong Xia. Spatio-Temporal Gating-Adjacency GCN for Human Motion Prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6437–6446, New Orleans, LA, USA, June 2022. IEEE.

[155] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully Convolutional Mesh Autoencoder using Efficient Spatially Varying Kernels, October 2020. arXiv :2006.04325 [cs].

[156] Hao Zou, Zae Myung Kim, and Dongyeop Kang. A Survey of Diffusion Models in Natural Language Processing, June 2023. arXiv :2305.14671 [cs].

[157] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D Menagerie : Modeling the 3D shape and pose of animals, April 2017. arXiv :1611.07700 [cs].