Leveraging Neural Networks for 3D Facial Animation Editing

Eloise Berson

UNIVERSITE BRETAGNE LOIRE MATHSTIC

CentraleSupélec

THÈSE DE DOCTORAT DE

CENTRALESUPÉLEC

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Signal, Image, Vision*

Par

Eloïse BERSON

Leveraging Neural Networks for 3D Facial Animation Editing

L'Utilisation de Méthodes d'Apprentissage Profond au service de l'Édition d'Animation Faciale 3D

Thèse présentée et soutenue à RENNES, le 22 Janvier 2021
Unité de recherche : CNRS, UMR 6164 - IETR, (équipe FAST)
Thèse N° : 2021CSUP0001

Rapporteurs avant soutenance :

Damien ROHMER Professeur à École Polytechnique, Lix (Paris)
Saida BOUAKAZ Professeure à l'Université Claude Bernard Lyon 1, Liris (Lyon)

Composition du Jury :

Président :	Edmond BOYER	Directeur de Recherche INRIA (Grenoble)
Examinateurs :	Edmond BOYER	Directeur de Recherche INRIA (Grenoble)
	Catherine SOLADIÉ	Professeure Assistante à CentraleSupélec (Rennes)
	Nicolas STOIBER	Directeur Technique à Dynamixyz (Rennes)
Dir. de thèse :	Renaud SÉGUIER	Professeur à CentraleSupélec (Rennes)

Remerciements

Ces trois années de thèse furent pour moi une fabuleuse aventure, tant sur le plan personnel que professionnel, et je tiens ici à vivement remercier toutes les personnes qui m'ont aidée et soutenue pendant ces trois ans.

Je tiens en tout premier lieu, à exprimer toute ma gratitude envers mes SUPERS superviseurs ! Ce fut un réel plaisir de travailler avec vous pendant ces trois ans. Merci d'avoir été aussi présents et de m'avoir transmis autant.

Je remercie Catherine Soladié pour sa gentillesse et son enseignement au monde de la recherche. Présente et à l'écoute, Catherine fut une aide en or à la fois sur le plan scientifique et moral. Elle s'est constamment intéressée à mon travail et à mon avancée, et a toujours pris le temps de me rassurer et de m'écouter dans les moments difficiles. Sans ses encouragements, son soutien et ses conseils judicieux, cette thèse n'aurait pas été la même.

Je remercie également Nicolas Stoiber, pour la pertinence et la très haute qualité de sa réflexion scientifique, pour ses talents d'écrivain, et pour son enseignement remarquable qui n'ont cessé de me faire progresser. Je ne saurais jamais assez le remercier pour avoir cru en moi et pour tout ce que j'ai appris grâce à lui (et même à jouer au basket). Cela fut un véritable plaisir de travailler et de discuter avec Nicolas pendant ces trois ans, et je garderai toujours à l'esprit ses grandes qualités humaines ainsi que son soutien sans lequel je n'aurais pu mener à bien ce travail.

Cette thèse n'aurait pu avoir lieu sans la volonté de Gaspard Breton et Renaud Seguier et je tiens, ici à profondément vous en remercier. C'est votre appréciation pour la recherche et votre engagement qui m'ont permis d'effectuer ces trois ans de thèse dans des conditions idéales. Merci de m'avoir donné cette opportunité et de m'avoir accompagné tout au long de cette aventure.

Je tiens tout particulièrement à remercier Saida Bouakaz et Damier Rohmer, pour avoir accepté de rapporter cette thèse et pour le temps qu'ils ont consacré à l'évaluation de mes travaux de recherche, et Edmond Boyer pour avoir accepté de juger ma thèse. Je remercie l'ensemble des membres du jury pour l'intérêt qu'ils ont manifesté pour mon travail et pour leurs retours, très instructifs, qui ont contribué à parfaire ce manuscrit.

J'exprime toute ma gratitude envers toutes les personnes qui ont entouré cette thèse. Je tiens à particulièrement remercier l'équipe software pour votre aide durant ces trois ans, et notamment Emilie, Vincent, Marc pour votre aide et votre collaboration dans la réalisation de ces travaux de recherche ainsi que votre gentillesse et votre soutien. Je tiens également à chaleureusement remercier tous mes collègues de Dynamixyz, Flora, Pauline, Olivier, Julie, Florian, Louis-Paul, Sandrine, Lucas et Arthur, pour leur bienveillance et leur convivialité au quotidien. Travailler au sein de Dynamixyz a été un réel bonheur et je garde d'excellents souvenirs de l'ambiance unique de travail et de cette expérience parmi vous.

Mes sincères remerciements vont également mes collègues de Supelec Morgane, Adrien, Bastien, Lilian, Corentin, Simon, Sarah pour nos discussions, nos cafés, nos déjeuners.

Je tiens également à remercie Karine Bernard, Fréderic Jan et Gregory Cadeau, des alliés précieux à centralesupelec, pour votre aide dans les démarches administratives et la résolution des problèmes techniques.

Merci au sport et à tous mes coéquipières et coéquipiers de foot, qui m'ont permis de prendre du plaisir à jouer toutes les semaines.

Un très grand merci à Martin pour ces trois années à Rennes ensembles, pour nos parties de squash hebdomadaires, pour ses conseils et son optimisme sans faille. Merci à Roxanne, la plus incroyable des colocataires, d'avoir partagé mon quotidien pendant deux ans. Merci pour nos soirées, nos spas, nos matins chaotiques, et nos fous rire ! Un énorme merci à tous mes merveilleux amis d'être dans ma vie et pour votre soutien sans faille et votre amitié toujours réconfortante ! Mille Mercis à Bouyssoux, Pablo, Elsa, Karen, Juliette, Gathou, Pyv, Mathieu, Rémi, Arielle, Elorri, Nastassia, Brieuc, Marcu, Greg, et bien d'autres pour tous nos moments passés ensemble. Je remercie de tout coeur également Alexandre et Morgane qui continuent à me supporter et me soutenir depuis tout ce temps.

Je tiens à exprimer bien plus que ma gratitude à Alex, mon allié du quotidien, qui ne cesse de me soutenir et de croire en moi depuis plus d'an. Sans ta motivation et ton aide, cette fin de thèse n'aurait pas été la même.

Finalement je voudrais remercier ma famille d'être à mes côtés depuis plus de 26 ans. Merci mamie d'être un tel modèle de courage, de volonté et de motivation. Je n'aurais pu en arriver là sans le soutien infini et l'amour de mes parents, les meilleurs du monde, de mon frère, Johan, et de ma sœur, Justine. Merci de croire en moi et de me donner le courage qui me permet d'avancer. Je vous aime !

Contents

Résumé en Français

0.1 Contexte et Motivation

La synthèse de contenu animé 3D réaliste, conférant aux personnages virtuels l'illusion d'une allure humaine, est un rêve de longue date. Au cours des dernières décennies, l'intérêt et la demande pour ce type de contenu n'ont cessé de croître notamment avec la montée en puissance des images de synthèse et la présence de personnages virtuels dans de nombreux domaines (cinéma, les jeux vidéo, la robotique ou le domaine médical). Parallèlement, le degré de réalisme et la qualité attendue ont fortement évolué, élevant le niveau d'exigence des productions.

La capture de mouvements est une solution prometteuse, gouvernée par un idéal visant à satisfaire l'ensemble de ces exigences. En effet, cette nouvelle technologie permettrait de capturer et retranscrire toutes les subtilités du mouvement humain dans l'animation des personnages virtuels. En pratique, la synthèse d'animation convaincante demande encore beaucoup de temps et de travail, notamment pour l'animation du visage : tromper l'œil expert des humains, affûté depuis la naissance à reconnaître et distinguer les subtilités des signaux faciaux reste un défi majeur dans le domaine de l'animation faciale. Même les systèmes de capture de mouvements les plus sophistiqués nécessitent fréquemment une étape additionnelle de traitement de l'animation pour atteindre la qualité exigée en production. Il est courant que certaines parties d'une performance ne puissent pas être capturées, dû à l'occlusion du visage de l'acteur ou parce que celui-ci est dans une position inattendue par exemple. D'un point de vue technique, la résolution de la caméra, le bruit du système de capture, le changement de lumière et les mouvements brusques de l'acteur sont des facteurs impactant la qualité du signal d'animation et la précision de la capture, qui appellent à une étape d'édition supplémentaire. Enfin, des modifications et corrections supplémentaires sont nécessaires dans plusieurs cas également, même si le signal de capture de mouvement est "techniquement parfait" :

- L'objet/l'intention de l'animation a changé a posteriori ; il est préférable de réutiliser et d'adapter le contenu déjà capturé plutôt que de réutiliser un système de capture (qui peut être très coûteux et massif).
- Certaines poses/expressions manquent à l'animation finale, et doivent être

manuellement rajoutées après le tournage. Un cas d'application serait, par exemple, si l'acteur a oublié une partie ou n'a pu faire certains mouvements d'une séquence capturée, comme lever qu'un seul sourcil.

- Le mouvement capturé n'est pas parfait et des erreurs dans le jeu d'acteur (discours, intention, performance) peuvent subvenir. Des corrections manuelles sont alors nécessaires pour parvenir à une animation finale satisfaisante.

C'est pourquoi une étape d'édition d'animation est cruciale et souvent obligatoire pour remédier au manque de flexibilité de la capture de mouvement et satisfaire aux exigences de qualité.

Actuellement, cette étape reste très laborieuse: la complexité des signaux d'animation ainsi que l'abondance des données sont notamment des facteurs limitant les performances des solutions actuelles. D'un point de vue industriel, cette étape reste très coûteuse en temps et en compétence, restreignant son accessibilité et sa diffusion.

Dans cette thèse, on s'attache donc à comprendre le processus d'édition d'animation faciale et à fournir des solutions algorithmiques répondant aux enjeux que celui-ci soulève.

Positionnement du Problème La plupart des visages que nous rencontrons et avec lesquels nous interagissons sont en mouvement; Durant une interaction sociale, les humains sont systématiquement emmenés à décoder l'ensemble des signaux dynamiques, diffusés par les visages de manière plus ou moins évidente (ex: subtiles expressions faciales, mouvements conversationnels). De nombreuses études ont montré la sensibilité accrue de la perception humaine aux mouvements faciaux naturels [DBS18]. Fondé sur ces observations, l'animation faciale est alors considérée comme un processus dynamique, constituée de motifs spatiotemporels intrinsèquement corrélés. La modification, la correction de tels signaux, assurant la synthèse d'une animation cohérente et réaliste, apparait donc comme une tâche compliquée. Manipuler des données d'animation est d'autant plus fastidieux que celles-ci englobent une pléiade de motifs temporels et biomécaniques, soit non linéaires. En effet, le visage possède différentes parties formant un ensemble très corrélé, mais dont chaque composant varie selon des caractéristiques fréquentielles et dynamiques très différentes. De plus, la demande grandissante de données d'animation de haute qualité impose de fournir des solutions performantes et extensibles à de larges bases de données.

Les méthodes d'intelligences artificielles ont investi de nombreux domaines depuis leur création, et en grande majorité, ont surpassé les méthodes traditionnelles dans de diverses taches, de par leurs innovantes et attractives capacités à maîtriser et traiter de grandes quantités de données diversifiées, dont le processus sous-jacent n'était pas simplement ou efficacement modélisable jusqu'à lors. En effet, les modèles physiques et dynamiques actuels, bien que précis, présentent des caractéristiques de

complexité, de lourdeur et d'instabilité qui ne permettent pas leur utilisation à grande échelle, et un fonctionnement pertinent sur l'ensemble des exemples possibles. Plus particulièrement, les réseaux de neurones, entraînés à apprendre à partir d'exemples, se sont imposés en tant que solutions majeures les plus performantes dans différentes tâches, surpassant les méthodes statistiques traditionnelles. Une caractéristique largement exploitée dans nos travaux est leur attirante capacité à générer des sorties réalistes, avec des dynamiques semblables aux données vues durant l'entraînement, et à extrapoler ses résultats sur un grand nombre de situations non-observées. De plus, les réseaux de neurones peuvent aujourd'hui être combinés à des processus aléatoires et permettre une inférence générative, créant des données perceptiblement similaires aux données d'apprentissage [GPAM+14]. Combinés aux succès de ces méthodes à traiter des données temporelles telles que le signal de la voix [GJ14] ou du texte [SMH11], l'ensemble de ces raisons ont motivé notre choix d'explorer ce paradigme mathématique pour l'appliquer au domaine de l'animation et plus particulièrement aux divers aspects de l'édition d'animation faciale.

L'objectif principal de cette thèse est de comprendre et de mettre à profit les dernières avancées en apprentissage automatique afin d'améliorer l'édition de signaux dynamiques d'animation faciale. D'un point de vue applicatif, nos solutions visent à améliorer le processus d'édition de mouvements, réduisant l'implication laborieuse de l'artiste et le temps de ce jalon dans la production d'animation faciale. Les solutions proposées se doivent de respecter certains préceptes:

- **Plausible** : l'animation finale, bien que modifiée, doit rester convaincante, les mouvements du visage générés doivent être naturels et avoir une dynamique fidèle à la réalité.

- **Pratique** : nos solutions sont dévouées à respecter les contraintes imposées par le domaine de l'animation faciale, notamment le format des données dicté par les standards utilisés dans les chaînes de production d'animation.

- **Accessibilité** : les travaux développés dans ce manuscrit visent à simplifier la manipulation de signaux d'animation, et notamment à élargir le champ des représentations des données temporelles de mouvements faciaux, afin d'offrir une gamme d'alternatives aux traditionnelles paramétrisations temporelles bas niveaux. Dans la pratique, cela se traduirait par un souhait d'élargir l'accessibilité de la manipulation de données d'animation à un public plus large que celui des animateurs expérimentés.

- **Vitesse** : Un de nos objectifs est d'accroître l'efficacité du procédé d'édition et de modification de signaux d'animation faciale. Les solutions créées se doivent d'être temporellement performantes, de montrer un faible de taux de réponse pour effectuer l'inférence d'une animation.

Contributions Cette thèse est organisée autour de 3 contributions principales, focalisées sur les différentes facettes de l'édition d'animation, qui prennent place dans la chaîne d'animation faciale :

- Un algorithme temps réel de nettoyage et de perfectionnement d'animation faciale, basé sur une architecture neuronale. Nous proposons un algorithme de post-traitement, dévoué à l'amélioration de l'édition bas niveau des signaux d'animation, qui préserve et même restaure les éléments clés des mouvements du visage, de manière autonome, sans intervention manuelle. Basé sur une paramétrisation originale du problème, celui-ci répond aux exigences de la production puisqu'il est capable de traiter des signaux avec différentes fréquences d'images.

- Un système interactif d'édition d'animation *contrôlable* reposant sur une architecture neuronale temporelle et robuste. Nous proposons un algorithme de régression permettant de manipuler des signaux d'animations à partir d'un jeu réduit de paramètres hauts niveaux. Notre système est conçu pour automatiquement gérer et assurer la cohérence temporelle du signal d'animation produite et préserver les dynamiques du visage malgré des signaux d'entrée approximatifs.

- Une architecture générative d'édition d'animation faciale, permettant la manipulation et la synthèse d'animation non-supervisée, ou bien supervisée à partir de signaux de contrôle discrets, sémantiques ou bruités et approximatifs. Nous proposons une vision originale du problème d'édition d'animation comme équivalente au problème de l'*inpainting*, technique consistant à reconstruire ou recréer des parties manquantes ou endommagées d'une image.

Ces contributions ont été rendues possibles par un travail préliminaire de rassemblement et construction de deux bases de données.

Dans la suite de ce chapitre, nous allons présenter rapidement les bases de données utilisées tout au long de nos travaux, puis nous résumerons nos 3 contributions.

0.2 Base de données d'animations faciales 3D

Cette thèse s'attache à explorer de nouvelles méthodes d'édition d'animation reposant sur le florissant et prometteur paradigme des réseaux de neurones. Ce dernier n'est cependant utilisable qu'à condition de disposer de grandes quantités de données adaptées à la tâche d'apprentissage sur laquelle le réseau puisse être entraîné.

Bien que l'accès à des images de synthèse et du contenu médiatique se soit fortement amélioré ces dernières années, il n'existe que peu de bases de données d'animation faciale, suffisamment larges et diversifiées, permettant un apprentissage efficace des dynamiques naturelles des mouvements du visage. Outre le manque de

données 3D dynamiques, nous voulons, à travers ces travaux, adresser les différentes facettes de l'édition d'animation, telles que le nettoyage de signaux d'animation, le contrôle d'animation faciale, ou encore la manipulation d'animation faciale multimodale, composée d'un signal visuel et d'un signal audio. L'exploration de telles pistes de recherche impose, d'autre part, une large diversité et une richesse de métadonnées (annotations, données multimodales, ...) accompagnant les données d'apprentissage.

Pour ce faire, nous avons utilisé les récentes technologies de capture de mouvements, solution technologique prévalente pour créer à grande échelle des données d'animation de haute qualité, pour rassembler une première base de données, incorporant toutes les caractéristiques et les défauts des animations issues de cette technique ; Puis nous avons construit une seconde base de données multimodales en tirant profit d'une base de données existante et disponible.

Le premier jeu de données de paires d'animations, comme schématisé Figure 1; une animation est directement produite à partir de la capture de mouvements d'un sujet, et comporte ainsi tous les défauts et le bruit acquis lors de la chaîne d'acquisition et de production d'une telle animation ; une seconde est issue de l'annotation et travail manuel d'un animateur à partir de la même vidéo de capture. Ainsi, au lieu de simuler synthétiquement le bruit du processus de capture de mouvement, nous nous attachons ici à traiter le bruit "réel" contenu dans les données actuelles d'animation issues de ce processus, afin d'éviter de créer des solutions dépendantes d'un bruit donné et planifié. Ceci permet également de faciliter et de garantir une intégration plus optimale dans la chaîne de production d'animation.

Avec l'objectif de varier les formes d'éditions temporelles, nous avons souhaité utiliser des données d'animations plus riches, accompagnées de métadonnées pertinentes pour une analyse multimodale des mouvements faciaux. Nous avons donc constitué une seconde base de données multimodales en utilisant celle fournie par Fanelli et al. [FGR+10], nommée 3D Audio-Visual Corpus of Affective Communication (B3D(AC)^2). Celle-ci rassemble des *longues* séquences de 14 sujets récitant des phrases avec ou sans émotion, et fournie, en plus des données 3D pour chaque frame, les annotations phonétiques synchrones des paroles prononcées. Bien que diversifiée dans son contenu sémantique, et attractive pour sa taille et les données (et métadonnées) fournies, la qualité des données 3D ainsi que la paramétrisation des animations ne permettent pas une utilisation directe de ces séquences pour notre problème. Nous avons donc fait un premier travail préliminaire, utilisant des méthodes géométriques et d'optimisation actuelles pour pallier à ces lacunes. Au total, ce corpus amélioré rassemble 85 minutes d'animation, reposant sur le principe des blendshapes.

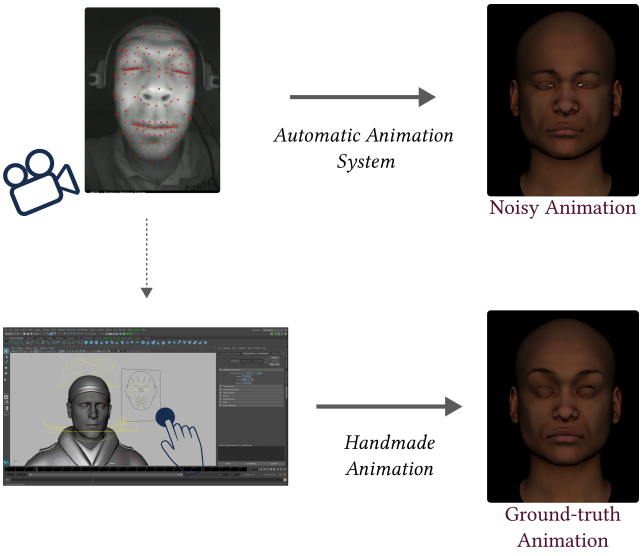


Figure 1 Présentation de la base de données d'animations issues de capture de performance. Une animation, bruitée et imprécise, est automatiquement générée à partir d'une vidéo originale grâce à un logiciel de suivi [Dyn19a]. Celle-ci n'a subi aucun post-traitement additionnel. Parallèlement, un animateur expérimenté a crée un animation *vérité terrain* associée à la vidéo originale, moyennant un logiciel professionnel [Dyn19b].

0.3 Filtrage de signaux d'animation en temps réel

Dans la chaîne de synthèse d'animation, l'édition la plus fréquente effectuée sur les signaux d'animation est le nettoyage et filtrage de l'animation. En effet, dû aux limitations des matériaux ou des logiciels qui la composent, il est très fréquent que l'animation en sortie soit bruitée ou même qu'elle ait subi des pertes. Ce post-traitement peut nécessiter la présence d'un animateur pour assurer une animation fidèle et réaliste ou d'un utilisateur pour adapter les paramètres de l'algorithme de traitement aux différents composants du signal d'animation.

Le nettoyage d'animation a été largement étudié pour traiter les erreurs de capture de mouvements du corps, survenant par exemple, lors de l'occlusion des marqueurs, du croisement des parties du corps ou encore à la détection des contacts avec l'environnement. Néanmoins, les travaux visant à améliorer à la capture des mouvements du corps se concentrent surtout sur les problèmes de détection, d'identification ou de correction de marqueurs.

Les méthodes actuelles de capture de mouvements faciaux, quant à elles, tendent à privilégier l'utilisation de technologie différente comme l'adaptation des signaux d'animation à un modèle global de visage. Contrairement aux technologies de capture de mouvement du corps, elles visent principalement à enlever les artefacts (sauts, bruit) produits par le détecteur qui apparaissent dans le signal d'animation de sortie. Les systèmes existants de capture en temps réel se sont en majorité tournés vers des techniques de traitement du signal classiques, restituant une animation lissée. Néanmoins, ces méthodes sont souvent appliquées aux paramètres d'animation de manière indépendante, omettant la cohérence spatiale qui existe au sein des différentes parties du visage, et échouent à restituer toutes les subtilités des mouvements du visage. De plus, fidèlement régler et ajuster ces méthodes à chaque coefficient d'animation est souvent une tâche laborieuse, imputable à la diversité des dynamiques qui gouvernent le visage humain. En effet, un clin d'œil forme un motif abrupt, court mais intense dans le signal du mouvement des paupières, doit être préservé tel quel ; A contrario, un tel motif dans le signal du mouvement du nez se doit d'être lissé.

Pour pallier ces lacunes, de nombreux chercheurs se sont tournés vers les méthodes d'apprentissage par ordinateur pour fournir des solutions capables de filtrer une grande diversité de signaux d'animation. Néanmoins ces solutions reposent sur des systèmes non causaux, de par l'utilisation d'architecture convolutionnelle [BBKK17, HSKJ15] ou de filtres adaptatifs [MLCC17]. Dans des premiers travaux, nous proposons de suivre cette orientation de recherche et d'utiliser l'apprentissage par ordinateur pour développer un système temps réel permettant de précisément et automatiquement filtrer et même restaurer la dynamique d'une animation faciale bruitée.

Notre approche s'appuie sur un modèle neuronal récurrent qui apprend de manière causale à générer un signal propre et précis, en s'appuyant sur les prédictions passées

et les caractéristiques du mouvement du signal bruité au cours du temps.

Apprendre de manière causale la dynamique des mouvements faciaux pour une application en temps réel ne peut s'appuyer sur une fréquence d'image constante et fixe. En effet, les sources de capture d'images telles que les webcams, n'ont pas une fréquence d'images constante au cours du temps. D'autre part, les algorithmes de rééchantillonnage classiques ne sont pas applicables dans ce cas, car ils nécessitent la connaissance des échantillons futurs.

Nous avons abordé le problème en reconsidérant le signal de mouvement sur lequel nous avons entraîné notre réseau. En exploitant les premières dérivées temporelles du signal, normalisées par l'intervalle de temps qui sépare deux images, l'information sur la fréquence d'images n'est plus directement contenue dans le signal d'entrée, et ainsi le système n'est plus dépendant de ce paramètre. Le réseau est également alimenté de l'information sur les estimations précédentes afin de traiter n'importe quelles animations, aussi longues soit elles.

La méthode proposée est alors plus précise que les méthodes de traitement du signal et permet une restauration du signal d'animation plus réaliste que des méthodes non récurrentes.

0.4 L'Édition d'Animation Faciale Contrôlable

Bien que le signal soit parfois techniquement parfait, le contenu même est souvent sujet à corrections. Ces modifications sont généralement faites par un animateur, expérimenté à manipuler des paramètres d'animation tels que les coefficients de blendshapes. Comme le nombre de paramètres est souvent élevé et que ces coefficients possèdent généralement un effet localisé, ayant une interprétation assez bas niveau, le travail d'édition prend beaucoup de temps. Contrairement au filtrage et nettoyage du signal animation, considéré comme de l'édition bas-niveau d'animation, dans ce chapitre on s'intéresse à une édition plus haut niveau de l'animation, dont les modifications effectuées se doivent de respecter des contraintes définies par un utilisateur.

Les méthodes précédentes de manipulation sémantique d'animation proposant des solutions temporelles pour propager les corrections ponctuelles de l'animateur [SLS+12, ASK+12] considèrent le problème d'optimisation dans l'espace des vitesses ou utilisent un modèle spatio-temporel bilinéaire. Bien que créatrices d'animations lisses, la résolution temporelle de ces méthodes dépend fortement d'hyperparamètres qui demandent à être manuellement ajustés, rendant la tâche d'édition plus laborieuse. Au contraire, nous avons opté pour une méthode d'apprentissage, automatisant la modélisation des mouvements du visage et de leur dynamique, afin de s'abstraire de toutes tâches de planifications, tout en produisant des animations convaincantes.

Ces travaux s'inscrivent dans la continuité de méthodes de contrôle de mouvement

du corps humain à partir d'apprentissage par ordinateur [HSK16]. Dans des travaux fondateurs du domaine, Holden et al. [HSK16] utilisent des réseaux convolutionnels pour construire une variété représentant les mouvements du corps, puis la régression de paramètres de dynamiques du mouvement du corps hauts niveaux à cette variété. Naviguer dans cet espace (par le biais de paramètres de contrôle) permet d'assurer la production d'animations réalistes. Cependant, nous avons démontré que ce système n'est pas applicable directement aux mouvements du visage, car il ne permet pas de préserver les mouvements hautes fréquences, tels que les fermetures de bouche ou les clignements des yeux, primordiaux dans la création d'animations convaincantes.

Néanmoins, les réseaux convolutionnels (CNN), appliqués le long de l'axe temporel, présentent des propriétés de continuité et d'invariance temporelle particulièrement adaptées pour encoder des motifs de mouvement. Contrairement dans le cas du filtrage d'animation, lorsque l'utilisateur souhaite modifier une animation, il désire voir le signal entier avant, afin de cibler les portions à retoucher et les changements à effectuer. L'aspect non-causal des CNNs permet de répondre à cette exigence. Ces raisons nous ont poussés à étudier une architecture convolutionnelle de régression, conservant les hautes fréquences [RFB15], afin de mettre en correspondance les paramètres de contrôle haut niveau choisis et les coefficients d'animation. Comparé aux systèmes précédents de contrôle de mouvement du corps, notre système préserve les mouvements hautes fréquences, générant alors une animation faciale plus fidèle.

Ainsi dans un deuxième temps, nous proposons une nouvelle méthode d'édition haut-niveau s'appuyant sur une structure neuronale robuste et de haute précision temporelle, adaptée à la modélisation spatiotemporelle de mouvements faciaux, et permettant une représentation plus fidèle des motifs dynamiques gouvernant le mouvement du visage. Comme les travaux de Seol et al. [SLS+12], notre système considère à la fois la nature spatiale et temporelle du signal d'animation. Toutefois, contrairement aux travaux précédents, l'approche proposée autorise des signaux d'entrée exagérés ou imprécis, tout en assurant la production de mouvements plausibles. En effet, le réseau, entrainé sur des données d'animations réalistes, assimile la cohérence temporelle des mouvements du visage et assure ainsi la génération d'une animation convaincante.

Nous avons spécifiquement conçu note système pour être robuste à ce genre de situation ; pour cela, nous avons ajouté un auto-encodeur débruiteur à la suite de notre régresseur afin d'assurer une animation réaliste en sortie en toute circonstance.

En pratique, la solution avancée s'accompagne d'une interface graphique par laquelle un utilisateur, non expérimenté, peut modifier une animation à partir de peu de paramètres sémantiques. Le système proposé fonctionnant avec peu de latence, il autorise l'utilisateur à interactivement appliquer ses modifications, observant instantanément le résultat de ses changements. Cet outil de visualisation a été aussi aidé à démontrer les performances du système, sa rapidité d'inférence ainsi que la robustesse. Nous avons démontré l'utilité et la pertinence du système sur le cas d'une

animation faite à partir de motion capture.

0.5 L'Édition d'Animation Faciale Générative

Bien que le système précédent permet une modélisation précise et robuste des signaux de mouvements faciaux. Néanmoins celui repose sur un modèle de régression, exigeant un signal d'entrée temporel géométrique dense et non ambiguë. La régression impose également une stratégie d'entrainement déterministe, qui présume la présence d'une animation cible à générer et d'une métrique pertinente qui atteste la qualité de l'animation produite. Cette méthode empêche par exemple un contrôle précis à l'échelle d'une expression, tel que le paramétrage par plan de séquence utilisé par les animateurs. Poussant les limites de la synthèse d'animation et de la modification d'animation, nous sommes ensuite tournés vers les systèmes de génération de signaux temporels et avons proposé une nouvelle approche du problème d'édition d'animation.

Nous proposons une manière originale de concevoir le problème d'édition d'animation, qui s'appuie sur le parallèle dessiné entre l'*inpainting* d'image et l'édition d'animation. L'image inpainting vise à remplacer des parties manquantes ou non désirées d'une image par des portions de pixels automatiquement générées, de manière à recréer une image réaliste. L'édition d'animation poursuit un objectif semblable, celui de remplacer ou régénérer des portions de signal de manière automatique. Pendant longtemps, les solutions d'image inpainting reposaient sur des méthodes bas-niveau de traitement de signal (gradient d'image). Bien qu'efficaces, ces approches s'avèrent incompétentes lorsque les parties manquantes, à régénérer, sont trop grandes ou très complexes. Un système neuronal génératif plus élaboré a alors vu le jour, le GAN développé par Goodfellow et al. [GPAM+14], et a permis de combler les parties manquantes avec des motifs visuels complexes à partir de peu ou pas de signaux d'entrée. Au cours des cinq dernières années, les GAN ont démontré leur efficacité en atteignant ainsi des résultats états de l'art dans différents domaines.

Les objectifs poursuivis dans ce chapitre sont semblables, bien qu'appliqués à un domaine différent d'édition. Nous avons donc proposé une approche générative, reposant sur le système du GAN, afin de régénérer de manière non supervisée ou faiblement supervisée, des portions d'animation manquantes, non satisfaisantes ou endommagées. Au lieu de comparer directement les animations produites à une animation cible bien définie, le modèle d'entrainement du gan évalue par lui-même si l'animation produite est plausible, réaliste.

Les GAN se composent d'un générateur, générant des animations réalistes à partir d'animations partiellement masquées, et facultativement, d'un signal de supervision, et d'un discriminateur qui assure le réalisme de l'animation générée. Afin d'apprendre les dynamiques complexes des mouvements du visage, nous proposons une structure récurrente bidirectionnelle, assurant une cohérence entre mouvements passés et futurs et respectant l'aspect non-causal du processus d'édition par un utilisateur.

La structure neuronale proposée peut également être façonnée et augmentée [MO14], élargissant les possibilités de contrôler et de synthétiser une animation à différents formats de paramètres de contrôle telles que :

- Non-supervisé : masquant de manière aléatoire des portions de l'animation, notre système comble les parties manquantes avec des mouvements réalistes, qui se raccordent aux bords de manière cohérente. Nous avons démontré que notre système permet de combler de 'longues' occlusions, souvent présentes dans les tournages de capture de mouvements, de manière plus convaincante qu'un système linéaire.
- Paramétrisation parcimonieuse d'expression. Notre système s'assure de reconstituer la temporalité du signal, restituant ainsi une animation réaliste, à partir de contraintes spatiales (expressions faciales) définies à certains points dans le temps. Ce cas d'inférence substituerait en pratique les fonctions interpolatrices, et non réalistes intégrer dans les outils d'animations actuelles, et utilisées lors de corrections manuelles faites par l'animateur lorsqu'une expression est manquante ou bien qu'une fermeture de bouche doit être corrigée par exemple.
- Animation bruitée. Notre système peut également jouer le rôle de débruiteur, restituant un signal lisse à partir d'un signal initial incomplet (masqué) et d'un signal de contrainte approximatif. En effet, en pratique, il est courant qu'une partie de l'animation ne soit pas satisfaisante : au lieu de refaire une session de tournage, l'utilisateur pourrait modifier une partie de l'animation à partir d'une autre animation bruitée, enregistrée à partir d'une webcam ou d'un mobile par exemple.
- Séquence de phonèmes. Une animation parlée est souvent sujette à correction lorsque le message n'est pas correct. Ainsi, notre système peut modifier une animation pour correspondre à un nouveau signal sémantique donné en entrée, tel qu'une séquence de phonèmes (regrouper ensuite en visèmes, selon leur représentation visuelle). Un exemple illustrant cet usage est le changement de mots dans le discours.

Le temps d'inférence de notre approche étant faible, celle-ci se relève ainsi plus performante que les méthodes manuelles actuelles. D'un point de vue pratique, notre solution permettrait de réduire le temps de l'édition et le travail de l'utilisateur actuellement nécessaire pour modifier une animation de manière convaincante.

0.6 Conclusion

Les travaux de thèse, passés en revue dans ce chapitre, visent à explorer les facettes de l'édition d'animation. De nouvelles approches ont alors été proposées pour améliorer les techniques actuelles de filtrage et de manipulation d'animation, respectant les

critères définis en introduction. Nous avons présenté de nouvelles méthodes d'édition bas niveau (section 0.3) et de plus haut niveau (sections 0.4 et 0.5) qui assurent la production d'animations faciales réalistes. L'utilisation de méthodes d'apprentissage par ordinateur a été mise en avant, afin de faciliter la manipulation des signaux temporels d'animation tout en préservant et même générant, les dynamiques naturelles des mouvements du visage. Les approches proposées présentent d'autres avantages, tels que leur rapidité d'inférence et leur généralisation à de nombreux motifs de mouvement de visage, qui émanent des architectures neuronales explorées.

Ce manuscrit s'articule autour des trois principales contributions de cette thèse, succinctement présentées ici, et détaille nos investigations.

CHAPTER 1

Introduction

Contents

1.1 Context

Making virtual characters move and act as humans has been a long-time goal. In many applications including the medical fields, robotics, communication, and the entertainment sector, the pursuit of this dream has never stopped. It is all the more true for the last decades, with the increasing appearance of synthetic characters in films, video games, and recently virtual reality and beyond. Indeed as observed in Figure 1.1, the demand for 3D content has undergone a rising growth, driven by the overall trend toward 3D Visual Effect Technologies (VFX), 3D visualization, 3D gaming or even 3D mobile applications [1]. Simultaneously, the desired level of realism of media content has evolved. The current animation production has to deal with an always increasing volume of animation data while reaching higher quality requirements.

One particular attention is paid on the human face, which is among all the different body parts, the one that conveys the richest, and most complex visual information, such as the identity, gender, age and emotions, allowing humans to interact in a challenging environment. The elaborated neural musculature innervation and flexibility of the face, as well as facial features devoted to recognizing and interpreting of facial communicative signals, have evolved in tandem with the development

[1] https://www.marketresearchfuture.com/reports/3d-animation-market-2760

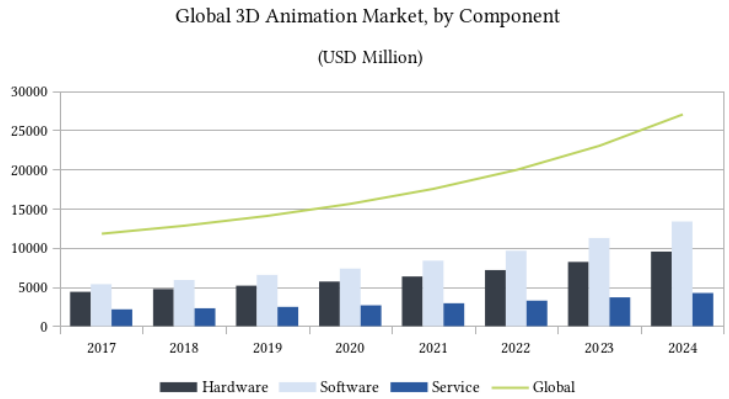


Figure 1.1 Global 3D Animation Market by Component (Source Market Research Future (MRFR)). According to a market report published in September 2019[1], the 3D animation market is expected to continue growing in the future due to the demand for Visual Effect Technologies (VFX), 3D visualization, 3D gaming, and 3D mobile applications.

of more sophisticated social interactions [Tov95]. Indeed, the muscles of the face have a "more complex pattern of innervation of extrafusal fibres than other skeletal muscules" [WV18]. Processing facial cues is an ubiquitous operation in our lives and involves a larger portion of our brain than the processing of other objects [RHD12]. Behavioral, neuroimaging, and other brain function studies [FRQL00, APM00, Kan10] have demonstrated that our brain involved specific and dedicated strategies to grasp faces information. These findings promote the hypothesis that our brain preferentially focuses on faces.

Despite the remarkable progress made within both the computer animation research and the industrial communities, designing aesthetic and realistic human-like character animation is still a tedious and highly time-consuming procedure. Indeed, since birth, the human brain is specially tuned to human face behavior [SG15]. Hence, human beings are experts at discerning inconsistencies in facial motions, even the most subtle ones, such as missing lip contact when the mouth closes during a speech. So that any implausible details can drastically reduce the perceived aspect of animation.

For the past two decades, motion capture has evolved into the leading technology to create realistic animation, making the process of animation content generation more reliable and accessible. It is now commonly used to produce body and facial animations in numerous applications. Motion Capture (MoCap) refers to the pipeline of recording performer movements and transferring the animation signals to a digital character. An illustration of this pipeline is shown in Figure 1.2. The theoretical

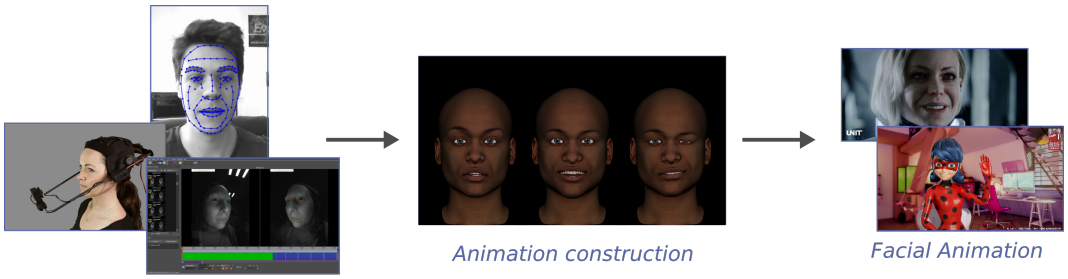


Figure 1.2 Illustration of the Motion Capture animation pipeline. The first step of this pipeline consists of capturing the subject's facial expression, with or without markers. Various MoCap systems exist, including simple devices (webcam, mobile phone) or a more sophisticated adapted technologies including magnetic or optical systems (eg Head Mounted Camera (HMC)), resting upon active or passive sensors. An automatic animation system processes the collected motion data to derive the corresponding animation parameters from the recorded performer movements. Finally, a retargeting system, transferring the animation deformations, enables animating virtual characters.

promise of MoCap is the ability to completely and flawlessly capture and retarget a human performance, from emotion down the most subtle motion of facial skin. However, the raw MoCap animation data hardly ever comes out perfect, requiring an additional editing step.

In reality, even professional motion capture setups often fall short of a perfect animation result. It is for instance usual that some part of a performance cannot be captured due to occlusions or unexpected poses. For facial MoCap, popular video-based technologies have known flaws as well: the camera resolution limits the capture precision, while signal noise, jitter, and inconsistent lighting can impair its robustness. Even so, the result of a performance animation does is "technically" perfect, it may not entirely fulfill the artistic intent (the performer wasn't able to reach some poses..), requiring additional corrections. Several reasons might explain the necessity of a supplementary editing step [Gle99, Hav06]:

- The intent/purpose of the animation has evolved after the shooting. Rather than capturing a new animation, with an expensive and massive setup, one might prefer reusing and adapting an already captured sequence.
- The real movement is not perfect: the performer is human and therefore, prone to make mistakes in the speech, the intent, or in its acting. Further postprocessing refinements are often necessary to meet the requirements.
- Adding motions are needed after the shooting. For instance, some motion might not be feasible by the performers (e.g., lift only one eyebrow) or have

been forgotten during the shooting, which can be manually drawn afterward; Virtual character's animation need further motion magnification to express the desired intent.

- The resulting retargeted animation is not directly usable and required additional manual works. Most of the time, transferring motion from one character to another produces a lot of artifacts. Morphological differences between the performer and the driven digital character can lead to inconsistencies in the final motion [SSK+11].

Hence, it is common to require human intervention to fine-tune the animation, addressing the lack of flexibility of performance-based animation. In the prevalent animation editing scheme, an animator directly manipulates a set of numerous low-level temporal curves of complex facial parameterizations. The industry standard for facial animation is the blendshape coefficients. This workflow usually goes hand in hand with the traditional and widespread keyframing process, a form of animation editing especially cherished by the artist community. Artists specify the configuration of the character at specific key points in time and let automatic interpolation generate the inbetweens motion.

In both workflows, the editing step requires technical and artistic skills to end up with a coherent and satisfactory final animation. In addition, this process requires a considerable amount of time as the expression of the face has to be manually specified over a certain period. Hence, handmade animation editing remains an expensive step in the performance-based facial animation production, that only the animation studios, capable of employing talented and experienced animators, can afford. The objective of this thesis is to study the complexity of the facial animation manipulation, and investigate potential algorithms to make this editing task more intuitive.

1.2 Problem Statement and Motivation

In this thesis, we consider the dynamic process of facial animation, i.e., the temporal behavior of the face when performing a sequence of expressions. The motion editing problem is defined as modifying a segment, or the full content of an unsatisfying animation, and then resynthesizing a plausible animation under potential external constraints guiding the generated motion segment [Gle01].

The Complexity of Facial Dynamics The major difficulty with facial motion manipulation comes from the inherent dynamic aspect of facial animation. Indeed, the human brain is very sensitive to recognizing biological motions and distinguishing real or fake movements from subtle temporal cues of facial motion. From the cognitive perspective, it has been widely proven that facial movement perception is vital in social and non-language communications. Giese and Tomasio [GP03] even report that complex movements and action recognition are crucial for the survival of

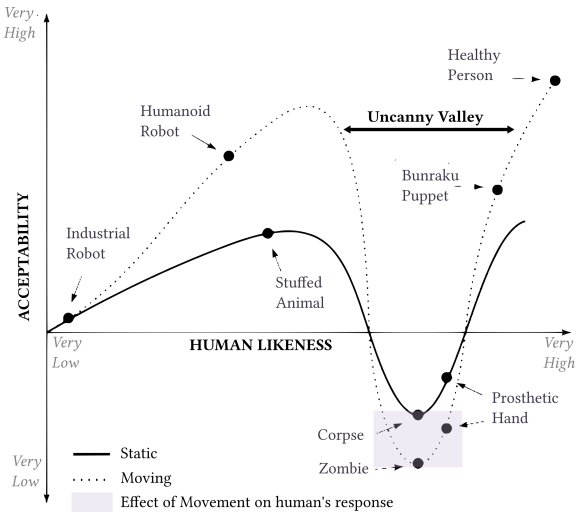


Figure 1.3 The Uncanny Valley hypothesis adapted from Mori works [Mor70, MMK12]. The uncanny valley is the area of negative response to characters that seem too human like. Movement amplifies the observer's response, in particular, deepen the uncanny valley (MacDorman and Ishiguro, 2006; Mori, 1970/2012).

many species. This hypothesis is in correlation with the phenomena of "Uncanny Valley", remarked by Mori et al. [Mor70, MMK12]: when an artificial character acquires higher similarity to human beings, it heightens the sense of affinity to the observer. However, when it starts acting almost as perfect as a human, one might notice a very strong drop in comfort and familiarity feelings. The presence of movements is crucial in human sensitivity response and amplified the peaks and the valleys of the "uncanny valley" graph, as we can see in Figure 1.3. These findings, combined with the fact that we mostly witness moving faces in the real world, put forward the importance of the temporality to synthesize high quality and believable facial animation.

That motivates our choice to regard facial behavior as an inherently dynamic rather than a static mechanism. The main requirement in animation editing is, therefore, to preserve the naturalness of facial motions and the consistency of facial expressions. However, facial animation is composed of a plethora of complex and highly nonlinear bio-mechanical temporal patterns, spreading out over a wide range of different temporal frequencies, which is challenging to model. Previously, data-driven research into animation synthesis has offered an alternative to complex models. Yet, much of the facial animation editing solution still required animation skills, preventing a broader deployment of great quality facial animation.

The Animation Data Management Another difficulty with motion editing is the duality of spatial and temporal coherence. Indeed, motion capture provides long sequences of animation data at every timestamp. As the movement is continuous,

motion corrections require changing many consecutive frames. A deep and empirical analysis has to be made before, to properly define the scope of the motion to edit inside the full animation. Facial motion data is all the more challenging to modify due to the strong and intricate correlation between the different parts of the face. In the physiological and cognitive fields, it has been widely proven that the perception of facial movements implied distinct areas of the face [WMK05, MK14]. For instance, through a Magnetoencephalographic study on facial movements, Miki and Kakigi [MK14] demonstrate the crucial spatial relationship between the contour and facial components, which are essential factors in the perception of the facial movements. Although it is imperative to preserve the whole face's coherence during the editing process, it is still difficult to precisely determine which sub-part of the face has to be revised and how it has to be modified. The difficulty comes from the diversity among the dynamics and the temporal features of the different parts of the face, requiring adaptive processing. Indeed, facial dynamic induces both low- and high-frequency of complex motion that is hard to model [ASK+12, BBG+14], rendering the facial animation filtering/cleaning problem all the more challenging. Hence, with the growth of synthetic data production, it appears necessary to provide solution scaling to large data sets of motion.

The Neural Networks Revolution For the last decades, learning-based methods, especially Neural Network (NN), have introduced innovation and advances to many research areas. Neural networks are a programming paradigm, a category of Machine Learning (ML) in which a computer is trained to perform a specific task by learning from examples. This paradigm easily derives an efficient statistical model of the process, which is relevant notably in the case of complex process modeling. A Neural Network can automatically capture the correlation and the temporal dependency between the movement of the different parts of the face. Biologically inspired, it is a stack of several layers of linear classifiers with nonlinear functions, named *neurons*. NN have become popular since the breakthrough of deep learning on the ImageNet classification challenge in 2012 [KSH12]; Back up with the well-demonstrated universal approximation property of multi-layer networks, their usage has been widely extended. In particular, neural networks have enabled modeling of complex temporal signals, such as the voice in speech recognition [GJ14], the image sequence in video compression, text-based image retrieval [ECPC18] or text-to-text [SMH11] or text-to-image generation [RAY+16].
In our context, neural networks have several advantages. As explained above, it bypasses the need for an accurate 3D facial motion model, which is highly challenging to derive and limit the generalization across various performance-based animations. Training neural approaches on real human facial motion allows condensing all the motion variations features and specificities into a mathematical process, which can be used to estimate new realistic motion afterward. In addition, neural networks allow being combined with random schemes, making possible to perform generative inference and create animation perceptually similar to the ones observed in the

training data. Recently, learning-based systems have outperformed regular ones in an abundance of tasks, opening novel possibilities to handle the complexity of human motion signals [HSK16] and generate perceptually high-quality facial animation results [TZK+17, LSSS18] using innovative optimization strategy [GPAM+14]. These successes, coupling with the inherent capability of neural networks in managing large and diversified data set, have motivated our work.

In this thesis, we explore the different aspects of facial animation editing and leverage machine learning methods to overcome the mentioned difficulties. We investigate ways to improve and facilitate the animation editing process and provide innovative tools that reduce the time and the manual effort involved in this task.

The main objective of this thesis is, thus, to design suitable and practical solutions to create powerful facial animation pipelines. The proposed solutions should be compliant with the following requirements:

- **Plausibility**: the complex space-time patterns of human facial motion should be respected. We want to provide solutions that preserve the dynamic and the naturalness of facial motion, in particular high-frequency movements, such as the blinks or the mouth closures in a speech-based facial animation.

- **Practical**: we wish to design solutions that can be easily integrated into the traditional facial animation pipeline, and supporting the actual animator's workflow.

- **Usability**: we aim at developing solutions compliant with animator requirements, but also accessible to a wider public than skilled animators. A facial animation editing framework should offer an alternative to the actual editable animation curves, with higher facial motion abstractions, more usable for non experts.

- **Speed** One of our goals is to make the process of facial animation editing more efficient by reducing the manual effort and relevantly resynthesis the modified animation.

1.3 Contributions

This thesis is organized around three main contributions that can find their application in the performance-based facial animation production pipeline:

- A real-time cleaning and refinement facial animation system. A post-processing performance-based animation signal algorithm that both preserves -and even restores- crucial facial cues, without relying on any manual intervention. We define an original parametrization of the problem to free our system from framerate dependency, enabling a real-time inference.

- A robust and interactive editing methods that automatically handles the temporal consistency of facial motions and preserves their natural dynamics. Based on semantic and high-level parameters, our system aims at pushing forward the intuitiveness of the facial animation editing while allowing a flexible motion controlling.

- A generative editing framework enabling the synthesis of motion segment in both a supervised and unsupervised fashion. The proposed solution enables fast facial animation modifications through versatile guidance inputs including discrete, noisy or semantic guidance.

These contributions are backed up by a preliminary work gathering and refining existing data to create a compliant database with respect to our requirements.

1.4 Thesis Organization

In this thesis, we aim at studying learning-based algorithms to advance the different facial animation editing process further. Leveraging the power of neural-based approaches, we explore different directions to handle various animation editing procedures.

First, we review the relevant related works in the field of facial animation editing in Chapter 2, covering the performance-based animation processing, the keyframing editing task, and the facial motion controllability.

Throughout this thesis, we leverage neural-based techniques to design editing solutions, preserving the natural dynamic of facial motions. Yet, these techniques rely on accurate animation data to operate well. In Chapter 3, we list the two databases used for all our experiments.

In Chapter 4, we investigate neural-based techniques to provide a real-time facial animation refinement system, removing artifacts in performance-based animation while preserving and restoring the natural dynamic of facial motions.

In Chapter 5, we present our second contribution that steps toward motion controllability. Currently, the animation editing remains dedicated to professional use; To address this shortfall, we propose a time resolution-preserving architecture, capable of generating complex and plausible motion patterns, from a set of few intuitive temporal high-level parameters.

In Chapter 6, we explore the generative learning process to forge a multifunctional framework that handles different supervised or unsupervised editing scenarios such as discrete keyframing editing, motion filling during occlusion, expression corrections, semantic content modifications, and noise filtering.

Finally, Chapter 7 concludes this thesis. We discuss the proposed solutions, our contributions and suggest future improvements. This thesis is based on the work published in the following articles:

- E. Berson, C. Soladié, V. Barrielle, and N. Stoiber. A Robust Interactive Facial Animation Editing System. In *Proceedings of the 12th Annual International Conference on Motion, Interaction, and Games - MIG '19*, Newcastle-upon-Tyne, United Kingdom, 2019
- E. Berson, C. Soladié, and N. Stoiber. Real-Time Cleaning and Refinement of Facial Animation Signals. In *Proceedings of the 4th International Conference on Graphics and Signal Processing - ICGSP 2020*, Nagoya, Japan, 2020
- E. Berson, C. Soladié, and N. Stoiber. Intuitive facial animation editing based on a generative rnn framework. In *Proceedings of the 19th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '20*, Montreal, 2020

CHAPTER 2

Literature Review

Contents

A tremendous amount of research efforts has been made in animation synthesis since the pioneering work of Parke et colleagues [Par72], handling the different aspects of the production pipeline [PASH13, NN98, DN07, ZTG+18]. As our thesis focuses on one crucial step of the facial animation pipeline, with the general goal of making the

facial editing process easier and faster without defacing the natural dynamic of facial motions, it follows an extensive body of literature.

In this chapter, we provide an overview of the main techniques concerning facial animation editing, omitting various aspects of the facial animation field, including 2D facial manipulation, face tracking or rendering techniques. Although this thesis focuses on facial motion, motion editing has been widely studied in the field of body motion. Hence, this chapter also incorporates the most relevant body motion editing methods for this thesis.

Throughout this thesis, we hypothesize a given *blendshape* parameterized animation sequence. Introduced in [Par74], *Blendshapes* are "linear facial models in which the individual basis vectors represent individual facial expressions." [LAR+14]. Due to the simplicity of the mathematical model, its interpretability and its straightforward implementation, this parametrization has been extensively used in the literature. The blendshape model also offers the advantage of being a semantic parametrization; As a consequence, it is the prevalent paradigm to create facial animation in the film industry. Among others, the blendshape model has animated characters in The Curious Case of Benjamin Button [Flu11] and The Lord of the Rings. A complete overview of the different rigging process and geometrical face deformation parametrization techniques are also beyond the scope of this work and can be found in [OBP+12].

We consider three categories of animation editing: keyframe-based editing (Section 2.1), motion controlling (Section 2.2) and animation processing (Section 2.3).

Historically, the initial approach to create and manipulate animation was *keyframing*, introduced in computer graphics in 70s [BW71]. It is the most basic form of animation creation and has been used since the first animated film, *The Dinosaur*, invented by Winsor McCay in 1914. The name, *keyframing* is derived from the traditional animation conception process, where every frame (pose) of animation was hand-drawn. The essential idea consists of correcting static key shapes at different specific moments in time, and then use interpolation to generate the motion between the edited keyframes and propagate the revised shape. Therefore, the illusion of movements is created by smoothly filling the sequences with *inbetweens* using manual tools, interpolation function, or a more sophisticated dynamic modeling process.

In the beginning, the most important frames, keyframes, were drawn by expert animators, involving handicraft techniques [Bla94], and *in-betweens* were then designed by a group of animators or by utilizing keyframing instruments. Since then, many techniques have proposed to ease the keyframe drawing part, using geometrical considerations (Section 2.1.1) or higher facial abstractions (Section 2.1.2). More sophisticated computed interpolation schemes have also substituted the laborious task of inbetweeners (Section 2.1.3). In the first section, we will, therefore, reference to the main techniques used to ease the static facial expression correcting approaches (Section 2.1).

Although keyframing is still widely popular, the prevalent approach to create an animation is performance-based techniques [PSS99, Wil06]. While motion capture is not directly suitable for the stage of motion editing, a quantity of work focuses on reusing

motion data to fit external constraints. In the second section, we will analyze the different strategies developed to edit the facial motion by using pre-captured motion clips (Section 2.2). Early works focus on reusing motion samples through different concatenation strategies. More recently, with the growth of available multimodal data, many works rely on learning techniques. We will detail those two approaches in Section 2.2.1 and Section 2.2.2 respectively.

A later tendency consists of directly performing temporal motion manipulation on the original signal. We depicted them in a last part (Section 2.3). Temporal motion manipulation strategies distinguish themselves from the previously described procedures in that they do not consider frames individually nor portions of animation.

Finally, we discuss the positioning of this thesis with respect of the existing literature.

2.1 Keyframing Editing

Historically, computer animation systems allow the user to control a set of discrete "key" frames, by specifying the configuration of the character at certain key points in time. The temporal continuity of the motion is generated afterward, without the ability to lead interpolating frames. In the most common scheme, users provide geometrical constraints to a keyframing tool which then solves the realistic animation parameters configuration matching these user's specifications. This process was primarily applied for determining the configuration of the character's parameters from temporally localized specifications, named *Inverse Kinematic*. Later, this concept was transposed in the facial animation field [LA10], known as the *Direct Manipulation* technique. We detail various methods existing in the literature for solving the underlying animation parameters in Section 2.1.1. However, geometrical constraints require skills and experience to be accurately defined. Many studies have imagined more intuitive scheme for non-expert users, while still allowing supervised keyframe editing. An overview of these propositions is presented in Section 2.1.2. Given the set of keyframes, the motion is synthesized by interpolating between these frames. In the beginning, the in-betweens frames workflow was labor-intensive and highly time-consuming as it was entirely made by hand. Since the pioneer work of Parke et al. in 1972 [Par72], many studies have explored new ways to automate and accelerate this process. The most relevant interpolation strategies are exposed in Section 2.1.3. This section ends with a discussion of the advantages and drawbacks of the current keyframe-based editing solutions.

2.1.1 Direct Manipulation

Since the early 2000s, creating a tool to help support the motion editing procedure has been an active area of research in the computer animation community. Animation editing was firstly explored in the literature, to manipulate character body motion, employing *inverse-kinematics* formulation [ALCS18]. Gleicher et al. [Gle01] describe

the IK as "a process for determining the configuration of a character's parameters (known as its pose) based on specifications of resulting features of the pose, such as end-effector positions."

This definition has been later adjusted to the facial animation field: facial editing consists of solving the key animation parameters from static users' constraints. Therefore, the crucial issue of inverse kinematic is to generate reasonable faces from only sparse constraints, which is a severely underconstrained problem. Early tendencies use direct manipulation for editing keyframe facial pose [JTDP03, ZLH03, ZSCS04, CGZ17] and handle this issue by leveraging reduced dimension space and computing the animation parameters maximizing their likelihood with respect to the data, using Principal Component Analysis (PCA) [BV99, ACP03, LCXS09] or Independent Component Analysis (ICA) [CFP03]. Further local improvements of the edited methods quality were achieved by segmenting the face into hierarchical regions a priori [JTDP03, ZSCS04, LD08, MLD09, TDlTM11]. Another solution to edit 3d vertices on the mesh is Radial Basis Functions (RBF) as it guarantees smooth geometric deformation [NFN00, SSK+12, SL14]. A new expression is derived from a sparse and localized collection of steering parameters, defined on or near the surface of the mesh [NFN00, ZSCS04]. To precisely render the nonlinearity structure of the face, Seol and colleagues [SSK+12] allow the artist to sculpt nonlinear corrections on the blendshape-based animation by exploiting the weighted pose space invented by Kurihara [KM04]. In the proposed framework, the vertex-level adjustments are interpolated using RBF inside the pose space, which is the blendshape model. Additional blendshapes can also be drawn anywhere into this pose space, appearing smoothly at the specified location. Xu et al. [XCLT14] give the user the possibility to control the retargeting process by editing fine-scale details (wrinkles) on the transferred animation, improving the visual rendering. In the same vein, Feng and colleagues [FKY08] suggest kernel canonical correlation to generate plausible facial deformations on animation from a sparse, underdetermined set of control points.

Lewis and Anjyo [LA10] introduce a novel paradigm called *Direct-Manipulation*, formulating the regression from 3D vertices constraints to blendshape weights as the regularized optimization constraints. The novel framework enables the artist to perform fast "pin and drag" operations on the 3D mesh's surface and solves for the underlying animation parameters, as illustrated in Figure 2.1. The number of users' constraints is usually smaller than the blendshape parameters. The optimization problem is thus regularized through different criteria: by constraining the value of the blendshape coefficients [LA10], by employing a statistical model [ATL12, WBLP11], by constructing an orthogonal blendshape model [LD08], by using geometric constraints [RZL+17], with transposition approach [CLO17] or, by adding face areas boundary constraints [TDlTM11]. Neumann and colleagues [NVW+13] perform the optimization on sparse and localized *deformation components*. These linear subspaces are automatically computed by imposing sparsity in the direct manipulation optimization problem, resulting in a *sparse PCA* [ZHT06]. Seo et al. [Seo11] subsume this framework but propose an innovative matrix compression scheme to handle non-

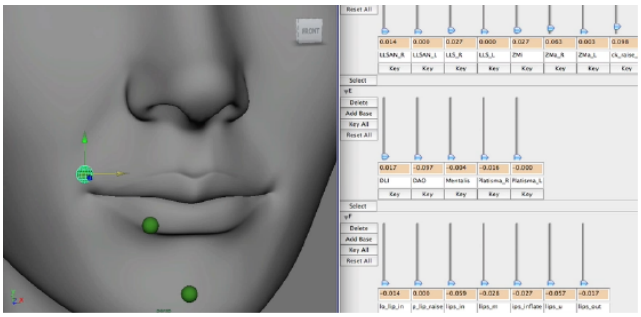


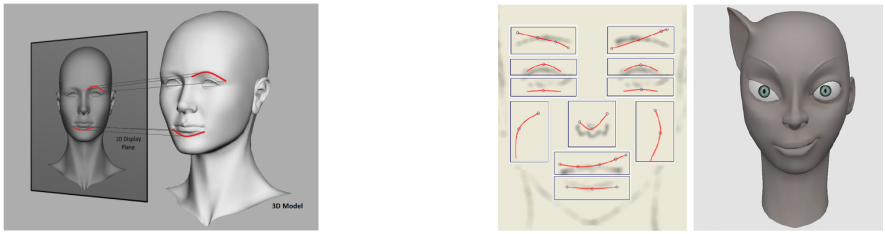
Figure 2.1 Rather than manipulating traditional animation parameters, Lewis et al. [LA10] propose a direct manipulation framework, that enables the artist to edit the face by directly moving vertices. The main challenge of direct manipulation is to effectively solve for the underlying parameters (the sliders) that generate a face which best matches the vertex constraints.

linearity effectively. Later, Cetinaslan et al. [CO18, COL15] suggest a new approach to localize the direct manipulation method theory and limit the spatial influence of each modification, avoiding global peculiarity.

Direct manipulation relies on "end effectors" or low-level geometrical constraints. While enabling another kind of control for animators such as the "pin-and-drag" operation directly on the 3D facial expression, it is not adapted to a non-expert user for it requires high animation skills.

2.1.2 Keyframe Editing Constraints

Creating efficient and intuitive facial expression editing tools encompass a reflection on the user's interface and the form of the control parameters. However, in this thesis, we are looking for a user-friendly solution to manipulate facial motion. Traditionally, artists edited facial expressions by changing the animation parameter values through a sliders-based interface. Although animators are familiar with this kind of interfaces, they are allowed modifying one parameter at the time, leading to switching back and forth operations between different sliders to generate the desired facial expression [LA10]. Sliders are hardly intuitive, as their effects are often low level and restricted to a particular part of the face. Combined with the fact that they require to be wisely manipulated to avoid reaching unrealistic configurations, it becomes clear that sliders remain dedicated to a targeted public, with high animation skills. Early works stepping toward more intuitive facial modeling and editing drive the face mesh by interactively dragging points on the face which are either image features [ZLH03], motion markers [JTDP03], the 2D projection of 3D vertices [ZSCS04, LA10, CGZ17] or lines on a 2D portrait [SMND08]. One group of works build upon these methods and enhance the facial animation modeling access through sketch-base interfaces. We



(a) (b)

Figure 2.2 Creating plausible facial expressions requires mastering sophisticated facial parametrizations, which is challenging for non-expert users. (a) Many works propose alternative user interfaces to ease facial geometry manipulation and solve for the underlying facial rigs parameters [CO18]. (b) Miranda and collegues [MAO+12] propose a sketching interface control system, allowing the user to deform a 3D face by drawing strokes.

review these methods in Section 2.1.2.1. Another group of work proposes to use more semantic inputs to edit animation. We present this category in Section 2.1.2.2.

2.1.2.1 Sketching interface

A group of work addresses the lack of accessibility to current rigging systems for novices using *sketching interface*, introduced in 2006 by Natanelli and colleagues [NF06] and extended in 2008 by Chang and colleagues [CJ08]. In 2009, Lau et al. [LCXS09] propose an interactive sketch-based framework to ease 3D facial expressions posing. Facial expressions are edited by drawing free-form strokes [GM10, CO18, ZHK+17], by defining distances between facial points, by incrementally manipulating curves on the face, or by directly specifying facial points in 2D screen space. The difficulty of the sketch-based procedures lies in the interpretation of the user's inputs, which may be consistent with many unnatural facial expression configurations. Sketch-based interfaces are a popular strategy to give a non-expert user the possibility to perform keyframes editing, manipulating only facial parameters [COL15, LCXS09], 2d landmarks [ZHP+19, LWW+18], fine details [ZHP+19] or just the lips shape [DBB+18]. Disambiguation of the user's input is handled using maximum likelihood framework [LCXS09, GM10], non-uniform rational basis spline curves (NURBS) [CO18], reference curves [CJ08], or PCA [SMND08]. Zhao et al. [ZHP+19] formulate the problem of sketch-based editing as reconstructing and deforming a 3D face model. The user's sketches serve at deforming 2D landmarks, and thereby the 3D model parameters, a new photorealistic face is rendered via the 3D model parameters. Rather than modifying animation parameters, Miranda et al. [MAO+11, MAO+12] define a external canvas allowing the users to edit the underlying animation rig by drawing 2D strokes (see Figure 2.2). In the same vein, Jin et al. [JGGN15] propose a sketch-based interface to easily edit a variety of meshes.

These above-mentioned studies mainly focus on keyframes expression editing conceiving spatial abstractions of a facial representation, without considering the temporal consistency of motions. Recently, several works investigate spacetime curves for editing body motions [GRGC15, CiRL+16, CGNS17, COS19]. Nevertheless, these approaches still involve controlling each keyframe individually and therefore, specifying a pose at each time. While the notion of trajectories is intuitive for body parts (arms, foot), representing time variations of facial motion is still an open question.

2.1.2.2 Semantic Guidance

Another group of works rests on more semantic user constraints for keyframing editing. One form of constraint is emotion cues. For instance, Stoiber and colleagues [SSB08] design a 2D interface with an emotional interpretation to control facial expressions. Several methods rely on discrete phonemes-related guidance, such as the viseme -the visual representation of a phoneme (speech unit)- [CM93]. Another tendency is to derive from data, high-level parameters using Independent Component Analysis [CFP03] or a bilinear model [CDB02]. In this way, they can modify the speech and emotional content independently.

Higher abstractions of facial expression are desirable for intuitive and fast correction computation. While high-level discrete parameters have been easily derived from body motion editing [HSK16], it is still an open question for facial animation. One effective solution is semantic or high-level keyframes representation. Reed and Cosker [RC19] circumvent this issue and create an evolutionary interface, in which a non-expert user can edit an animation by iteratively choosing the most appropriate example, until convergence to the desired one. Although this method is user-friendly and flexible; it precludes a rapid and efficient application. Yet, the effect of corrections should not be temporally localized due to the dynamic nature of facial motion and has to be spread smoothly through the surrounding frames to maintain the temporal coherency of the original sequence. Thus, this editing strategy must be coupled with an editing propagation scheme.

2.1.3 Automatic Interpolation

Improving facial manipulation and providing high-level parameters are a crucial ingredient to simplify the editing pipeline. However, as exposed in the introduction, it is not sufficient for generating a realistic animation. Creating a believable edited animation pass through rendering the natural and complex dynamics of the facial motion. It is all the more challenging, as humans are very sensitive to detecting temporal inconsistencies in facial motions (Chapter 1). The impression of movement was initially created by stacking an infinite number of hand-drawn frames, making objects moving in *cartoon* style. Later, in the process of facial animation editing, only the most important frames, keyframes, were drawn by expert animators and *in-betweens* were then designed by a group of animators or using keyframing tools.

Although the motion was more natural; it required hours and millions of in-betweens to generate a whole animation film [JT81]. When animation became computer-generated instead of hand-drawn, the laborious task of *in-betweeners* has been replaced by predefined interpolation functions, exposed in Section 2.1.3.1. Recently, many work propose to connect keyframes using generative techniques, outlined in Section 2.1.3.2.

2.1.3.1 Interpolation Functions

Keyframe interpolation often relies on a linear workflow, enabling artists to control and edit the in-between frames easily. Indeed, many animation software (Maya, Blender, ...) still propose the keyframing procedure with automatic in-between frames solving. Most of the automatic motion completion systems apply basic interpolation [Par72], usually linear, because of its simplicity and its execution speed. More sophisticated interpolation methods were investigated such as a bilinear [AKA96] or spline function [KB84], collisions were further detected to correct interpolation [Neb99]. The pioneer work of [Par72] suggests keyframing with cosine interpolation between 3D surface-based poses. Later, several studies considering the temporal behavior of the face propose to spread the edition wielding a more sophisticated function such as a Catmull-Rom spline [LD08] or a B-spline curve [CLK01] on the edited animation parameters. While interpolation has proved efficient for short segments with dense sets of keyframes, these curves are agnostic about the nature of the animation, restraining the quality of the produced animation. The smooth and monotonous motion patterns they produce are far from realistic facial dynamics when used on longer segments. Nowadays, animation cleaning usually relies on *keyframing*: having artists replacing faulty animation with numerous carefully-crafted keyframes to interpolate a new animation. Not only *keyframing* is a time and skill demanding process, but also it requires acting on several of the character's low-level animation parameters, which is not intuitive for non-experts.

2.1.3.2 In-betweens Generation

Several works explore automatic hand-drawn in-betweens generation [BW75] in the field of human motion. Initial methods propose user-guided systems by looking for correspondence between strokes, and automatically interpolate between matched strokes [Kor02, WNS+10]. Dalsein and colleagues [DRvdP15] continuously interpolate in both space and time, 2D vector drawing through a novel structure, named the Vector Animation Complex. With the aim of reducing the manual involvement of the user, many works opt for deep learning models, filling large gaps of motion segments between sparse keyframes constraints, without an explicit supervision on the transition motion control. An overall trend to generate in-betweens automatically is recurrent neural networks [ZvdP18, HP18]. Recently, considering that human motion dynamics can be learned from data, Zhang et al. [ZvdP18] learn in-betweens patterns

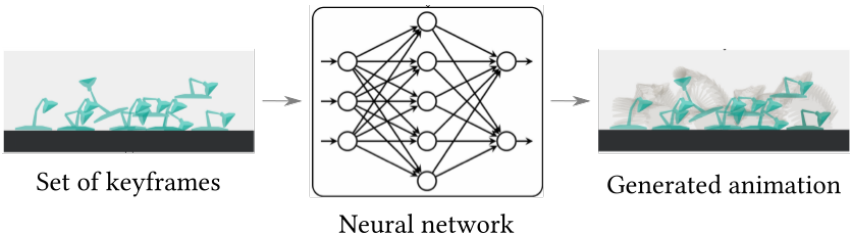


Figure 2.3 In the traditional keyframing workflow, the motion is synthesized by interpolating between a set of given keyframes. Most common automatic completion systems apply predefined functions that cannot reflect the complex non-linearity of facial dynamics. Recent works propose to leverage neural networks to learn realistic motion patterns, producing a believable animation. Zhang et al. [ZvdP18] use an auto-regressive model that interpolates a hopping lamp motion given a set of keyframes.

with an autoregressive two-layer recurrent network to automatically autocomplete a hopping lamp motion between two keyframes, as shown in Figure 2.3. Harvey and al [HP18] propose a Recurrent Transition Network (RTN), extending the RNN to more complex body-motion actions. However, their system only works for a fixed interval of 60 missing frames. Building upon their previous work, the same authors [HYNP20] combine a LSTM structure and an adversarial training to generate automatic motion transitions. Their system handles various in-betweening gaps by inducing temporal awareness through an inventive *time-to-arrival* embedding. They enforce the variability in the generated transitions using a innovative scheduled noise strategy. Zhou et al. [ZLB+20] use a fully Convolutional Neural Network (CNN) in an autoencoder architecture to interpolate motion in long-term segments guided by sparse keyframes positioning at arbitrary points in the timeline, demonstrating good results on full-body motions. The motion of face and body limbs is naturally the deterministic outcome of muscle contraction between bones and tissues, obeying physical laws. Yet, facial muscles and skin layers have a smaller scale and a much denser structure than our main body muscles. These methods have not been tested on facial motions, which are exhibit dynamic patterns that are significantly different than those of larger body limbs (higher frequency, low inertia).

Discussion Keyframing remains popular due to its linear workflow, enabling artists to control and edit specific keyframes easily, which is valuable for artists [IMH06]. Although manual keyframing can conduce to impressive results, the main interpolation scheme fails at reproducing the natural temporal mechanic of facial motions, which might be very disturbing in case of long-time gaps. While recent studies in characters animation editing interface propose to steer both the temporal and spacial dimension of the body motions by sketching line-of actions [GCR13] and gesture drawings [BVS16], most of the facial animation framework still provides no mechanism to control and

manipulate the resulting displayed motion. Handmade fine-tuning can conceal unrealistic transitions, but it is certainly far from being a fast solution. Hence, actual keyframing editing frameworks are ineffectual regarding our requirements. Considering the temporal aspect of the motion into an editing framework was mainly introduced in body animation with the concept of *spacetime* editing [WK88, Gle97]. Seol and colleagues [SLS+12, XCLT14] propose an innovative optimization scheme to propagate edits in a more plausible way employing a movement matching equation. In the same spirit, Dinev and colleagues [DBB+18] use a gradient-based algorithm to smoothly propagate sparse mouth shape corrections throughout an animation. Tangent-based optimization has been newly adopted to perform spacetime body-motion editing [KG18, COS19]. While ending-up with smooth animations, these strategies rest on a set of well-edited and accurately chosen keyframes to represent the motion, that is inappropriate to non-expert users. Besides, the resulting animation's quality is still dictated by the number and relevance of user-created keyframes [ZLB+20]. Recently, Neural Network have shown promising results to synthesize very natural human motion transitions [ZLB+20, ZvdP18]. Even so, the presented solutions remain action specifics. In this thesis, we propose to incorporate the keyframing paradigm into a global facial animation editing framework, tackling the above issues by leveraging recurrent networks and generative framework which have recently demonstrated successful results at modeling human motions [ZvdP18], but remain unexplored in the field of facial animation.

2.2 Motion Controllability

Although keyframing is still in use, the leading technique to produce animations is performance-based animation, as the capture data intrinsically encompass the natural facial motion dynamics. Real-time mocap solutions enable to drive online an avatar [NSX+19, WBLP11, CWW+16]. However, it is an ineffective approach to synthesizing new animations satisfying constraints. For instance, performing a particular expression, at a specific time, or synchronizing a viseme with a given audio track may require many iterations before succeeding in acceptable results. Hence, many works has come up with systems that adapt performance-based animation data to meet temporally-dense external constraints, usually user defined, authoring the generation of motion sequence. Several methods resynthesize motion by simulating the dynamics of muscles and skins using the universal laws of physics [BSC16, SNF05, IKKP17]. Building and simulating a plausible physical-based model is an arduous task because of the intricate dynamics of the facial musculature and the layers of skin tissues. Rather than manipulating a set of frames independently without explicit guidance on the dynamic of the synthesized motion, in this section, we focus on techniques that aim to control the generated motion sequence. A strategy to edit motion data is reusing and appropriately recombining available examples using *motion blending* approaches or by cleverly concatenating samples. In Section 2.2.1, we review the most relevant works, an exhaustive overview on this topic may be

found in [Gle08].

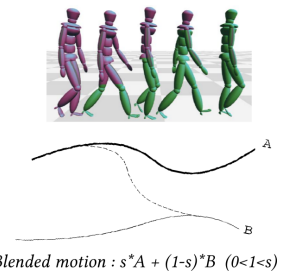
In more recent works, there has been an overall trend towards learning-based methods leveraging motion capture data availability. Rather than concatenating existing motion, another strategy consists of learning an underlying parametrization of motion from motion capture data. A new sequence can be synthesizing afterward in an inference step, respecting user constraints. Early systems rely on statistical models but recently a flourishing panel of works investigates neural network paradigms to generate natural motion matching input constraints. These approaches are presented in Section 2.2.2. We discuss the limitation of the current approaches at the end of this section.

2.2.1 Motion Sample Editing

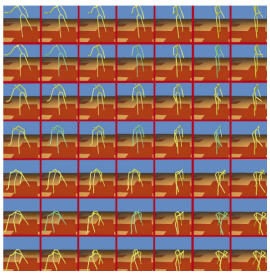
Reusing and manipulating motion samples implies different strategies to bond motion segments together. This can be done by blending motion segments (Section 2.2.1.1) or by wisely concatenating animation samples (Section 2.2.1.2).

2.2.1.1 Motion Blending

Motion blending refers to the process of interpolating two or more motion clips, to produce a sequence of motion respecting desired properties [FHKS12]. It is illustrated in Figure 2.4a. Relying on motion capture examples, it was a promising approach for motion editing and motion generation under the user's constraints. Early approaches linearly blend motion samples to create a new sequence [WH97]. However, linear interpolation limits the complexity of the produced motions and precludes reproducing of the face's mechanical and nonlinear behavior. Hence, researchers have explored different strategies to cleverly stick together existing sequence with K-Nearest Neighbors interpolation [KG03] or decomposing the motion into human behavior using Fourier coefficients [UAT95]. Rose and colleagues [RCB98] distinguish types of human action, called "Verbs", and their style "Adverbs". They enhance previous work by adding a nonlinear function, RBF [PSS02, RISC01], to interpolate between "Verbs" under a set of "Adverbs" constraints, making possible to create various motion sequences (see Figure 2.4b). Kernel-based interpolation function has been widely adopted to interpolate motion clips [MK05] smoothly. One downside of this technique is its inability to exactly specified constraints. Many efforts strengthen the kernel-based blending motion process, coming closer to space constraints, by optimizing the blending weights afterward in "Inverse Blending" framework [HK10] or proposing a (geo)statistical approach [MK05]. As demonstrated by Huang et al. [HK10], inverse blending framework is suitable for generating parameterized walking motion steered by precise feet placement controllers. Body-motion samples were recently blended using neural network interpolation [GLSR19]. However, the proposed system tends to oversmooth the dynamic in the blending motion part and does not generalize well on complex and specific motion actions.



(a) Traditional motion blending.



(b) The original RBF interpolation [RCB98]

Figure 2.4 One popular technique for animation synthesis is motion blending. (a) It consists in interpolating similar motion clips using blending weights, in order to produce a sequence that respects some high-level constraints. (b) Rose et al. [RCB98] develop a new motion interpolation scheme combining the Radial Basis Functions with polynomial terms, and generate new sequences from parameterized motion examples, call "Verbs" characterized by their style labeled by "Adverbs".

Overall, motion blending is a straightforward technique to generate natural and excellent quality results, which has seen widespread adoption in various motion synthesis domains. Although kernel-based methods are the most popular, it is linked with a high computational cost [RB05]. For that reason, subsidiary approaches have been explored.

2.2.1.2 Motion Concatenation

One strategy to resynthesizing a new motion is to concatenate motions clips. The principle is to retrieve from a large corpus of motion data, a corresponding motion clips sequence that fits the user's specifications. As the found motion piece was not modified, it contains all the subtle details of the original animation. We distinguish two main ways to concatenate motion samples presented in this section: nonparametric sampling techniques and motion graphs.

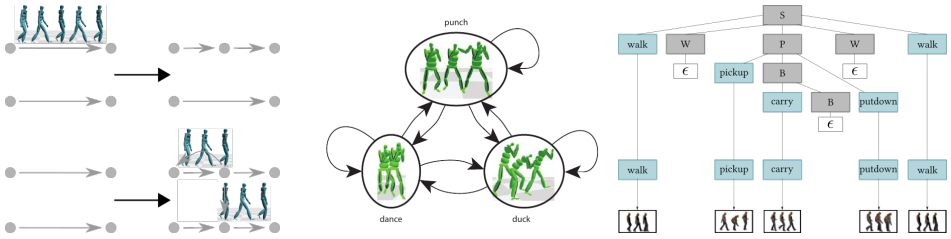
Nonparametric Sampling Methods Inspired by texture synthesis [EL99, LLX+01], one approach consists of randomly reassembling motion data patches from qualified candidates in the input motion. Deng et al. [DBNN04] consider a phoneme projected representation as patch samples and regenerated speech animation with this kind of patch-based texture sampling procedure. Recently, Roberts and colleagues [RAAL19] improve a blocked facial animation (an animation that contains the basic movements and timings but details are needed), by concatenating motion patches from motion capture samples, including rich and subtle details of facial movements. They use patch-based sampling and Gaussian falloff to synthesize new animation containing the original animation's coarse motion information and the feature of realistic facial motions. Bermano and colleagues [BBG+14] enhance the temporal behavior of a facial animation using the optimal projection of this animation onto a pre-captured motion

capture database. The projecting sequence is then resampled to get the desired number of frames. However, sampling methods cannot extrapolate samples beyond the ones in the database. Besides, managing and properly sectioning a sequence of clips yielding plausible and natural animation sequences is a daunting task.

Motion Graph An effective technique to perform temporal coherent editing and generation is motion graphs [KGP02, AF02, LCR+02]. Motion Graph optimizes the path to smoothly combine a set of motion clips, respecting the user's constraints. This technique leans on building a graph where nodes encode static poses or short-term motion blocks [AF02]. The edges between nodes encode the likelihood of the transition between those two blocks being plausible, so realistic animation reconstruction consists of finding paths of minimal cost in the graph. A distance metric evaluates the extent to which a transition is plausible. The metric distance measures the similarity between either the motion parameter positions [LCR+02], velocities, accelerations of motion samples, [AF02] or even the vertices at the boundaries of the motion sample [KGP02]. Pioneer works [AF02] in this direction develop a hierarchical graph representing the connectivity of motion data and conducting a global randomized search to retrieve a satisfying animation. Kovar et al. [KGP02] use a "branch and bound" search algorithm to perform efficient controllable motion generation. This workflow has been later exploited for speech-driven animation [CFKP04]. Lee et al. [LCR+02] propose a similar approach but relies on cluster trees to encode plausible transitions between motion data. At runtime, the graph can be navigated to recreate a convincing animation sequence, meeting the user's constraints. While the edges enforce the similarity between two nodes, small discontinuities might be noticeable, requiring a blending algorithm [LCR+02, LS99] or a smoothing function [AF02]. Kovar et al. [KGP02] precompute a distance matrix between frames in the database and constructed automatic smooth transitions at local minima (see Figure 2.5a).

Regarding facial animation, Motion Graph technique was adopted by Zhang et al. [ZSCS04], named *Face graph* to interpolate frames realistically. Cao et al. [CFKP04, CTFP05] develop an *Anim Graph* to encapsulate speech labeled motion data, based on a greed search algorithm, to provide a real-time speech animation synthesis system. In the same vein, without explicit use of the graph paradigm, many works create speech animation by searching for the optimal combination examples from a database [TMTM12, MCP+06, DN06, LO11, CG00]. Taylor et al. [TMTM12] generate a cluster of dynamic viseme and propose a searching algorithm with both semantic and temporal cost to find the best matching sequence. Finally, graph-based methods can also be used in a texture space to improve the temporal coherence of a facial animation [TZS+16].

Motion Graph enables steering virtual character in application with complex environments, such as slow video games [LCR+02]. Recently, motion graph has also shown to be useful for motion segmentation [VKK14, MC12]. Temporal sequencing of the motion segments has been handled by using state-space search [AF02, ZSCS04], dynamic programming [AFO03], reinforcement learning methods [TLP07, MP07], min-max



(a) Kovar et al. [KGP02] (b) Heck et al. [HG07] (c) Hyun et al. [HLL16]

Figure 2.5 Motion Graph is a popular data-driven technique that gathers motion samples in graph data structure encoding the transition probability between different motion samples. A new motion sequence is produced by searching this graph such as satisfying user constraints. (a) Kovar et al. [KGP02] present a new algorithm that automatically creates a motion graph from a database. This graph encapsulates both the original motion samples and automatically computed smooth transitions. Due to their flexibility and powerful motion controlling capability, many further works have found extensions to reduce the complexity and the computational cost. (b) For interactive control, Heck et al. [HG07] build a graph where each node represents a parametric motion space, and the edges define valid transitions between source and destination parameterized motion spaces. (c) Hyun et al. [HLL16] propose a motions grammars paradigm, whereby motion transitions are enforced through semantic rules.

search [SKY12]. However, the main drawback of this technique is the cost involved in locating candidates, growing exponentially with the number of examples. Due to their flexibility and powerful motion controlling capability, many further works have found extensions to reduce the complexity and the computational cost. These methods include precomputation graph properties [SMM05, LL04], grouping similar data [KPS03, KG04], using physic-based optimization [RZS10], or gathering examples in "hub" according to the occurrence in the database forming a parameterized motion graph [GSKJ08, SO06, HG07, MC12] (see Figure 2.5b) or by semantic connections [HLL16]. Hyun et al. [HLL16] propose a motions grammars paradigm, whereby motion transitions are enforced through semantic rules, as illustrated in Figure 2.5c. The motion graph has later shifted toward motion planning for motion interactive applications in highly constrained and dynamically challenging environments [LLKP11, KBG+13].

Motion graph is a relevant and widespread technique for motion editing in the games industry as it combines the realism of motion capture and the flexibility of graph algorithms. Nevertheless, efficient candidate searching strongly relies on generating a precise descriptive label, which is still produced by hand. Despite improvements to automate the graph construction [AFO03, LZWM05], the resulting computed graph gets a low responsiveness of the character affecting the efficiency of the editing and controlling process. Properly determinate the unit of motion, its length, boundaries, and indexing in the database remains a challenging problem.

Alternatively, Lee an colleagues [LWB+10] use reinforcement learning to propose the *motion field* representation, which enables the user to control the character operating in a continuous space. While a short window increases the "controllability" power of samples-based algorithms, enabling more flexibility and enlarging the complexity of generated motions, it also strongly impacts their computational cost. Aristidou et al. [ACOH+19] define a method to extract *motion motifs* and *motion signature* to represent motion samples using a bag-of-word model [KN14, LHP+17] but fail at creating an universal representation that encompasses every motion. With the same goal, a recent research direction is *Motion Matching*, which was first presented by Büttner and Clavet [But15, HHC+19, BCHF19, HKPP20] and quickly deployed in the game industry. While still requiring a large database of motion capture sample and looking at minimizing a "movement matching cost", this new technique efficiently produces an animation encompassing specific body motion properties specifying by the users, by automatically selecting the best fitting match.

While some motion patterns are easy to identify in facial motion, based on semantic indexing (dynamic viseme, blink), the labeling and the motion segmentation still require manual labor to ensure the quality of the synthesized animation. In addition, it imposes a high memory usage as it needs retaining the whole graph for inference: a balance has to be reached between expressivity, which can be obtained by a graph with a large number of connections (and therefore a greater memory cost), and physical plausibility, which is better enforced with a sparser graph featuring only consistent transitions. Thus, researchers have turned interest toward approaches handling larger databases with fewer manual work.

2.2.2 Learning Techniques

Instead of relying on manual motion correction, the task of resynthesizing animation can be simplified by learning the natural dynamics of motions from data. Hence, a number of studies turn toward learning models of the facial dynamics offering a relevant flexibility for motion editing. In the literature, we observe three approaches to modeling high-dimensional time series data properly: low-dimensional spaces, dynamical systems, and neural-based methods.

2.2.2.1 Low-Dimensional Space

The major issue with motion data is "the curse of the dimensionality". Relevantly and accurately covering the space of realistic motions implies a growing number of samples with respect to the size of the database. One hugely widespread strategy consists of deducing a low-dimensional data representation. Indeed, dimension reduction is an effective approach that has been extensively adopted to improve accuracy and performance in various algorithms [AM00]. A motion sample can be represented in a reduced dimension space, which can be created in a linear or non linear fashion. We outline the two strategies in this part.

Linear Subspace Early statistical approaches rely on PCA [Jol86] to analyze motion data [Bow00] to model complex temporal problems such style motion synthesis [BH00] and temporal motion editing propagation [ZLH03]. Chai et al. [CXH03] take advantage of local PCA, along with motion capture, to produce high-quality facial animation from 2D vision-based control parameters. Later, they leverage PCA to generate full-body motion from a latent variable signal lying in a low-dimensional subspace [CH05]. The first group of methods derive from data, a subspace of realistic motion, and perform trajectory optimization, ensuring plausible motion generations such as speech animation [EGP02, KMT03, SHP04, MCC09]. The continuous property of this manifold guarantees the synthesis of coherent temporal facial animation. Akhter et al. [ASK+12] embed both the spatial and the temporal dimensions of facial motion in a bilinear subspace. The user constraints can be incorporated into an optimization system which aims at solving for the optimal model parameters. This framework ensures that both the spatial and temporal correlation of the edited motion are preserved. Because the manifold of facial motion is nonlinear, several works investigate nonlinear dimension reduction techniques for a better representation of the data points.

NonLinear Subspace Locally Linear Embedding (LLE) [WHL+04, RS00] is a geometrical-motivated technique that assumes locally smooth manifolds with respect to sufficiently small neighborhoods. Wang and colleagues [WHL+04] use LLE and manifold alignment to generate new *expression style* facial motions. Isomap [TDSL00] and its variants [ST03, JM04] take into account the similarity between pairs of data and allow a better perceptual representation of projected data [Ple03]. Deng et al. [DN06] propose a *Phoneme Isomaps* enabling the user to interactively browse and choose motion examples. These nonlinear dimension reduction algorithms hypothesize densely sampled data to create the manifold, which is rarely the case for animation data.

Gaussian Process A nonlinear generalization of PCA is Gaussian Process Latent Variable Model (GPLVM) capable of modeling nonlinear data distribution [GMHP04, Law04, YL10]. GPLVM derives low-dimensional latent variables referring to motion data, and learns the nonlinear mapping from observations (the users constraints), by optimizing latent variables. Levine and colleagues [LWH+12] build a statistical model based on GPLVM for interactive continuous character control. The low dimensional embedding associated with innovative connectivity priors enables generating new motions, with unseen transitions, without leaving the space of realistic motions. Inspired by space-time body motion editing [Gle97], Ma et al. [MLD09] learn the editing style on few frames through a constraint-based Gaussian Process, and then utilize it to edit similar frames in the sequence. Their system is efficient at the time-consuming task of animation editing, but it does not ensure the temporal consistency of the motion.

GPLVM has been widely exploited in animation to derive a new motion by trajectory optimization in the latent space for human motion generations [WMC11], to learn a shared latent space between motion capture and various character representations leading to a high-quality retargeting [BP14]. Compared to the linear dimension reduction algorithm, GPLVM allows lower latent space dimensions without degrading the quality of the reconstructed animation, requiring less computation and a better generalization from small datasets.

Linear models of facial motion, such as PCA cannot encapsulate the full range of motions. On the other hand, nonlinear methods such as GPLVM better reflect the complexity of motion, but for a heavy memory cost, as the training dataset has to be retained. Indeed, a trade-off has to be found between the ability to represent various kinds of motion and the training complexity (the scalability of GPLVM is cubic with respect to the number of examples) [TH09]. Performing the optimization on the latent variable offers the attractive property of preventing exploring unnatural configurations, which often leads to displaying visually unrealistic dynamics, such as mapping the walking motion to the mean pose [WFH08]. However, such methods assume that data are generated independently and preclude a temporal consistency among the latent space.

2.2.2.2 Dynamical Statistic Models

Another statistical technique to model time-series motion data is dynamical systems, where a function describes the behavior of points extracted from a manifold across time. A new motion sequence is generated by deriving the most likely motion, given user constraints.

Hidden Markov Model Early approaches describe the dynamics of motions using Hidden Markov Models (HMM) [RJ86]. HMM have been extensively used to model coarticulation in speech animation since pioneer work of Brand [Bra99, TKMK99, CM93, GBBB06] (see [MV15]). Anderson et al. [ASWC13] extent HMMs-based system by adding a cluster adaptive training, enabling expressive visual text-to-speech. However, HMM models tend to be under articulated, leading to oversmooth motion. This limit might be due to the limited number of hidden states, reducing the motion's naturalness.

Linear Dynamical Models Dynamical systems are very relevant for motion data processing due to their capability for modeling the temporal evolution of the signal, by learning transition between time steps. The simplest form is the Linear Dynamical System (LDS) that presumes linear models for both the mapping function and the dynamical process [DM08]. Chai et al. [CH07] generate new motion satisfying space-time constraints, by extracting low-dimensional latent variable (PCA coefficients)

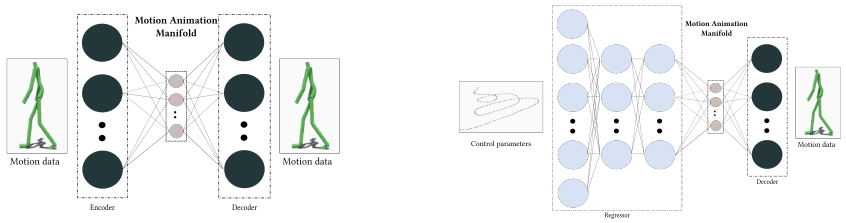
from motion data and modeling motion priors using LDS. Li and colleagues [LMPF10] adapt the LDS algorithm and add bone constraints to handle occlusion filling.

NonLinear Dynamical Models Yet, it has been demonstrated that the complex temporal signature of motions is far from a simple linear pattern [SBS10, BS09]. Stoiber et al. [SBS10] demonstrate that the short-term facial behavior is more precisely approximated with a Hammerstein model than with a linear one. Hence, nonlinear dynamical systems have been studied to increase the expressiveness of the generated motions. Switching Linear Dynamical System hypothesizes locally linear motion behavior, applying LDS in piecewise fashion. The full sequence is fragmented into motion samples, modeled as a set of switching states [Bis05]. Although SLDS has proven to improve linear systems for human motion synthesis [PRM01, LP02], precisely quantify the number of states is not trivial, combined with a large number of parameters to model the posterior distribution lead to an intractable inference. A large body of literature suggests improvements to overcome this issue by approximating the posterior distribution using sampling methods [ORBD05, KEG16]. Englebienne et al. [ECR08] investigate talking face generation suggesting a simplified SLDS model, parameterized with phoneme labels and sound features. The SLDS model appears more tractable with a simplified covariance structure and performs better than HMM for this task. Wang et al. [WFH08] combine GPLVM and dynamical systems, and propose a *Gaussian Process Dynamic Model* that explicitly learn the temporal connection between frames. This model was used to model speech-driven facial animation [DHG13].

Overall, the main difficulty of dynamical models is to properly determine the underlying nonlinear rules governing the temporal features of motions. Indeed, dynamical systems tend to overfit a particular type of motion or require hand-tuning parameters. Another major shortcoming of these methods is their lack of scalability required to deal with the high-dimensional nature, the complex dynamics, and the long-range dependencies of motion data. For these reasons, researchers have stepped toward more sophisticated learning algorithms: Neural Network.

2.2.2.3 Neural-based Solutions

Controllable motion generation The first one to propose a fully learning-based human motion editing system is the seminal work of Holden et al. [HSK16] (see Figure 2.6). They map high-level control parameters to a learned body motion manifold using a fully convolutional structure presented formerly by the same authors [HSKJ15]. Navigating in this manifold of body motion allows easy alteration and control of body animations while preserving their plausibility. Later, Holden et al. [HKS17] design a specific phase-functioned neural architecture to steer the locomotive human body from direct keyboard control in real-time. This model consists in disambiguating the input motion signal by specializing network weights in each step of the walk



(a) Auto-encoder. (b) Regressor.

Figure 2.6 The first one to propose a fully learning-based human motion editing system is the seminal work of Holden et al. [HSK16]. They propose to learn the mapping between a set of high-level controllers and animation parameters using a two-step learning strategy. (a) First, an auto-encoder is trained to learn a manifold of realistic body-motion. (b) Then, a regressor maps high-level control parameters to the motion manifold. Thereafter, the pre-trained decoder enables producing realistic animation from the control parameter code.

cycle, assuming a periodic motion signal. This technique was taken up and later improved to model quadruped motions [ZSKS18], or to guide a character in an interactive environment [SZKS19] through a sophisticated gating network. Although the later extensions are less restrictive, they also hypothesize the modal nature of the motion. While this assumption is appropriate for the body motion that can be segmented into activities, it is less straightforward for facial motion. Recurrent Neural Network have also been explored to perform densely constrained motion synthesis [HHS+17, MBR17, LLL19] due to its attractive internal memory states, that enables context-aware motion synthesis.

Motion Prediction Alternatively, many works propose instead to control the motion synthesis given previous contextual frames. One of the earliest works to investigate a neural network approach for motion prediction is Taylor et al. [THR07]. They explore several variants of Conditional Restricted Bolzmann Machine (CRBM) for human motion modeling [THR07, THR07, THR11]. CRBM is a kind of recurrent network which estimates the next frames from previous hidden units, with explicit autoregressive connections between frames. This new paradigm successfully increases the quality of the generated motion and allows the manipulation of the style of the motion. Zhao and colleagues [ZJS15] suggest Factored Conditional Restricted Bolzmann Machine (FCRBM) [TH09] to generate emotional facial animation. Given an emotion label and initial Facial Action Parameter frames, the trained FCRBF generates the corresponding animation parameters sequence using the Gibbs sampling. Although, RBF-based techniques enhance the data representation capability that may be learned from data, the tendency to generate noisy and floating motion data remains [TH09, SHT09]. Besides, the training procedure still rests on sampling for inference.

As this task implies accurate knowledge on the motion dynamics, many studies have naturally looked toward RNN methods to generate motion based on past context, motivated by its relevant internal memory of the temporal dynamic. Fragkiadaki et al. [FLFM15] predict the dynamics of human motions through two architectures: an Encoder-Recurrent Decoder (ERD) and a 3 LSTM layers-based network (LSTM-3LR). Jain et al. [JZSS16] create a structural RNN to perform the same task by mixing high-level spatio-temporal graph with the efficient sequence learning of RNN. Both propose to handle long-term horizon forecasting by gradually adding noise to the input during the training. The noise scheduling enables their system to produce plausible motions far into the future, before flattening into constant motion. However, this kind of curriculum learning is hard to implement accurately. Crnkovic-Friis et al. [CFCF16] generate dance choreography with an overall style and a composition consistency using RNN. Martinez et al. [MBR17] introduce a residual sequence-to-sequence architecture to predict short-term motion. While demonstrating state-of-the-art short term motion prediction, they fail at generating long sequences. Long-term motion prediction remains a major issue in motion prediction. Many works propose to solve this issue by incorporating additional hints to the network, including derivative information [GMK+19], action label supervisions. [BBKK17], objects in a scene [CPAMN20], geometrical constraints [GWLM18] or by using more sophisticated recurrent architecture [LWJ+19, CAW+19, TMLZ18, GSAH17, LZX+18].

More recent works combine recurrent network with generative models [WCX19, HHS+17, BKL17, KML18, GWLM18]. Wang et al. [WCX19] stack a "refiner" neural network over the RNN-based generator, trained in an adversarial fashion to enhance the realism of the generated motion sequence, while Habibie et al. [HHS+17] sample new motion using the variational autoencoder paradigm. Ruiz et al. [RGMN19] forecast long-term body motions using a CNN-based architecture and novel metrics that reflect better motion capture temporal patterns. Generative learning [GPAM+14] has also been extensively used for motion marker recovery [KBK18, CSLK19, MLCC17, PHMP19].

While a consensus about the temporal motion control parameters has emerged in the human body community, there is no such explicit and standard controllable skeleton in facial animation. A group of works have explored other temporal modalities to synthesize facial animations.

Temporally Dense Facial Motion Signal Constrain As building a model that successfully generates realistic facial motion is highly challenging, many works propose alternatives. One group of works proposes to edit a (2D) facial animation using dense motion control signals, whether they are video signals, dense animation curves, or semantic controls, such as those used in facial reenactment.

Facial reenactment consists in driving the facial performance in an existing video

by one from another source and recomposing a new realistic animation. The works of Thies et al. [TZN+15] and Garrido et al. [GVR+14] have paved the way for facial appearance transfer: Video facial reenactment has been a very fruitful area of research [KEZ+19, KTC+18, FTZ+19, TZN+15, TZS+16, GVR+14, SSKS17, VBPP05] for the last decades. The most common workflow, presented in Figure 2.7, consists in fitting a 3D parametric model [VBPP05, CWZ+14] to an input RGB video [TZS+16, KTC+18, KEZ+19, FTZ+19, DSJ+11] and then, rendering the target video by replacing the expression parameters [TZN+15, TZS+16, KTC+18], with eventually the pose [KTC+18] of the target video with the source ones. Early works rest on traditional computer graphic techniques and adopt an energy optimization strategy to carry out the motion transfer [TZS+16, GVS+15, DSJ+11]. Since a few years, a rapid rise of neural-based techniques has occurred in this field, overcoming most tracking, person-specific model, and 3D reconstruction inaccuracies issues. NN has been adopted to either map 3D facial parameters to photorealistic images animation [KEZ+19, NSX+19], to map source video to the target one [WKZ+18], or to improve the quality of the target video rendering [FTZ+19, KTC+18, SWQ+20]. Pushing forward the limits of facial reenactments, neural network has allowed further high-level control on the generated animation, such as the head pose control [KTC+18, NSX+19, ZSBL19, WKZ+18, YYZ+20, ZPW+20] and the eyes [KTC+18, NSX+19] and a better generalization over multiple identities [ZSBL19, NSX+19, ZSBL19], yielding very impressive results. Facial reenactment also involves many works that rely on 2D image warping [AECOKC17, SSKS17] leveraging state-of-the-art image-to-image translation techniques [IZZE17, KTC+18]. However, in this thesis, we focus on animation, not on rendering, and a complete review of facial image synthesis is out of the scope of this thesis.

One instance of facial reenactment is Visual Dubbing. This task consists of controlling the lips in a target video with a new input, given audio track reciting by another subject, with eventually a foreign language [SSKS17, BCS97, GVS+15, CE05]. Accurate synchronization between the lips movements and the speech is mandatory [SP54, OCIM07] to create realistic animation. Indeed, inaccurate alignment between the speech and the visual lips motion can strongly alter the comprehension of the sentences: the *Mcgurk Effect* [MM76] occurs when the pronounced sound is associated with the visual movement of another sound, creating the illusion to hear another sound. To overcome this pitfall and consistently modifying the lips motion, several works rest on facial reenactment transfer methods. Others produce a new target animation by deriving a speech/audio to lips motion mapping. Neural networks have emerged as the leading way to build a consistent and generalizable mapping. A new video, representing a target subject animated with a desired speech-based lip movement, can then be generated from a target audio track only [TET+19], or by combining a target audio track with a source video [SWQ+20] or a source still image [ZLH+20]. Nonetheless, whereas facial reenactment methods have achieved a high degree of performance, they are devoted to a particular (2D) facial animation editing scenario, in which either a semantic or source animation is available, prevent-

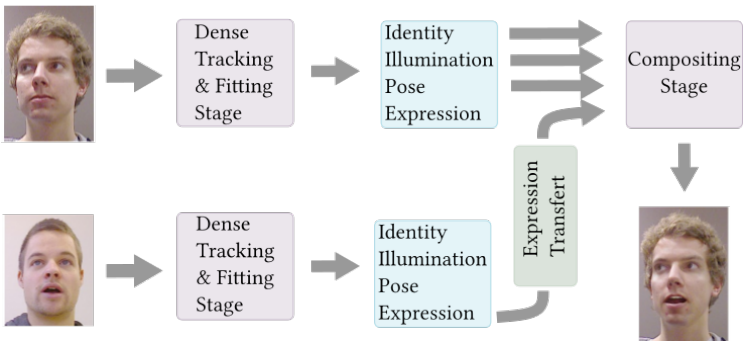


Figure 2.7 Many works propose alternatives to edit a (2D) facial animation using dense motion control signals, whether they are video signals, dense animation curves, or semantic controls, such as those used in facial reenactment. Facial reenactment consists in driving the facial performance in an existing video by one from another source and recomposing a new realistic animation. Early work in this direction, Thies et al. [TZN+15] propose a workflow that has been taken up by numerous following works. This workflow consists in fitting a 3D parametric model to an input RGB video and then, rendering the target video by replacing the expression parameters.

ing flexible and creative editing applications.

In the same line, another group of works synthesizes the whole animation based on speech or text. Although, speech and text driven animation has stimulated the research for a long time [BCS97, BBEO03], neural network has revolutionized the field, allowing high quality facial animation synthesis [TKY+17, ZXL+18], video [SSKS17], or an audio stream [KAL+17]. Speech-driven animation is an extremely popular subject; we provide here only the most relevant works. An extensive review can be found [Hti17]. One major work by Taylor and colleagues [TKY+17] show an innovative system that automatically generates very natural looking talking facial animation. Their approach, based on fully connected mapping a sliding window [KYTM15] of phonemes to animation parameters, has engendered further speech-driven animation studies. The main scheme consists of associating a sound unit, a phoneme, to their 3D visual counterpart, which is either a face rig [ELFS16, ZXL+18], a mesh [KAL+17] or the blendshape weights [PCP17]. Recent studies turn toward generating photorealistic 2D portraits, using a joint embedding trained by a CNN auto-encoder [JCZ19, DBR20], image-to-image translation technique [KTC+18], or a recurrent Generative Adversarial Network (GAN) architecture [VPP19, VPP18, ZLL+19, KEZ+19]. Suwajanakorn et al. [SSKS17] synthesize Obama's speaking videos from an audio track leveraging the number of available data on Obama to train a recurrent network. Outstanding photorealistic results are obtained by relying on 3D parametric models [BV99, BBPV03] and neural-based rendering network [KTC+18, KEZ+19, SWQ+20]. Fried and colleagues [FTZ+19] propose a new workflow to edit a video by modifying the associated transcript. The system automatically regenerates the corresponding altered viseme sequence using a

two-stage method: a coarse sequence is generated by searching similar visemes in the video and stitching them together. Then, a high-quality photorealistic video is synthesized using a recurrent neural network. The quality of the generated animation is improved by feeding the network with a still image [JCZ19, VPP18].

Discussion Both dynamical or low-dimensional approaches show failures at properly tackling the temporal dependency of motions. Long-term dependencies require more data from the past and the future, or an increased memory of the model to handle further past time. Most of the time, handling long-term dependencies implies oversize inputs leading to intractable models or an overload of the model's memory. Besides, the length of the motion dependency is often unknown, expending the difficulty of accurately coping with temporal motion sequences. Neural Networks have shown promising results in efficiently modeling and generating motion data, displaying the realism of the animation while allowing the flexibility of modeling approaches. Theoretically, NN is devoted to learning from data the intrinsic dependencies, permitting reproducing the dynamics of the motions faithfully. Overall, state-of-the-art results in both human motion editing and 3D facial animation synthesis and 2D facial animation manipulation have been derived using neural-based techniques [TFT+20].

That is why, in this thesis we explore different neural-based approaches to model respectfully the dynamics of facial animation and improve the facial animation editing workflow. The attractive properties of neural networks to provide both the flexibility and the responsivity of linear models, and the powerful capacity to model complex temporal data of nonlinear ones have been demonstrated throughout our works. Falling within the general trend, we demonstrate a breadth of edits by exploiting recurrent GAN architecture [KEZ+19, SWQ+20] in Chapter 6. We produce realistic facial animations matching semantic constraints, presenting a step forward facial animation editing and cleaning which is important in vfx industry.

Although the notion of controllable motion has been explored in the field of body motion, it remains an open question regarding facial animation. Indeed, while an intuitive and high-level parametrization steering a body motion have generated a consensus, there is no such standard abstraction to guide facial motion. Moreover, in this thesis, we demonstrate that the inherent difference between the dynamics of the body and the face, precludes applying actual body-motion systems to facial dynamics. Thus, following works in controllable motion generation, we propose in Chapter 5 a learning-based method to perform temporal animation editing, given meaningful temporal vertex distances.

2.3 Animation Processing

While the previous sections focus on explicit constraint-based animation editing, this section delves into non-constraint-based motion editing. Much like constraint-based

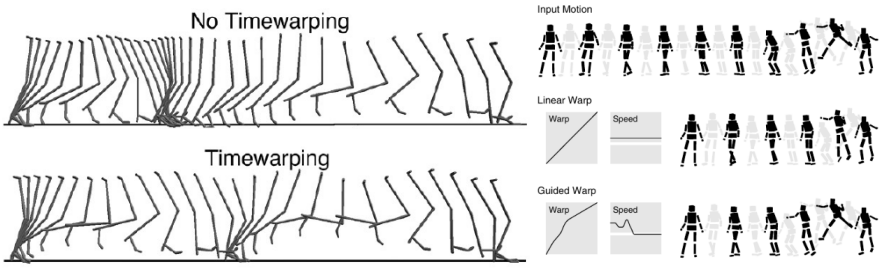
approaches, these methods aim to modify the motion sequence (hence, it is an editing operation) while preserving some original features. In that case, the editing process is not explicitly described as modifying features in motion, but rather as processing animation parameters used to represent the motion. These methods encompass motion signal techniques such as motion retiming and motion filtering or operations.

2.3.1 Motion Warping

One kind of editing operation that an animator is often brought to perform is time warping, that is, adjusting the timing of an animation. For instance, when motion samples are concatenated or blended, they should be dynamically aligned and structurally similar. Indeed, in this case, no timewarped animation might display unrealistic motion as you can see in Figure 2.8a. Time warping is, therefore, an editing procedure to temporally reshape the motion signal without losing the content and style properties [HdSP07]. Time-warping has been exploited in speech signal processing to analyze akin utterances or to align lips motion and the speech [BCS97, CFKP04]. Overall, it generally determines the optimal sample correspondences between signals and applies a series of compression and extension operations, to optimally "warp" signals [BW95]. Common techniques to align two motion signals are Dynamic Time Warping (DTW) [BW95] and curve alignment [GS05, WP95]. Witking and Popovic [WP95] introduce the term in 1995 and choose Cardinal splines as timewarping functions to generate a mapping between two motion signals respecting a set of sparse keyframes. Adjusting the timing of animated character often requires significant manual intervention. Hsu et al. [HdSP07] develop a new technique to guide this process more efficiently through a set of keytimes and reference motion examples, while ensuring the production of natural animated character. Even recently, DTW was applied to synthesize talking animation to mimic the speaker dynamics [SSKS17]. Motion retiming is a powerful method to edit motion sequences. Even so, it remains at a low level of abstraction, requiring animators to describe edits in a one-dimensional space. Recently, many studies turn toward innovative frameworks to provide flexible and intuitive interactive interface to perform motion retiming [GRGC15, YAMZ+15, CiRL+16, CGNS17]. However, such abstractions are devoted to manipulating body motions.

2.3.2 Animation Filtering

In facial animation, real-time motion capture based animation has demonstrated high-quality results [WBLP11, LYYB13, CHZ14]. Yet, it is frequent that the animation has to be cleaned to remove the noise, coming from the capture setup, or the failures of the tracking software. Indeed, most of the time, the final animation is not the raw mocap-based signal but rather a processed version, which has been roughly smoothed. In this section, we review the main animation filtering techniques.



(a) Kovar et al. [KG03]. (b) Hsu et al. [HdSP07]

Figure 2.8 One widespread editing operation that an animator is often brought to perform is time warping, that is, adjusting the timing of an animation. (a) Motion blending or concatenation often fail when two samples have different timing. Kovar et al. [KG03] shows that stitching together a walking and jogging sample spanning two locomotive cycles with different timing results in a non-realistic right leg motion. As out-of-phase frames are combined, the character floats in the air with a leg almost straight. (b) Automatic motion blending might lead to unrealistic motion, such as quick jump (middle). Hsu et al. [HdSP07] propose a new approach to guide the timing alignment process more efficiently, while ensuring realistic motion generation.

Signal Processing Methods Motion filtering has been a long-term research topic. Early works applied standard signal processing model algorithms [BW95] such as Kalman Filter [SLSG01, LMPF10], wavelet transformation [HK08], linear time invariant filters [LS02, YN03] or exponential smoothing. For instance, the majority of performance-driven facial animation systems based on depth sensors [WBLP11, HMYL15] remove high-frequency jitters with a temporal filter with exponential adaptive weights. Some works further improve the temporal coherence of tracking by enforcing an animation prior [CWLZ13]. In the same vein, Cao et al. [CHZ14] penalize the magnitude of temporal derivatives of the output animation. Garrido et al. [GVWT13], as well as Valgaerts et al. [VWB+12], use structure-aware regularization to strengthen optical flow estimation. While producing smooth results, their final animations are not free from artifacts, notably the loss of high-frequency motions in the eyes and mouth. Indeed, these post-processing techniques are not aware of the motion nature, and thereby, tend to lose dynamic cues of the original facial motion.

Modeling the dynamics of the face is highly challenging due to the complexity and the non-linear nature of facial motions. Methods using linear observation models such as the Kalman filter [WKT06, Bre97, HDF12] appear insufficient for facial motions. More complex motion filtering systems have been explored fusing particle filter with Markov Chain [KF01] or belief propagation [LNLN06]. Huang et al. [HDF12] combine AAM models with a Kalman filter to create a more robust face tracking. Another trend among the motion filtering area is extended non-linear approaches and associate particle filtering with local optimization [DDFG01, BKMVG07] or global one [GRBS10],

but these are often hard to tune and too memory expensive for long sequences.

Data-driven Methods To capture the underlying temporal dynamics, many works leverage the availability of motion data to perform motion refinement. One investigated strategy to remove noise is to applied dimension reduction methods (PCA) on the motion samples, removing non-informative signal components using PCA [ASK+12, TS06, WCP10, GF16] or by learning an autoencoder manifold [VLL+10, HSK16, LZZ+19, LZZL20]. Many works propose to filter motion through prior-based strategies, modeling motion as either dynamic process, low-dimensional Gaussian Processes [WFH08, UFG+08], a bilinear spatiotemporal model [ASK+12], Markov models [LGN14] or as binary latent variables [THR07].

Discussion Performance-based motion signals are high dimensional and often contain noise. Signal processing approaches are often easy to implement and efficient for short-term motion cleaning. However, these methods have several downsides. In these approaches, some valuable information might be lost, or it demands a painstaking parameter tuning to filter every facial motion signal accurately. Additionally, these techniques are applied to each motion signal independently, ignoring the correlation among the different animation parameters. Most of these studies are designed to be non-causal, limiting their applications to offline processing. Early signal processing techniques rest on a stationary hypothesis and a fixed and specific framerate, which is far from the facial animation dynamic process. Recently, several studies turn toward adaptive filters for each parameter demonstrating successful results in cleaning any kind of actions, utilizing neural networks [MLCC17, FLFM15, BBKK17], dictionary learning [XFJ+15, FJX+15, WLQ+16] or optimization framework [HJ10]. Even so, these filters are applied to the motion signals afterward, precluding a real-time application. That is why in this thesis, we proposed a recurrent neural network system and an original parametrization to clean and refine an inaccurate motion signal that overcomes these issues.

2.4 Conclusion

The facial animation editing task encompasses all the acts of modifying the motion content. As exposed in this chapter, we found a wide literature, dealing with many aspects of animation post-processing. Especially in body animation synthesis, the field of motion editing has a long history. Yet, the stakes of the facial animation editing are different; Most of the actual learning-based methods operate on body-motion using a well-standardized skeleton representation, developing innovative training strategies relying upon this animation parametrization. As far as facial animation is concerned, there is not such a consensus on the facial motion representation, and no benchmark database exists. Overall, this thesis lies alongside state-of-the-art works

and builds upon previous knowledge, trying to push and evaluate new ideas in the field of facial animation editing.

CHAPTER 3

3D Facial Animation Dataset

Contents

As observed in the previous chapter, learning-based methods have led to tremendous progress in the computing field, paving the way for more perspective in motion manipulation and synthesis. Considering this, we aim at exploring current promising learning-based techniques to revise the animation editing process. Yet, machine learning breakthroughs have been made possible through the rise of synthetic media content and, thus, the availability of large databases.

As far as facial animation is concerned, a rising number of static facial expressions [CWZ+14, SAD+08, RBSB18, ZWL+13] and dynamic expression unit corpus [CKH11, ZYC+13, CVTV05] have promoted constant improvements in still facial modeling [LSSS18]

and facial animation rendering [WSS+19][1]. Despite the explosive growth of 3D contents, only a few 3D animation databases are suitable for our applications. Indeed, most of the existing databases integrating temporal information are either too small or provide animation sequences with limited variety of temporal information (only unit expressions) to learn the natural dynamics of facial motions, precluding a thoroughgoing facial dynamic investigation. The lack of diversified and high-quality datasets is still an open challenge, as recently observed by Egger et al. [EST+19]. In this thesis, we investigate the different facets of facial animation editing: animation manipulation and various forms of facial motion controlling. For instance, one line of research explored in this work is the multimodal nature of facial animation, the relation between animation and speech or text information. This implies a broad diversity among the data and relevant annotated metadata. Yet, most released facial datasets aim at capturing emotional expressions, and few one provide animation caused by speech.

To bypass these deficiencies and meet our needs, we use two different databases: we take advantage of the recent motion capture technologies to produce satisfying new data and extend an available dataset to leverage its multimodal metadata.

In this chapter, we briefly expose the main principles of neural networks in Section 3.1 motivating the use of a large and high quality database. Then, we present the two datasets used in all our experiments, that include different valuable properties. On the one hand, we gather performance-based animation sequences coming from real-time tracking software [Dyn19a]. This animation data presented in Section 3.2, is akin to ones that we might find in the production industry without post-processing and data priors. Hence, this corpus relevantly reflects the real data prone to animation post-processing editing.

On another hand, we target semantic motion controlling, and therefore we covet a multimodal (audio/text-visual) database. Previous literature has put forward the 3D Audio-Visual Corpus of Affective Communication (B3D(AC)ˆ2) dataset released by Fanelli and colleagues [FGR+10]. This corpus appears as an appropriate database for our purpose as it is a large database of 3D geometry data, representing dynamic facial animation produced from the capture of 14 actors reciting *long* sentences. Moreover, it is supplied with valuable annotated metadata. In Section 3.3, we present this corpus and its numerous advantages that have motivated our choice to use this data for our experiments.

Yet, this dataset carries two major weaknesses: the first issue with the B3D(AC)ˆ2 dataset is the low quality of the 3D geometry data, preventing us from having a good depiction of subtle mouth closures and blinking. Those are, however, crucial to verbal and non-verbal communicative cues that facial animations convey. The second concern relates to the animation parametrization formulation of the data. The amount of ambiguity in this 3D geometry data, which entangles morphology and expressive information, makes the inference task difficult. Besides, the 3D data

[1] A comprehensive survey of the existing 3D facial database can be found in [EST+19]

is supplied with raw, dense meshes, which cannot be directly fed to neural networks as it would imply a high memory cost to process the high resolution 3D structures overtime. Yet, extending neural network architecture to the non-Euclidean domain is not trivial and belongs to another research area [RBSB18], which is outside the scope of this work.

To provide solutions compliant with the traditional facial animation pipeline, we want facial expressions to be encoded through the standard blendshape parametrization, which is the predominant model used by animators to create realistic facial animation. Indeed, the blendshape paradigm offers several advantages [LAR+14], including an intuitive and semantic meaning of the parameters. Each blendshape corresponds to sculpted and localized deformations of the neutral face mesh, which can be more or less emphasized by activating blendshape weights. The defined set of blendshapes, thereby, covers the range of possible facial expressions, enabling artists to easily manipulate the face's mesh with a low computational cost. These compelling reasons have motivated our choice to use this parametrization. In Section 3.3.3, we present the processing work to modify and extend this database, addressing the above issues, and meeting our requirements.

3.1 Database Motivation

As introduced in Chapter 1, we aim to handle the complex nature of facial motions using neural-based approaches. In this section, we briefly introduce the main concept of statistical learning in Section 3.1.1. Then, we present the more complex notion of neural networks in Section 3.1.2 and put forward the need for large datasets in Section 3.1.3.

3.1.1 Statistical Learning

The main objective of statistical learning techniques is to learn an approximation function $f_{\boldsymbol{w}}$, characterized by a set of parameters $\boldsymbol{w}$, minimizing a loss $\mathcal{L}$ with respect to a set of data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. This dataset gathers pairs of input data/target annotation or input data/target data depending on the objective task. The goal of statistical learning is to find the optimal $\hat{\boldsymbol{w}}, \hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{J}(\boldsymbol{w})$, with $\mathcal{J}$ the full objective function gathering the empirical loss $\mathcal{L}$ and a regularization penalty $\mathcal{R}$.

The optimal $\hat{\boldsymbol{w}}$ is either computed through a closed-form solution in case of linear regression (using the least squares framework) or exploiting iterative gradient descent algorithms if there is no tractable solution.

3.1.2 Neural Network Principle

In the most basic form of feedforward network, neural networks consist of progressively transforming the input data through a sequence of hidden projection layers and non-linear activation functions. Formally, a feedforward network with l layers is

represented as follows:

$$f_{\boldsymbol{w}} = f_{\boldsymbol{w}_l}(f_{\boldsymbol{w}_{l-1}}(...f_{\boldsymbol{w}_1}(\mathbf{x}))). \tag{3.1}$$

The more layers, the *deeper* the network. The main feature of neural networks is their differential nature, which enables computing the gradient $\frac{\partial \mathcal{J}(\boldsymbol{w})}{\partial \boldsymbol{w}}$ of the objective function with respect to the coefficients $\boldsymbol{w}$ through the efficient backpropagation algorithm [RHW86]. Formally at each iteration, the new parameters are computed as:

$$\boldsymbol{w} = \boldsymbol{w} - \eta \frac{\partial \mathcal{J}(\boldsymbol{w})}{\partial \boldsymbol{w}}, \tag{3.2}$$

with η stands for the learning rate, weighting the gradient direction. Evaluating the gradient on every sample of the dataset is never done in practice due to time and memory issues. To alleviate the *computational burden*, stochastic approximation algorithms of the gradient descent replace the exact gradient optimization. Faster and efficient algorithms such as the Stochastic Gradient Descent (SGD) [Pol64, SMDH13] or the Adam [KB14] iteratively compute the gradient on a batch of data, while preserving a high enough convergence rate.

Despite the fast pace of progress in the field of deep learning, the theory underlying the learnability of neural networks is still an open problem. Although many improvements have been made in neural architecture design, gradient descent optimization, data robustness, parameters initialization, training stability, and convergence speed, the convergence to the global minimum is still not guaranteed due to the neural networks non-convex nature. Deriving the optimal solution in high dimension space remains highly challenging. For this reason, most of the deep learning approaches are adopted from an empirical point of view.

3.1.3 The importance of Machine Learning Data

Given this definition, the dataset $\mathcal{D}$ appears as the backbone of any machine learning algorithms which aim to assimilate high-level representation of such data to infer future prediction on newly seen samples. Having relevant databases is therefore, the crucial ingredient to train ML algorithms and make them comprehend and ingest the realistic and useful subjacent features of the data. In this thesis, we wish to exploit representative databases for the problem's input, covering all the characteristics of samples that might be observed in the real animation pipeline, and with useful metadata to define the training strategy, the problem statement and the task's supervision rightly. Hence, we use two different datasets, exposed below, with different properties related to real editing use cases.

3.2 Performance-based Animation Dataset

In this thesis, we put emphasis on the animation editing using neural-based approaches. As motivated is the previous section, we need a large amount of realistic animation, with plausible dynamic properties. The best way to get this data at a

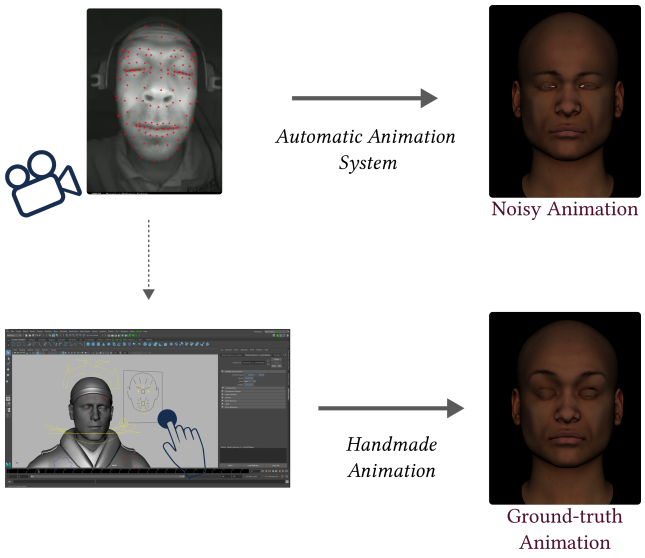


Figure 3.1 Performance-based Animation Dataset Pipeline. From the original video, a noisy animation is automatically generated by a real-time tracking software [Dyn19a] without post-processing. An experimented animator creates the *ground-truth* animation associated with the original video, using a professional software.

good scale is motion capture. We thus use a professional tool to gather as much as realistic-looking animation as we can from a set of prerecorded MoCap-based videos.

In our research, we investigate new methods to clean and refine performance-based animation (see Chapter 4 and Chapter 6). Interestingly, less faithful and less accurate motion tracking can be obtained from the same raw videos through regression techniques for minimal effort. We employ one of these automatic face tracking solutions to generate coarse, noisy animation corresponding to the initial MoCap-based videos. These generated samples contain all the combination of noise and inaccuracies that might appear on a real performance-based animation pipeline, including the captation, the software processing and the retargeting. This feature makes the generated animation data relevant for our purpose, as it avoids simulating an artificial noise process to degrade the animation signals.

An illustration of the different components of this dataset is presented in Figure 3.1. This dataset gathers 56 sequences recorded at different framerates: 30, 60, and 120 frames-per-second (fps), grouping talking animation and continuous successions of dynamic expressions. In total, we gather around 286 000 frames of each paired data, representing around 49 minutes of animation. Due to intellectual property interests, we will not publish this dataset.

Figure 3.2 The B3D(AC)^2 Corpus is composed of 4O sentences reciting by 14 subjects (8 females and 6 males).

3.3 Multimodal Facial Animation datset: B3D(AC)^2 Corpus

3.3.1 Presentation and Motivation

Through the use of professional tools, we gather an animation database of reasonable scale, but metadata is lacking. As we wish to explore editing algorithms based on high-level concepts, at least part of our data should be annotated. Thus, we leverage another database, the B3D(AC)^2 dataset released by Fanelli [FGR+10]. The corpus contains the 3D geometries and the corresponding RGB images of 40 English sentences recited by fourteen native English subjects (eight females and six males), with both neutral and emotional tone. The 14 subjects are shown in Figure 3.2.

The pronounced sentences last on average 4.67s and have been recorded simultaneously with the visual information in a quiet environment (without other voices or background noise), which are attractive properties for audio-visual mapping. Every sequence is provided with the speech signal annotations such as the phonetic transcriptions of the speech, an accurate phoneme segmentation and alignment, and two other speech features (fundamental frequency and the signal intensity estimation).

The 3D dynamic nature of the corpus, combined with the diversity among the subjects of this dataset and the quality of the audio-visual data labeling, have motivated the utilization of this database. The overall corpus amounts for 85 minutes of animation, with an average of 4.67s by sequences. We believe that the public availability of this dataset is a strong asset for research purposes, as it enables the reproducibility of our work.

3.3.2 Shortcomings

While this corpus provides an amount of exiting 3D facial animation data, this corpus suffers from several shortcomings. Some of them are displayed in Figure 3.3. For our purpose, the major drawbacks are listed below:

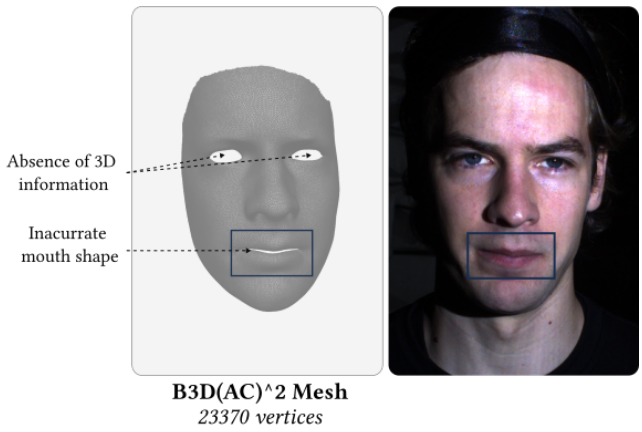


Figure 3.3 Illustration of some shortcomings of the 3D B3D(AC)^2 data. The 3D data provided by the B3D(AC)^2 database [FGR+10] is a dense 3D-vertice mesh with more than 23 000 vertices. This parametrization overtime would overload any current neural network. In addition, the data suffer from some shortcomings (left): for instance, there is no information about the eye movement and the animation of the mouth is occasionally inaccurate.

- **Low Animation Quality**. The supplied 3D data does not carry information about the eyes, eyelids, and inner mouth [FGR+10]. Moreover, the quality of the mouth animation is low due to noisy patterns and inaccuracies in the 3D geometry data.

- **Inappropriate Animation Parametrization**. The 3D animation data, materialized by a mesh of 23 370 vertices, are not in an adequate format. The mesh is overly dense to be directly tractable by a learning model. Moreover, the vertex-level representation entangles different types of information about the pose, the morphology and the expressivity, making the editing task more difficult. It often remains deserved to the drawing of the shape. The standard instrumental technique to efficiently model and edit a facial animation is instead to manipulate higher abstraction of the face's deformations. The *rigged* format, where a set of parameters are pre-defined, remains the prevalent technique, permitting the creation of a wide range of plausible expressions with artistic guidance.

We address the above issues using a two-part process; In the first part, we improve the quality of the 3D data by fitting a standard deformable template, with a sparser mesh, to the neutral geometry of each actor in a coarse-to-fine approach. In the second part, we derive a new data representation by transferring a blendshape model onto the aligned deformable template and computing the blendshape coefficients for each frame. An overview of the database enhancement steps is given in Figure 3.4.

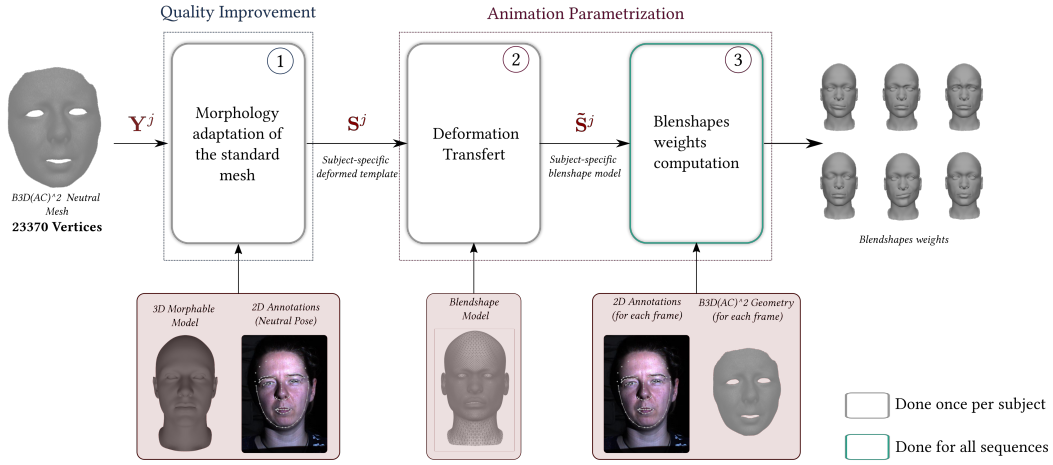


Figure 3.4 B3D(AC)ˆ2 Dataset Extension Workflow. We improve the available B3D(AC)ˆ2 corpus to derive a more accurate and suitable animation parametrization. This pipeline encompasses two parts: the firsts steps aim to derive a novel and more accurate neutral subject's facial model. The last steps lead to the desired and precise representation of this corpus: the blendshape weights.

In the following section, we detail this procedure, that is strongly inspired by previous works [ACP03, ARV07, LAGP09].

3.3.3 Database Extension Process

The pipeline of the database extension can be divided into two parts, presented in Figure 3.4. The first part leads to the construction of 3D animatable model for every subject in order to improve the quality of the 3D data (Section 3.3.3.1); This one-step part is applied only once per subject and consists of the morphology adaptation of each subject's geometry. While, the second part, divided in two steps, intends to deriving a lightweight animation data representation (Section 3.3.3.2). First, one subject-specific blendshape model is created for each subject. Then, a blendshape weights fitting is computed for each shot.

The construction of this facial animation database assumes the availability of a 3D blendshape model sculpted in a template $\mathbf{T}$ represented as a triangle mesh with $\mathbf{v}_i \in \mathbf{T}, i = 1, ..., N$ vertices, combined with a 3D Morphable Model generated using the same template triangle mesh connectivity.

3.3.3.1 Morphology Model Fitting

The B3D(AC)ˆ2 corpus provides faces parameterized as triangle meshes, with $L =$ 23 379 vertices and an identical topology. The pipeline begins with the template mesh fitting and alignment on the neutral mesh of every subject, using a stan-

dard optimization-based framework for 3D facial model registration [ACP03, ARV07, LAGP09, LYYB13]. In what follows, we briefly review the main milestones of this step, see [Li10, LYYB13] for more details.

Let $\mathbf{S}^j = [\mathbf{x}_1^j, ..., \mathbf{x}_N^j]$ be a deformed and aligned version of $\mathbf{T}$, that must fit the neutral face $\mathbf{Y}^j$ of the subject j, also represented as a triangle mesh with $\mathbf{y}_l \in \mathbf{Y}^j, l = 1, .., L$. In the remaining of this section, j will be omitted for clarity.
In a 3D geometrical point of view, the goal is to align and deform the mesh vertices $\mathbf{x}_i \in \mathbf{S}$, to match $\mathbf{Y}$ i.e., to minimize the energy function E_{align} formulated as:

$$E_{align}(\mathbf{S}) = \sum_{i=1}^{N} w_{icp} E_{icp}(\mathbf{S}) + E_{prior}(\mathbf{S}), \tag{3.3}$$

where E_{icp} measures how close $\mathbf{S}$ is from $\mathbf{Y}$, while E_{prior} represents the prior knowledge about the 3D deformable surface, limiting the allowed deformations.

Morphable Model To do this morphology adaptation, we impose a global structure representation to the 3D data and assume the 3D mesh X as a linear model, using a generic 3D Morphable Model (3MM) paradigm, developed by Blanz and Vetter in 1999 [BV99, PKA+09].
A 3D Morphable Model (3DMM) is a parametric model derived from applying a PCA onto a database, expressed as $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mu}, \mathbf{M})$. In this formulation, $\boldsymbol{\mu} \in \mathbb{R}^{3N}$ is the mean vector, and $\mathbf{M} \in \mathbb{R}^{3N \times P-1}$ is the principal components matrix, forming an orthogonal basis. We leverage the PCA statistics derived on the 200 scans such as described in [BV99, PKA+09]. We notice that the standard morphable model gets two independent (texture and shape) components, but here, we just take account of the shape component. A new face can be generated by linearly combining the principal component of the model using a novel coefficient vector $\boldsymbol{\alpha}$. A deformed mesh in this linear parametrization is represented as a flattened 1-dimensional vector expressed as:

$$\mathbf{s} = \boldsymbol{\mu} + \mathbf{M}\boldsymbol{\alpha}. \tag{3.4}$$

Relying on such statistical models as 3DMM drastically reduces the dimensionality of a shape optimization problem, going from N dimensions, the number of vertices in the deformable mesh S, to 200, the length of $\boldsymbol{\alpha}$. It provides a robust framework for an initial coarse 3D template alignment.

Optimization function definition The straightforward method to perform 3D points alignment of different topologies is the Iterative Closest Point algorithm, developed by Besl and McKay [BM92], consisting of iteratively computing point re-matching and solving:

$$\mathbf{S}(t+1) = \arg\min_{\mathbf{S}} \sum_{i=1}^{N} w_{point} ||\mathbf{x}_i - \mathbf{c}_i(t)||^2 + E_{prior}(\mathbf{S}). \tag{3.5}$$

where $\mathbf{c}_i$ is the closest point on the input mesh $\mathbf{Y}$ from $\mathbf{x}_i$.[2] The first term is generally called the *point-to-point* metric. Alternatively, the *point-to-plane* [CM92] term is used for an optimal alignment [3]:

$$\mathbf{S}(t+1) = \arg\min_{\mathbf{S}} w_{point} E_{point}(\mathbf{S}) + \sum_{i=1}^{N} w_{plane} ||\mathbf{n}_i^T (\mathbf{x}_i - \mathbf{c}_i(t))||^2 + E_{prior}(\mathbf{S}), \quad (3.6)$$

where $\mathbf{n}_i$ denotes the surface normal at $\mathbf{c}_i$.
An extensive literature enhancing this algorithm provides efficient implementation algorithms for the 3D registration problem [RL01, PLH04, LYYB13].

Alignment and deformation problems typically rest on prior knowledge of the underlying 3D object included in the second term. These priors limit and specify the allowed deformations of the input mesh subject to geometrical or physical properties. In what follows, we detail the priors used throughout the fitting process.

First, we make some assumption about the rigidity of the deformed mesh. Formally, the 3D mesh, considered as a globally rigid 3D object, can be aligned with the subject's mesh by finding the optimal global rotation $\mathbf{R}$ and translation $\mathbf{t}$ minimizing:

$$E_{rigid} = \sum_{i=1}^{N} w_{rigid} ||(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - \mathbf{c}_i||^2. \quad (3.7)$$

To further improve the fitting, the 3D structures can be locally deformed instead, in a non-rigid framework [ACP03, ARV07, LSP08]. In that case, the per-vertex pose parameters $(\mathbf{R}_i, \mathbf{t}_i)$ for each vertex $\mathbf{x}_i$ are optimized, with strong priors ensuring an accurate 3D model structure [LSP08] and preserving the ICP convergence properties [ARV07].

As we aim to improve the quality of the data, we consider the 2D landmark information derived from the 2D RGB frames. We assume that the projection of some selected 3D vertices onto a 2D image plane should match targeted key image points. Inspired by Li et al. [LYYB13], we improve the quality of the alignment around the mouth and eyelids using 2D image landmark information for each frame, obtained with a commercial face tracking software [Dyn19b] and manual cleaning. The energy prior enforcing the 2D projection of the 3D vertices of the mesh $\mathbf{S}$ to match the K landmark positions delivered by the tracking software [Dyn19b] can be formulated as:

$$E_{2D} = \sum_{k=1}^{P} w_{2D} ||\Pi(\mathbf{x_k}) - \mathbf{u}_k||^2, \quad (3.8)$$

[2] We conserve only closest point pairs meeting some criterion: we prune all point pairs with incompatible normal directions and a distance larger than several millimeters.

[3] Undesirable oscillations may appear if only point-to-plane is considered. Combining the two metrics improves stability [LAGP09].

where $\Pi \in \mathbb{R}^3$ refers to the projection operator, $\mathbf{u}_k = [u_k^x, u_k^y]$ is the 2D landmarks corresponding to the vertex $\mathbf{x}_k \in \mathbf{S}, k = 1, ..., K$.

Finally, for a more accurate and finer fitting, we leverage surface-based deformation techniques, which operate directly on the 3D mesh. Supposing the initial triangle mesh $\mathbf{S}$, the optimal deformed mesh vertices $\tilde{\mathbf{x}} \in \tilde{\mathbf{S}}$, is commonly derived by minimizing a point-to-point energy. Yet, surface-base deformation approaches strongly rely on regularization to yield to attractive mesh, with desired surface properties. The prominent Laplacian surface-based regularization [BS08] is an established 3D prior which enables local modifications while preserving local curvatures as much as possible.

Global-to-local Fitting Strategy The first part of the database extension workflow consists in a coarse-to-fine strategy which involves three optimizations, resting upon prior knowledge. These priors go from the global face representation to the local curvature of the face's surface.

Coarse Model Fitting:
We coarsely align and deform the source mesh $\mathbf{S}$, using a set of precise and salient points correspondences, wherein the deformed surface is parameterized with the morphable model (Equation (3.4)). We employ the Gauss-Newton algorithm, assuming a rigid 3D object (Equation (3.7)) to infer the optimal rigid pose parameters $(\mathbf{R}, \mathbf{t})$ and $\boldsymbol{\alpha}$ coefficients vectors, using both point-point and point-to-pane metrics.

Local Model Fitting with 2D Landmarks Prior:
We improve the deformed mesh by adding some details that are not present in the B3D(AC)^2 meshes, using information from the 2D landmarks. We use the non-rigid ICP algorithm, with a 2D point prior (Equation (3.8)), still optimizing for the pose parameters and the $\boldsymbol{\alpha}$ vectors [LYYB13].

Local Mesh Refinement with Laplacian Prior:
Finally, we further refine the deformed model by adding local details that were absent in the morphable model. In order to do this, we directly optimize the vertices $x_i \in \mathbf{S}$ by leveraging a Laplacian deformation algorithm [BS08].

At this point, we obtain the 3D registration of the 14 subject's neutral faces, represented in our template structure. The next part of this process is about deriving a blendshape model for each subject and the corresponding blendshape weights sequences.

3.3.3.2 Personalized Blendshape Model Adaptation and Fitting

In the second part of the database extension pipeline, we aim to express the subjects' facial expressions over time in terms of blendshape weights sequences. In the following

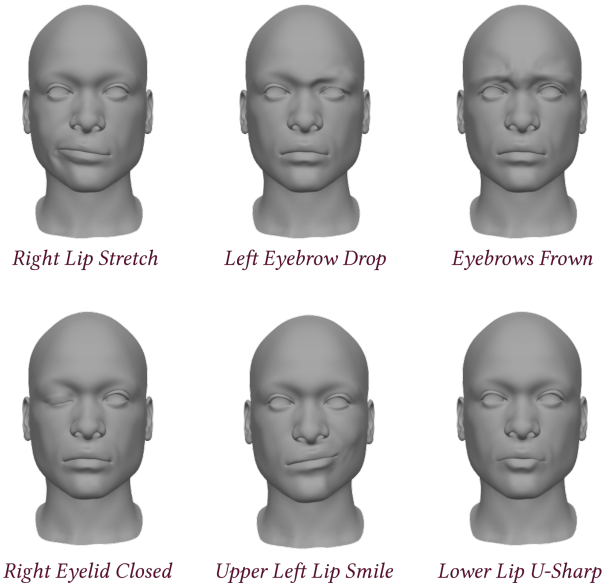


Figure 3.5 Subset of the blendshapes used in this work. The blendshapes correspond to local and semantic deformations, often referenced as facial expression units.

parts, we present the blendshape paradigm,and detail the expression transfer workflow to derive our extended database encompassing the steps ② and ③ (see Figure 3.4).

Blendshape Parametrization The blendshape model defines animation as a linear combination of shapes. These shapes express semantic deformations and approximate the facial muscle activation [CLK01] described by the Facial Action Coding System (FACS) established by Ekman and Friese [Ekm97].

A face mesh vector $\mathbf{s}$ can be expressed in the *delta* blendshape model as:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{N_b} w_i \mathbf{b}_i, \tag{3.9}$$

where $\mathbf{s}_0$ is the face in the neutral configuration vector, $\mathbf{b}$ are the blendshape vectors and N_b is the dimension of the blendshape model. The blendshape weights w conventionally vary within the range $[0,1]$ and reflect the extent to which the corresponding blendshape is activated.

Due to its attractive properties, the blendshape model has been extensively used to manipulate performance-based animation [LAR+14]. A subset of the blendshapes used in this work is presented in Figure 3.5. We leverage existing blendshape models that have been sculpted by an artist on the face template mesh $\mathbf{T}$.

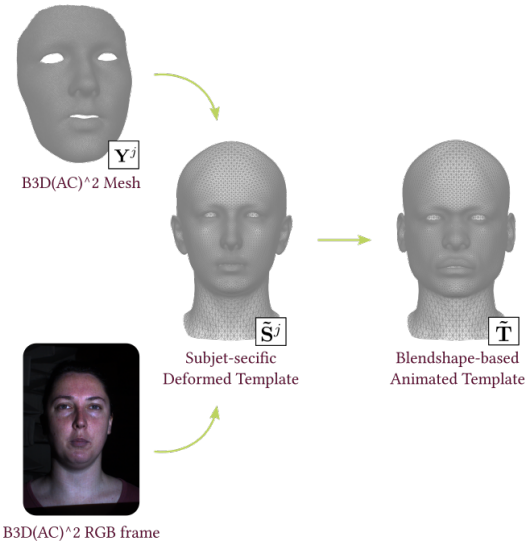


Figure 3.6 Examples of the 3D blendshape weights derivated using our workflow. From the subject specific blendshape model $\tilde{\mathbf{S}}^j$, the B3D(AC)^2 input mesh and the 2D information extracted from the input RGB frame, we derived the corresponding blendshape coefficients.

Deformation Transfer The second part of the database extension workflow, illustrated in Figure 3.4, incorporates two steps: the subject-specific blendshapes model construction and blendshape weight fitting.

The goal of the first step is to determine from the subject-specific deformed template mesh $\mathbf{S}^j$, the corresponding subject-specific blendshape matrix $\mathbf{B}^j = [\mathbf{b}_1, ..., \mathbf{b}_{N_b}]$. To do so, we exploit the popular deformation transfer introduced by Sumner and coworkers [SP04], to map the set of pre-defined expressions deformations (the blendshapes), $\tilde{\mathbf{T}}^i_{bs}, i = 1, .., N_b$, sculpted in our template mesh $\mathbf{T}_{bs}$ onto the subject's deformed models $\mathbf{S}^j$. We applied this efficient framework to compute the set of blendshapes $[\tilde{\mathbf{S}}^j_1, ..., \tilde{\mathbf{S}}^j_{N_b}]$ for each subject and derive the desired subject-specific blendshape matrix for each subject.

Blendshape Weight Fitting on Sequences Finally, we derive our final dataset of N_b blendshape animation weights. The last step corresponds to the fitting of the given blendshape model (Equation (3.9)) on the input 3D data. In this step, we solve the optimal blendshape weights for each frame using a non-rigid optimization and the 2D information, as described by Li and colleagues [LYYB13]. Examples illustrating this step and the different notations are shown in Figure 3.6.

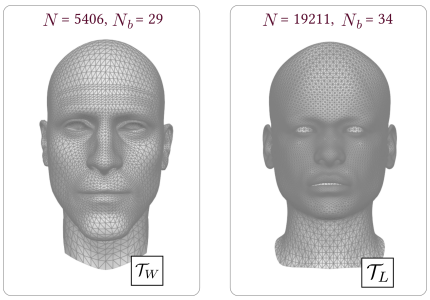


Figure 3.7 Examples of templates and their Specifications. We use different handcrafted blendshape templates for our work with the N vertices and N_b Blendshapes.

3.3.3.3 Processing Details and Results

Our processing work leads to suitable 3D animation data representation for learning techniques with fewer parameters $N_{bs} < 50$ instead of $L = 23370$, while respecting the 3D original shape (see Figure 3.8). This representation disentangles expressive and morphology information, necessary to provide efficient animation editing solution, and is compliant with the actual animator's workflow, enabling our work to take place in the facial animation pipeline. Through this data processing step, we enhance the original B3D(AC)^2 data by incorporating information about the eyes and the eyelids in the 3D model, using a better template than the original one [FGR+10]. We notably refine the mouth shape to more accurately replicate the original actor's expression, as shown in Figure 3.8. This workflow was successfully performed using different templates, which then enables us to validate the results presented in this manuscript. An example of these templates, with their specifications, is presented in Figure 3.7.

Discussion We leverage this extended dataset suitable for neural-based approaches to perform our experiments. We have made public this dataset for reproducibility purposes, and with the goal of contributing to the enrichment of the limited available facial animation resources. The data are available at `http://www.rennes.centralesupelec.fr/biwi3D`. Despite the presented work improving the suitability of this data for learning-based techniques, some issues remain unaddressed. Indeed, the native capture frame rate of the videos is 25 fps, which is too low to acquire all relevant natural facial cues. Important high-frequency information has already been lost at acquisition time.

Since then, a new multimodal database has been released: Cudeiro et al. [CBL+19] introduce a 4D facial animation corpus, VOCASET, based on a more sophisticated model, the *FLAME* morphable model [LBB+17]. We perform all the steps presented in this section, except that we do not incorporate 2D landmark information, to test the generalization of our solution on the last released data (see Chapter 5). Yet, we note that the quality of this data remains below professional quality standards.

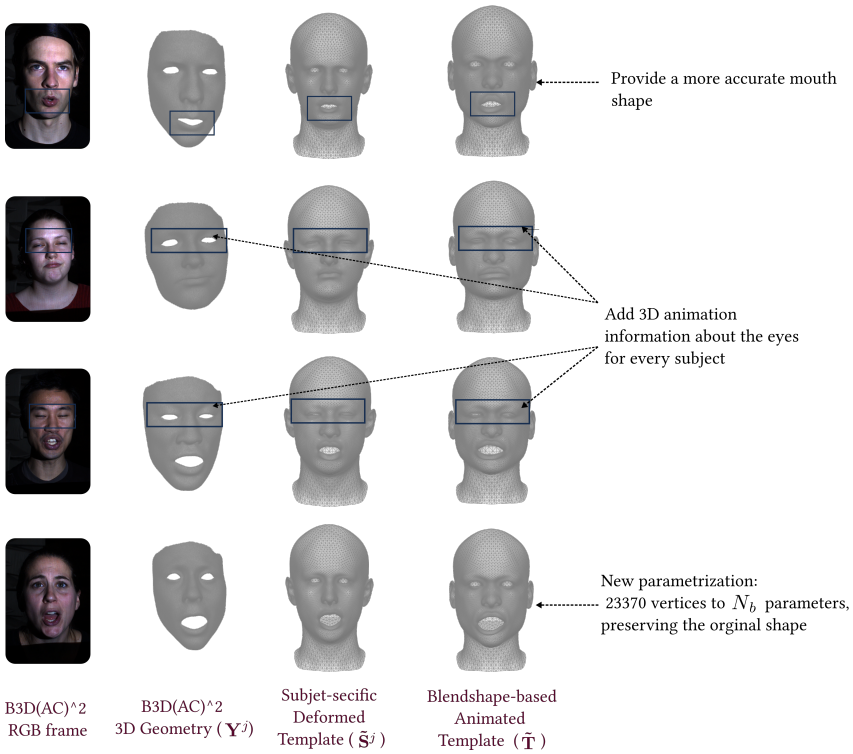


Figure 3.8 Examples of 3D animation derivated using our workflow. Our processing work leads to suitable 3D animation data representation for learning techniques with fewer parameters and compliant with the actual animator's workflow. In addition, it enhances the original B3D(AC)^2 by adding animation information about the eyes.

Our work is not devoted to enhancing the realism and the accuracy of the synthesized 3D performance-based facial model but instead, emphasizing novel approaches for intuitive and efficient tools to manipulate existing facial animation. Despite the quality of some sequences not being up to professional quality standard, it exhibits sufficient accuracy and diversity to support learning-based algorithms. It is however natural that the animation generated by learning-based techniques will match, at best, the quality of their learning database. Beyond this limitation, we aim to further advance in the facial animation editing and exploring the facial animation mechanism.

3.3.4 Conclusion

In this chapter, we presented the two databases used throughout our work, which enable us to explore the different facets of facial animation editing. A summary of the properties of these databases is presented in Table 3.1.

Table 3.1 Summary table of the databases properties used in this work.

	Performance-based	Extended B3D(AC)^2
Creation	Made by Artists	Computed
Quality	++	+
FPS	30,60,120	25
Included	MoCap-based noisy animation	Audio, transcription & alignment
Status	Private	Public
Total Animation Time	49 min	85 min

Next chapters focus on our contributions: we investigate new methods to improve various aspects of the facial animation editing.

CHAPTER 4

Real-time Cleaning and Refinement of Facial Animation

Contents

4.1 Introduction

This thesis investigates scientific approaches to tackle facial animation cleaning and editing. In this chapter, we focus on the most problematic issue with modern mocap-based facial animation pipeline, which is cleaning out noise and local artifacts. Due to hardware limitation and software restrictions, it is usual that the resulting 3D MoCap-based animation signals contain noise. Among other reasons, such artifacts

might be due to environmental interferences, such as lighting changes, sensor noise, data occlusion, and induce reduced accuracy and jitters in the resulting animation. To obtain clean and high quality data, further post-processing usually involves manual intervention to ensure a realistic and accurate animation.

Animation cleaning has been widely investigated to deal with body motion capture errors, including body penetration, marker occlusions or wrong body contact with the environment. Yet, body MoCap issues revolve around marker mis-identification, swapping, disappearance or slipping. Modern facial motion capture tends to favor different technologies, fitting global shape models on dense signals such as video pixels or depth values. Unlike body motion capture technologies, their dominating artifacts are fitting jitters due to sensor noise, or spikes/aberrations in the source material. Existing real-time motion capture solutions have opted for standard signal processing methods to strengthen temporal coherence of the resulting animations and remove inaccuracies. However, traditional signal processing methods such as the Kalman filter or Gaussian smoothing process often fail at preserving the subtleties of facial motions. For instance, a blink constitutes an abrupt spike in the eyelid motion signal. With aforementioned filtering frameworks, transient motion like blinks end up oversmoothed. Therefore, while these methods produce smooth results, they inherently filter-out part of the dynamics of facial motion, such as high-frequency transient movements. Indeed, in the case of facial animation, automating filtering/cleaning is a tough problem requiring a careful understanding of the signal-to-noise ratio, as facial dynamics induce both low- and high-frequency of complex motions that are hard to model [ASK+12, BBG+14]. Besides, the frequency content can undergo significant variations over different frames, when the actor is talking and emoting quickly, changing his facial expression suddenly.

Current works have made the jump to learning-based methods, to cope with the deficiencies of the traditional filtering. Mall and colleagues [MLCC17] learn adaptative filters for each animation parameters, demonstrating successful results in cleaning any kind of actions. Alternatively, many works address the topic of learning natural motion model with neural networks, typically with CNN architectures [HSKJ15, HSK16, BBKK17]. CNN architectures and filter-based techniques are however non-causal, as they use future time samples to process the current one, limiting their applicability to offline tasks. Causal architectures such as Recurrent Neural Network have proven successful in processing sequential data in language modeling [SMH11], human motion prediction [FLFM15] or speech recognition [GJ14].

In this chapter, we follow these works and propose a real-time animation refining system that preserves -or even restores- the natural dynamics of facial motion. Our system learns the complex facial motion dynamics from data, and thus has the ability to preserve natural-looking motion, even transient ones. We leverage Long Short-Term Memory architecture to produce natural motion models for animation filtering, presented in Section 4.2.2.

Learning the dynamics of facial motions for real-time applications also differs in that it cannot rely on a known, fixed sensor frame rate. Traditional resampling algorithms are not viable, as they require to know future samples. Also, real-time source for face images, such as webcams, can have nonconstant framerates. We suggest a new reformulation for the parametrization of the input signal to bypass this difficulty. Rather than a system predicting the next frame values given the past frame values, our recurrent network is trained to learn the values of the signal's derivatives at the current frame. Considering the temporal derivative of the motion sidesteps the problem of frame rate dependency at run-time. We observe that feeding a recurrent network with the previous estimated states and the dynamic features of the mocap-based signal, allows for processing animation with infinite length overcoming (see Section 4.4)

One tricky aspect of motion capture signal cleaning is that the 3D facial animation ground truth matching facial motion capture signals is hardly ever available; it would require a really cumbersome and expensive setup to acquire data. We propose to overcome this difficulty by leveraging handmade animation database created with a professional performance-based animation software [Dyn19b], presented in Section 3.2, and train our network to minimize the difference between the resulting animation and these created data.

Chapter Overview This chapter is organized as follows: Section 4.2 is dedicated to the presentation of our overall system, combining an effective parametrization and a specific learning strategy with a popular neural architecture for temporal data processing. In Section 4.3, we present qualitative results showing that our system is able to retrieve natural motion signals from noisy or degraded input animation, along with comparisons with standard motion signal processing methods, highlighting the effectiveness of our solution. In Section 4.4, we present additional experiments and ablation studies, bringing to the fore a more comprehensive understanding of the relevance of the proposed system. Finally, we discuss the application of our system on a related problem in Section 4.5 and further perspectives in Section 4.6.

4.2 RNN motion cleaning system

Our goal is to enhance the accuracy and remove artifacts of a performance-based animation. To this end, a learning-based solution is trained to turn a noisy animation into a realistic one. In this section, we begin by detailing the particular parametrization we use to make values in our system independent of the framerate of the input data (Section 4.2.1). Then, we explicit the proposed architecture based on a Recurrent Neural Network (Section 4.2.2), as well as the training procedure of our recurrent neural network (Section 4.2.3). An overview of our system is shown in Figure 4.1.

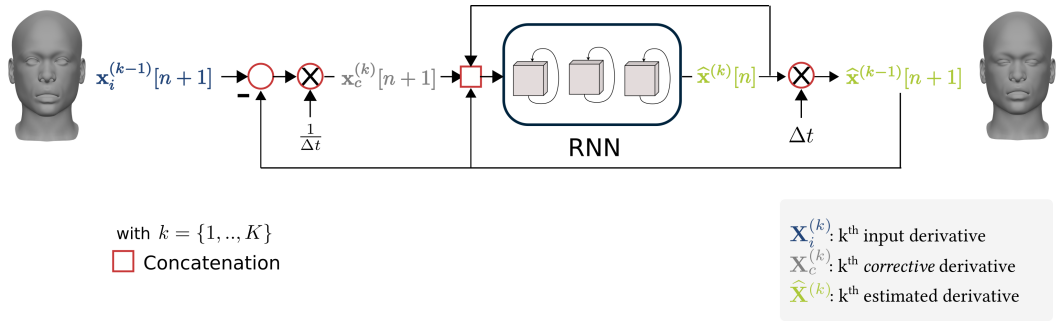


Figure 4.1 System overview. Our recurrent system takes as an input the first n moments of the estimate signal at time t-1 as well as a corrective moments of the inputs and regress n derivatives at time t.

4.2.1 Parametrization of the system

Our filtering system essentially refines MoCap-based facial motion to produce natural looking animation. We use the performance-based animation database presented in Section 3.2 to perform our experiments. Formally, the input data of our system is represented as a sequence of N frames of M=34 blendshape coefficients $\mathbf{X} = \mathbf{X}^{(0)} = [\mathbf{x}[0], .., \mathbf{x}[N]]^T \in \mathbb{R}^{N \times M}$. We design our system to be framerate-independent: instead of correcting the absolute value of the current motion, we consider the normalized temporal k-th order derivatives of the motion signal:

$$\mathbf{x}^{(k)}[n] = \frac{\mathbf{x}^{(k-1)}[n+1] - \mathbf{x}^{(k-1)}[n]}{\Delta t[n]}, \tag{4.1}$$

where $\mathbf{x}^{(k)}$ denotes the forward k^{th}-order derivative of the motion at the frame n and Δt is the time between two consecutive frames. With this formulation, the framerate information is factored out of the input, preventing our network to be reliant on it at both training and inference time.

At each frame n, our system aims at predicting the forward k derivatives, $\hat{\mathbf{x}}^{(k)}[n]$ with $k = \{1, .., K\}$ (*green* on Figure 4.1) given the previous estimated animation $\hat{\mathbf{x}}[n]$, the estimated derivatives $\hat{\mathbf{x}}^{(k)}[n-1]$, and the current *corrective* forward derivatives $\mathbf{x}_c^{(k)}[n]$ (*grey* on Figure 4.1, see below for details).

Finally, from $\hat{\mathbf{x}}^{(k)}[n]$, we recover the estimate $K-1$ derivatives using the equation 4.1 (k=0 corresponds to the absolute blendshape values).
In this study, we have observed that feeding back to the system a measure of how much the currently produced state deviates from the real input signal improve the performance of our system. Hence, we give as input to our network the k *corrective*

temporal derivative, $\mathbf{x}_c^k$, of the input signal, $\mathbf{x}_i$ formulated as:

$$\mathbf{x}_c^{(k)}[n] = \frac{\mathbf{x}_i^{(k-1)}[n+1] - \widehat{\mathbf{x}}^{(k-1)}[n]}{\Delta t[n]}. \tag{4.2}$$

where $\mathbf{X}_i$ and $\widehat{\mathbf{X}}$ are respectively the input and the generated animation. Besides, we add information about the dynamics of the sequence by adding a residual connection between the input and the output of each RNN cells of our network as proposed by Martinez et al. [MBR17].

4.2.2 Recurrent Neural Network (RNN) for motion modeling

Our approach is based on a recurrent network [FLFM15, MBR17]. The goal of this work is to learn to generate proper facial dynamics from data. Our network is depicted in Figure 4.1 and mainly consists of a sequence of LSTM layers with a stacked final dense output layer to get dimensions matching the output features.

As explained in Chapter 1, computing the facial animation inherently implies to take the temporal aspect of the motion signal into account. RNN is a type of neural architecture that has been widely used to model temporal behaviours of complex patterns. Its relevance cames from the parameters shared over time which learn to focus or forget temporal relations between input features. The temporal context is propagated through hidden states, retaining relevant information from history. The RNN represents the motion signal at the instant t through the hidden state expressed as:

$$\mathbf{h}_n = n_{\boldsymbol{w}}(\mathbf{x}_n, \mathbf{h}_{n-1}). \tag{4.3}$$

This definition puts forward the temporal dependency of this representation to the previous states $(\mathbf{h}_{n-1}, \mathbf{h}_{n-2}, ..., \mathbf{h}_0)$. The initialization of the first state $\mathbf{h}_0$ is either randomly or deterministically set, depending the objective task. RNN is therefore rightly suitable for handling temporal dependence of facial motions.

However, RNN might suffer from the vanishing gradient pitfalls in case of long sequence processing. The gradient value of RNN is evaluated considering the temporal dependence of the coefficients over time, using Back Propagation Through Time (BPTT) [Wer88]. Passing through the numerous activation layers, the RNN gradient magnitude is prone to a drastic reduction or expansion overtime.

Hochreider and Schmidhuber [HS97] develop the Long Short-Term Memory (LSTM) structure and overcome this issue by introducing an internal state into the RNN preserving the long-term memory. The LSTM relies on a combinaison of gates to control the flow of information passing through the network, and updates its internal memory $\mathbf{c}_n$ considering previous states and current input sample, shown in Figure 4.2. Three gates, named the *forget* $\mathbf{f}_n$, *input* $\mathbf{i}_n$ and *output* $\mathbf{o}_n$ gate, learn to select over time which information must be added for later prediction and which information is no longer needed and has to be forgotten. These gates are expressed as follows:

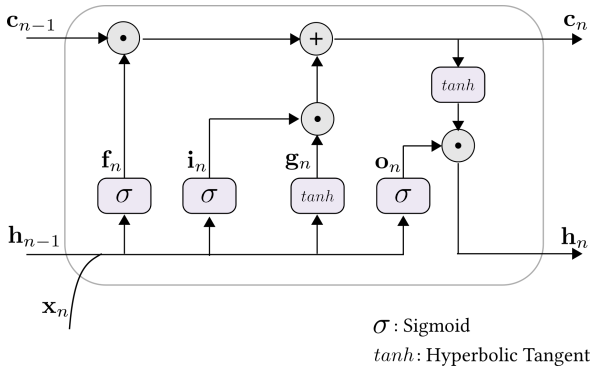


Figure 4.2 The Long Short-Term Memory (LSTM) structures. Through a gates game, the LSTM network controls the information flow over time. Its internal state is updated considering the previous states and the current input sample, preserving or forgetting its temporal motion knowledge.

$$\begin{aligned}\mathbf{f}_n &= \sigma(\mathbf{W}_{fx}\mathbf{x}_n + \mathbf{W}_{fh}\mathbf{h}_{n-1} + \mathbf{b}_f),\\ \mathbf{i}_n &= \sigma(\mathbf{W}_{ix}\mathbf{x}_n + \mathbf{W}_{ih}\mathbf{h}_{n-1} + \mathbf{b}_i),\\ \mathbf{o}_n &= \sigma(\mathbf{W}_{ox}\mathbf{x}_n + \mathbf{W}_{oh}\mathbf{h}_{n-1} + \mathbf{b}_o),\end{aligned} \tag{4.4}$$

where σ is the Sigmoid nonlinear activation function. The internal memory cell $\mathbf{c}_n$ and the hidden states $\mathbf{h}_n$ are iteratively updated at each step as:

$$\begin{aligned}\mathbf{g}_n &= \tanh(\mathbf{W}_{gx}\mathbf{x}_n + \mathbf{W}_{gh}\mathbf{h}_{n-1} + \mathbf{b}_g),\\ \mathbf{c}_t &= \mathbf{c}_{n-1} \odot \mathbf{f}_n + \mathbf{i}_n \odot \mathbf{g}_n,\\ \mathbf{h}_t &= \mathbf{o}_n \odot \tanh(\mathbf{c}_n),\end{aligned} \tag{4.5}$$

where $\odot$ denotes the pointwise multiplication. The LSTM cell formulation ensures a tractable gradient through time while memorizing relevant temporal dependencies. Overall, LSTM have been largely adopted to process long sequences. Thus, we benefit from well-established LSTM [HS97] capacities to model and forget temporal dependencies to carry out this task.

4.2.3 Learning Details

As our network is thought for real-time animation, it is inputted with past time samples. Its objective is to predict a plausible estimation of facial motion given previously estimated states and the corrective derivative of the input signal (Equation 4.2). Therefore, at training time, we formulate the cost function as the mean square error (MSE) between the animation made by an artist, $\mathbf{X}_{gt}$ and the system's estimate output state $\widehat{\mathbf{X}}$.

$$\mathcal{L}_{MSE} = ||\mathbf{X}_{gt} - \widehat{\mathbf{X}}||^2. \tag{4.6}$$

We also encourage our network to focus on the higher-order dynamics of facial motion with an MSE between the derivatives of the estimate motion and the ground truth one:

$$\mathcal{L}_{der} = \sum_{k=1}^{K} ||\mathbf{X}_{gt}^{(k)} - \widehat{\mathbf{X}}^{(k)}||^2. \tag{4.7}$$

The blendshape parameterization is not the most representative of the importance of each movement they encode. Movements such as mouth openings/closures carry more expressive and communicational weight than others such as nose movements. The loss that our network learns to minimize should reflect this aspect. We add a loss $\mathcal{L}_{dis}$, to focus on the preservation of some key inter-vertices distances between the estimate and the ground truth animations:

$$\mathcal{L}_{dis} = ||\mathbf{D}_{gt} - \hat{\mathbf{D}}||^2 + \alpha_{dis}||\mathbf{D}_i - \hat{\mathbf{D}}||^2, \tag{4.8}$$

In this loss, we include six distances: the first three, $v_{lips_s} \in \mathbb{R}$, where $s \in \{middle, right, left\}$, measure the extend between the upper and the lower lips (at the middle and at one and two third of the mouth), the fourth is between the mouth corners $v_{corners} \in \mathbb{R}$ and the last ones between the right and left eyelids $v_{eyes_s} \in \mathbb{R}, s \in \{right, left\}$. All these quantities are concatenated into a large matrix $\mathbf{D} = [\mathbf{d}^0, ..., \mathbf{d}^L] \in \mathbb{R}^{L \times 6}$, with $\mathbf{d}^l = \{v^l_{lips_{middle}}, v^l_{lips_{right}}, v^l_{lips_{left}}, v^l_{corners}, v^l_{eyes_{right}}, v^l_{eyes_{left}}\}$. This loss emphasizes the salient role of the lips and eyes to convey expressivity and communicational cues in facial animation. Finally, we optimize the following cost function:

$$\mathcal{L} = \mathcal{L}_{MSE} + w_{der}\mathcal{L}_{der} + w_{dis}\mathcal{L}_{dis}. \tag{4.9}$$

For all our experiments, we set w_{der} and w_{dis} at 0.01 and 0.1, α_{dis} at 0.8 and use K=1. The dimension of the hidden states is set to 128 for every LSTM layer. Our network is optimized using the ADAM algorithm [KB14]. During the training, we add a dropout [SHK+14] of 0.3 to avoid overfitting. We set the initial learning rate at 0.001.

4.3 Animation Filtering Results

In this section, we demonstrate the capability of our model to clean-up a noisy performance-based animation while preserving a plausible facial motion dynamic in Section 4.3.1. We also compare our system to standard signal processing methods to highlight the difficult task of hyperparameters tuning in the case of motion signals filtering (see Section 4.3.2). Finally, we demonstrate the relevance of our recurrent structure by comparing our system with non-recurrent learning methods in Section 4.3.3.

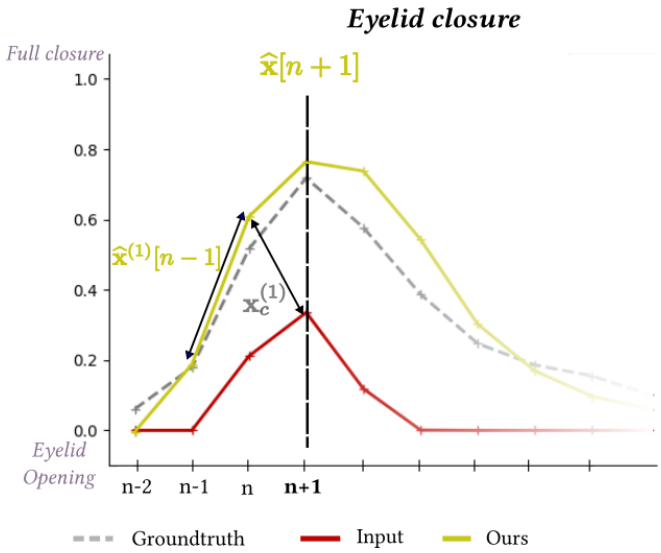


Figure 4.3 Our system detects a blink pattern and correct the motion to retrieve a realistic full closure of the eyelids.

4.3.1 Motion Refinement

Facial performance-base systems often rely on video sources to capture motion and solve for animation. Most of the time, either due to sensor quality or environmental factors (lighting changes, occlusions), the delivered animation contains noise and inaccuracies. Some crucial properties of facial animation, such as the amplitude of the movements, are often lost resulting in a less expressive animation. As shown in Figure 4.3, our system recovers the natural dynamic of the facial motion. For instance, providing the previous estimate velocities, our learning-based method detects an upcoming blink and correct the animation to get a natural full closure of the eyes.

Our system has to predict the expression parameters of the whole face at each time step, without having the ground-truth available. Hence, it learns the correlation between animation parameters, as we can see in Figure 4.4 (top). Our system can "magnify" the motion by augmenting movements in the animation, which were absent from the noisy input. Conversely, when unrealistic blendshape activation patterns appear, our system efficiently smooths the signal (see Figure 4.4 (middle, bottom)).

The time to infer one frame is less than 0.5 ms on GPU (GeForce GTX 1060). Hence, our system could be integrated in any real-time facial animation software [1].

[1] More results on full animations are provided at `https://elo-nsrb.github.io/homepage/publi_data/ICGSP2020/video.mp4`

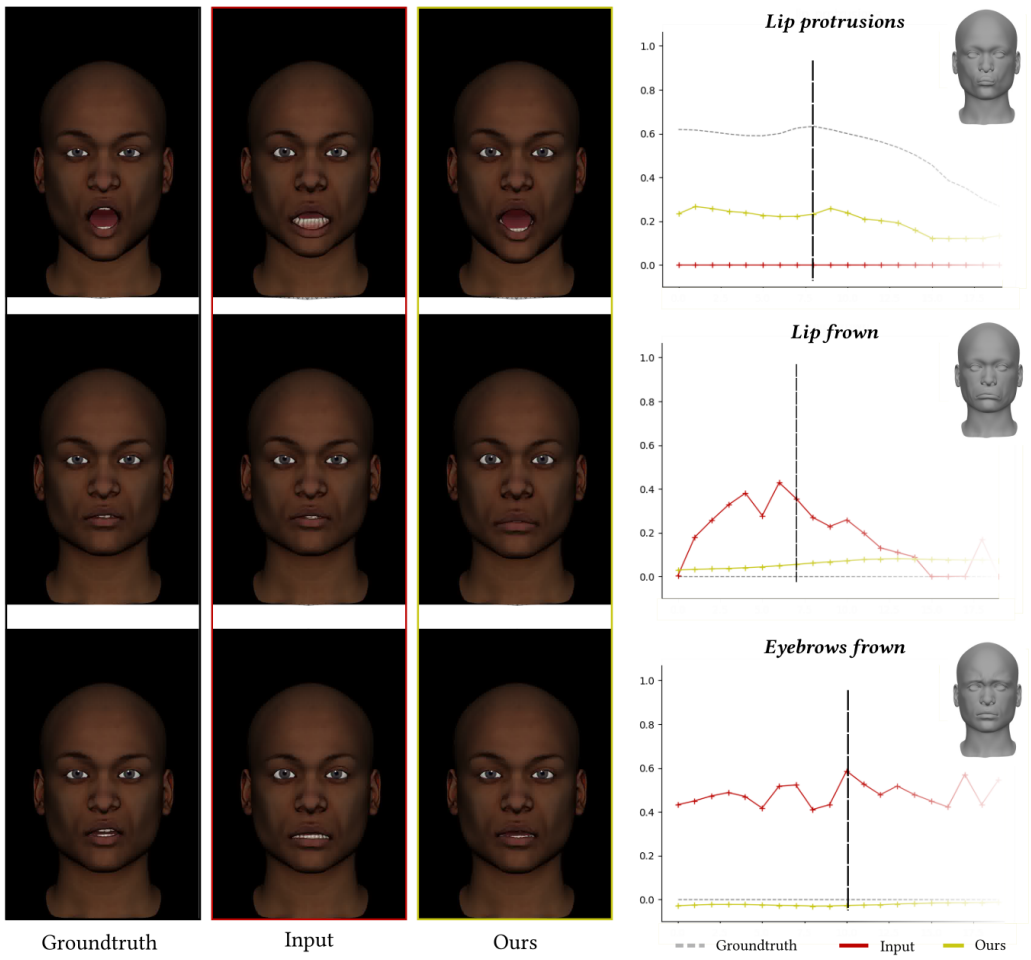


Figure 4.4 Our system corrects the motion of every part of face: either by increasing the motion such as the "lip protrusion" motion or by smoothing the lips or the eyes frown movement.

4.3.2 Comparison with traditional temporal filters

One shortcoming of standard filtering methods is hyperparameters tuning. One needs to find a trade-off between preserving high-frequency patterns such as a blink or noise or getting a smooth animation and losing the natural dynamics of some part of the face. For instance, one popular filtering algorithm for real-time processing is exponential smoothing:

$$\widehat{\mathbf{x}}[n] = \gamma \mathbf{x}[n] + (1-\gamma)\widehat{\mathbf{x}}[n-1]. \tag{4.10}$$

Setting a high γ results in an estimate signal, which is more faithful to the input. In this case, more subtle motion patterns of the input signal are kept. Conversely,

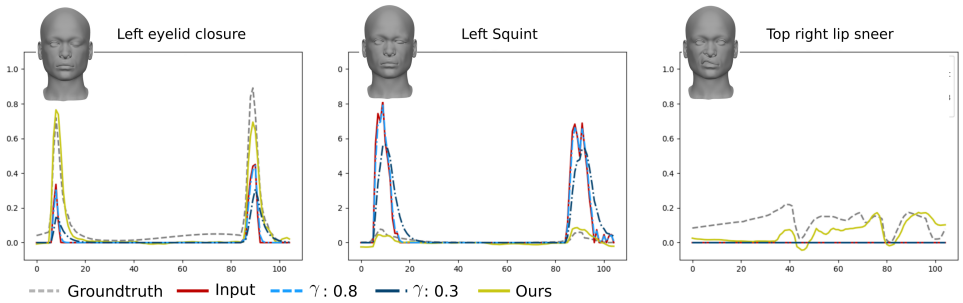
setting a low γ prevents from high variations in the estimate motion signal leading to a smoother animation.
Another popular smoothing scheme is Gaussian-based filtering. It consists of convolving the input signal with a Gaussian window, and thus requires having the whole signal. The smoothness of the output signal depends on the resolution of the window fixed by standard deviation σ of the Gaussian. The lower the standard deviation, the higher the temporal resolution of the window. A narrow window better preserves the fine temporal details. In both cases, the techniques only filter the animation and cannot refine it. The dynamics of the different part of the face is very different and complex to model. While the eyelids motion is composed mainly of flat portions and quick spikes corresponding to blinks, frowning movements consists of more subtle variations with variable lengths. Handmade tuning of γ or σ parameters is thus a cumbersome task. By learning the inherent dynamic faces, our method is free from such frequency parameters tuning.

We compare our system with the temporal smoothing algorithm, parameterized with two different values of γ 0.3 and 0.5 (see Figure 4.5a) and with the Gaussian smoothing using a window with σ of 1.0 and 5.0 (see Figure 4.5b). Our system is able to enhance eyelids signal producing accurate closures of the eyes. Indeed, our system detects the inaccurate spikes observed on the "*Top right lip sneer*" blendshape and corrects it to produce a smoother and a more natural motion signal. Conversely, both methods process the spikes observed on the "*Top right lip sneer*" blendshape and on the "*Eyelid*" motion similarly, by either preserving it or smoothing it. At each time step, our system is fed with the motion of the whole face. As shown in Figure 4.5, our system is able to learn natural correlations in facial motion and use this knowledge to correct and generate more accurate motion sequences, even if the input tracking is inexact.

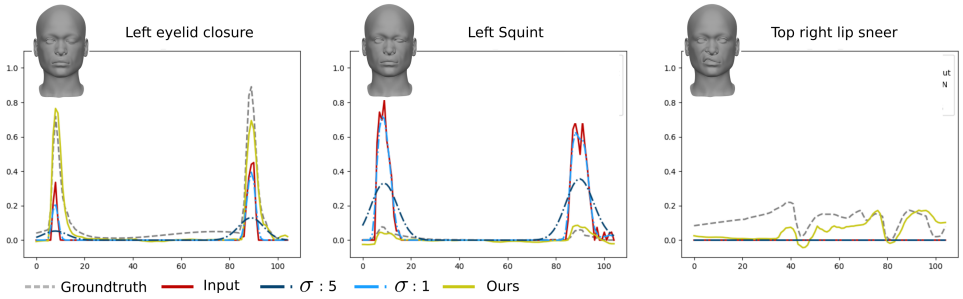
Table 4.1 Quantitative comparison with Exponential and Gaussian smoothing algorithms.

	MSE
Exponential (γ:0.3)	0.0170
Exponential (γ: 0.8)	0.0172
Gaussian (σ: 1)	0.0170
Gaussian (σ: 5)	0.0167
Raw input	0.0173
Ours	**0.0140**

We also numerically compare the MSE of those algorithms on the test set in Table 4.1. Our system gets the lowest MSE, while smoothing methods get a MSE similar to the MSE between the input and the ground truth.



(a) Comparison with exponential smoothing.



(b) Comparison with gaussian smoothing.

Figure 4.5 Common filtering methods require hyperparameter tuning to balance between oversmoothing and details preservation. Our system learns the dynamic of facial motions so as to tailor the filtering process for each motion. Hence, it can enhance a blink motion while removing unrealistic spikes in "left squint" motion. It is also able to retrieve natural patterns in the "right lip sneer" motion for instance.

4.3.3 Comparison with non recurrent learning methods

We compare our system with non-recurrent learning methods: a Fully Connected neural network (FC) and a non-neural machine-learning algorithm using Gradient-Boosted Trees (GBT). In these algorithms, the estimates of previous derivatives are not fed back at train time. Hence, we adapt the input parameterization by replacing the estimate of previous outputs with parameters of the input signal. At each frame n, we estimate the K derivatives of the corrected signal at time $n+1$, given the K derivatives of the input animation at n and $n-1$, and the current state of the input.

As these methods are not recurrent and to avoid accumulating errors through time, the estimate animation is derived as:

$$\mathbf{X}_r^{(k)} = \alpha_r(\Delta t\widehat{\mathbf{X}}^{(k+1)} + \mathbf{X}_r^{(k)}) + (1-\alpha_r)(\widehat{\mathbf{X}}^{(k)}),$$

with:

$$\widehat{\mathbf{X}}^0 = \mathbf{X}_i, \mathbf{X}_r^{(K)} = \widehat{\mathbf{X}}^{(K)}$$

$\mathbf{X}_r$ is the resulting animation, while $\widehat{\mathbf{X}}$ is the estimate animation by the FC/GBT

algorithms. We augment the training set by upsampling and downsampling each sequence with a factor 2 to avoid overfitting. We train both the GBT and FC using K=4 and produce the final animation using empirically chosen values of α_r=0.97 (GBT) and α_r=0.9 (FC). During the training, we optimize the loss $\mathcal{L}_{der}$. We also test these algorithms on animations with different framerates.

Figure 4.6 depicts two frames extracted from the resulting animation of GBT or FC and the animation estimated by our system when it is fed with a performance-based animation recorded at 30 fps. Compared to these algorithms, our system not only rectifies the motion signals but also enhances the expressiveness of the animation. As we can see, non-recurrent methods tend to flatten the motion signals, whereas our system produces natural motion patterns preserving the amplitude of eyebrows frowning or lip protrusion movements. We also numerically compare the MSE error obtained on the test set, and observe that our recurrent method gets a lower MSE than GBT or FC architectures (Table 4.2). We improve the accuracy of these methods and the plausibility of the output animation by limiting the regressed coefficients to the range $[0, 1]$ (see *Clip* in Table 4.2). Dynamic animation results are shown in the supplementary video.

Table 4.2 Quantitative comparison with non-recurrent methods.

	MSE
GBT + *Clip*	0.0451
FC + *Clip*	0.0462
Ours	**0.0140**
Ours + *Clip*	**0.012**

4.4 Ablation Study and Model Understanding

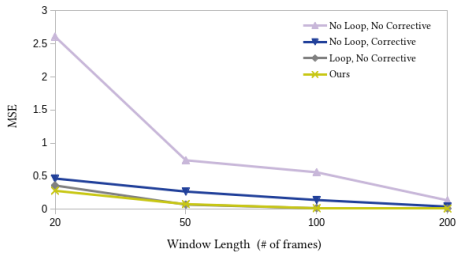
With the mind of pushing forward our understanding of the proposed system, we conduct several experiments highlighting the contribution of each part of the architecture in the overall performance. To do so, we train our system with different input segment lengths in Section 4.4.1 and various architecture components in Section 4.4.2 on more epochs than in the previous paragraph to avoid an early stop. For our experiments, we evaluate the different systems retaining the configuration that gets the lowest validation error during the training.

4.4.1 Optimal Input Segment Length

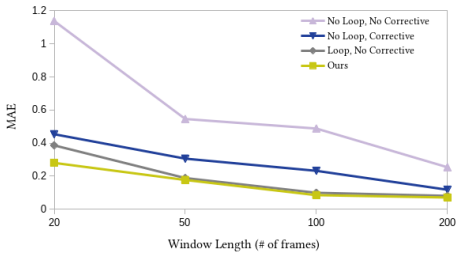
One key ingredient of temporal sequence modeling is to properly define the scope of the motion representation. While at runtime, our system can generate sequences with arbitrary length, during the training the size of the input sequence is fixed to

Figure 4.6 Comparison with non-recurrent machine-learning algorithms on a new-recorded performance-based animation: a Gradient Boosting Trees (GBT) and a Fully Connected neural network (FC). Our system not only rectifies the motion signals but also enhances the expressiveness of the animation, such as enforcing the eyebrows frowning movements or the lip protrusion.

enable parallel computing. Yet, the length of the input sequence should be large enough so as to capture all the salient temporal dependencies. We observe the effects of the input length on the performance of our system measured with two metrics the Mean Square Error and Mean Absolute Error. As shown in Figure 4.7, better performances are obtained with a larger input length. These results confirm the importance of learning the dynamics of facial motion on reasonably long motion chunks [TKY+17], taking at least temporal patterns of 1.7 seconds. Learning the motion with a too small context may not allow our system to properly disambiguate between two plausible motion dynamics.



(a) Mean Square Error with respect to the length of the input window.



(b) Mean Absolut Error with respect to the length of the input window.

Figure 4.7 Effect of the proposed architecture with respect to the length of the input window.

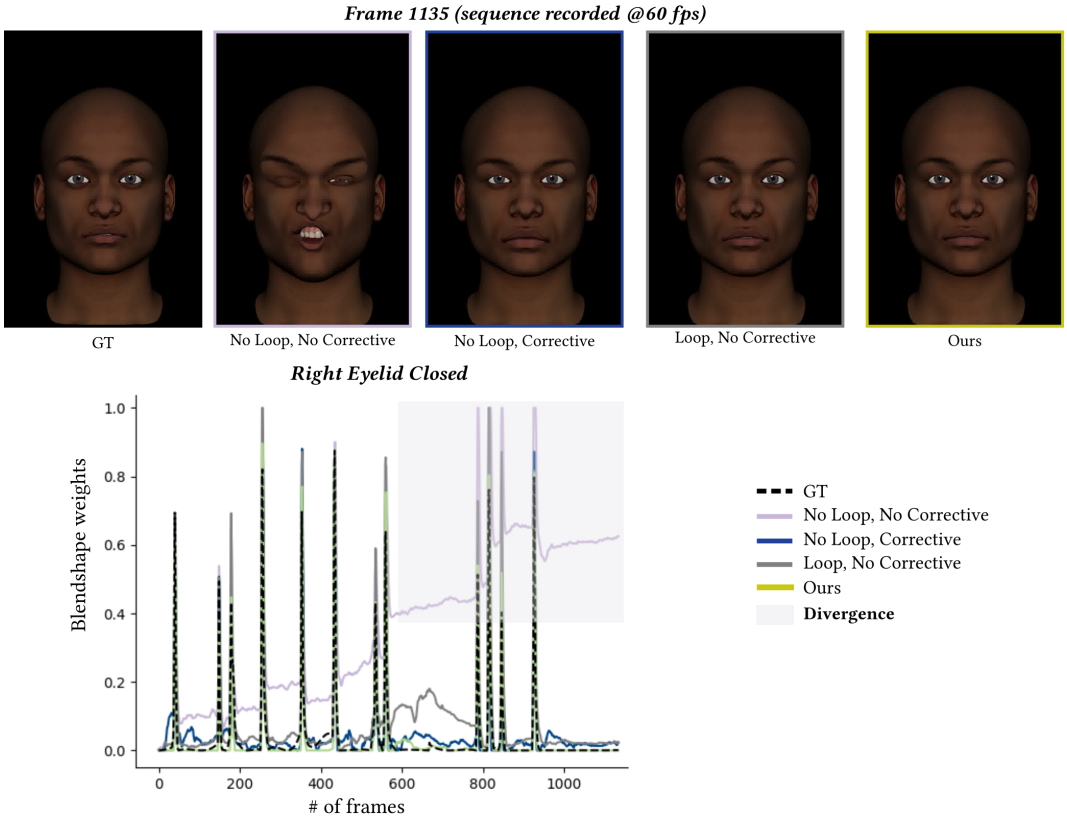


Figure 4.8 Effect of an external loop. We observe that feeding back the previous estimated state, either directly, concatenated to other inputs or indirectly by using the corrective input formulation (Equation (4.2)), precludes the divergence of the output even in case of very long sequences.

4.4.2 System Understanding

Model Efficiency We measure the impact of our looping strategy and our original corrective formulation of the input by comparing our system with three baselines. A first baseline, *No Loop*, consists of the system without the loop feeding the recurrent network with the previous estimate state. We introduce a second baseline, *No Corrective*, by replacing our corrective input formulation (Equation (4.2)) with the standard derivative formulation described in Equation (4.1). Finally, the third baseline, *No Loop, No Corrective*, refers to the system trained with neither the looping strategy nor the corrective formulation. As shown in Figure 4.7, both examined components contribute to the overall performance. However, we notice a difference in the degree of impact on the quantitative results: the looping strategy gets a greater influence on both the MSE and MAE. Our system gets the lowest MAE and MSE similar to the *No Corrective* system.

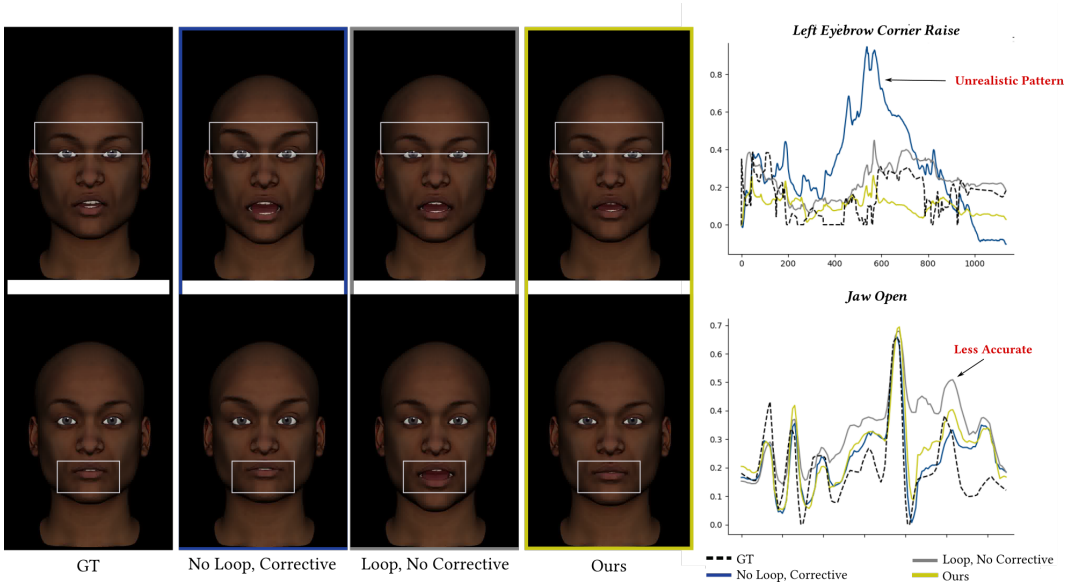


Figure 4.9 Relevance of our architecture. The looping strategy precludes unrealistic temporal motion patterns, while the corrective formulation of our input enables a more accurate output animation.

Model Suitability We further examine their role through qualitative analysis, putting forward the suitability of the proposed architecture for facial motion cleaning. As illustrated in Figure 4.8, feeding back to the system a measure of how much the currently produced state deviates from the real input signal through either an open-loop or indirectly by computing the *corrective* input states, prevents the system from drifting apart in case of long sequences inference. More precisely, the external loop precludes the system from ending-up with unrealistic patterns, still present in the *No corrective* system (see Figure 4.9). The corrective formulation beneficially leads to a more subtle but yet relevant improvement. As observed in Figure 4.9, the corrective formulation enables a more accurate restoration of motion dynamics. The *No corrective* system might lose valuable motion cues in case of highly dynamic animation segments, as we can see in Figure 4.9.

These results pose our system as an animation controller: it takes into account the difference between the previous generated state and the current input state to explicitly reduce the drifting effect happening on long-term predictions and improving the accuracy of the refined animation.

4.5 Text-based Animation Restoring

The objective pursued with this approach is the ability to deduce plausible dynamics from a coarse or noisy motion signal in order to ease manual animation tasks. We have seen above that we can restore/hallucinate realistic looking animation from

low-fidelity signals. This motivates investigation of whether we can do the same with even coarser inputs, such as restoring a plausible coarticulation dynamic from a coarse viseme-based animation. Herein, using the same paradigm, we discuss additional experiments we have done, extending the application of our system to a related problem, the text-based animation restoring.

Problem Statement and Motivation Given a sequence of viseme[2] along with the corresponding blendshape activation values, we derive a basic and unrealistic talking animation by converting the sequence of visemes in a sequence of blendshape coefficients, as illustrated in Figure 4.10. However, humans do not pronounce each phoneme separately, but instead, every phonetic realization impacts the surrounding articulations. In the linguistic field, this phenomenon is called *co-articulation* and refers to the mutual influence between the articulators involved in the movement to finish one sound and anticipate the next one [JM08, MC95, Sca04]. It is the leading mechanism responsible for smooth dynamic transitions between phonemes, allowing to realistically produce overlapping movement segments. However, this phenomenon is not well-modeled due to the intricate nature of the phoneme-to-viseme mapping, raising several questions about its representation. The nature of this mapping, which was early assumed as a many-to-one, appears insufficient to model the complexity of the visual movements underlying the sequence of uttered sounds. Besides, there is not a clear consensus about the visual unit and the boundaries of the visual segment to consider in this mapping. Taylor et al. [TMTM12] demonstrate that a dynamic visual unit better represents the visual appearance of an utterance than a discrete one. We believe that our system could address the above issue by retrieving this complex motion dynamics, in a real-time framework.

Discussion and Future Work We observe natural-looking and promising output animations. Yet, we still face issues such as the lack of a relevant testing strategy to quantitatively assess the quality of our results. As explained above, the way to pronounce sounds hinges on a subject-depend dynamic and might vary within the same subject performance, depending on the actor's style of acting and emoting. Thus, for the same utterance, one might find corresponding different equally valid animation within our database. It is all the more true as the sentence included in the B3D(AC)ˆ2 are uttered both with a neutral intent and an emotional intonation [FGR+10]. Although the original animation sequences, standing for the ground-truth, intrinsically carry these features, it is not included in the initial sequence of phonemes. Thus, we cannot expect our system to restore it at all. The main limitation we face is thus the absence of a relevant metric measuring animation plausibility to guide our training. The use of the MSE loss should be replaced by a more relevant function that takes into account this discrepancy between the ground-truth data and the regressed output. In Chapter 6, we experiment with generative framework to overcome this issue. One interesting experiment would be to use only neutral sequences. Another

[2] A group of phonemes with the same visual representation

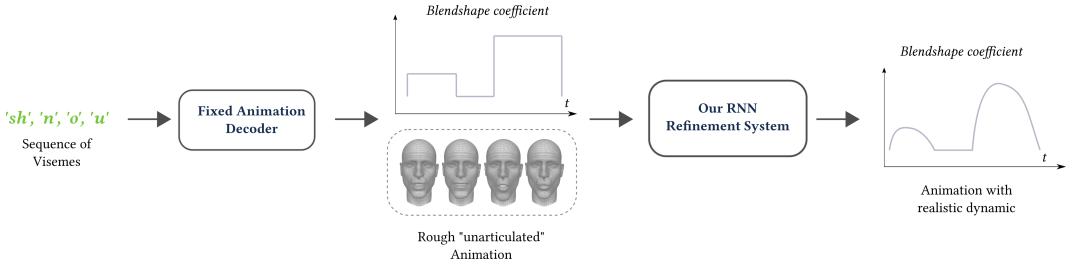


Figure 4.10 Potential application of our system. Instead of restoring the real dynamic of a noisy animation, our recurrent system could process a coarse talking animation, and derive the real coarticulation between the visemes.

issue comes from the time resolution of the animation signal which should be high enough to capture all the visual speech units and align an uttered sound to a visual gesture. We believe further research in this work is necessary to understand and evaluate the performance of our framework. More experiments should be carried to optimally leverage the presented architecture. In particular, a more in-depth analysis on our results with an accurate validation on our system on this attractive application.

4.6 Conclusion

In this chapter, we reported our investigation on a real-time facial animation cleaning and refinement system. Taking blendshape animation as input, such as raw motion-capture animation, our system successfully filters and enhances the animation, in real-time, regardless of the input framerate. Contrary to traditional signal processing method, our system learns the dynamics of facial motions on realistic data, to be able to remove noise and inaccurate signal patterns regardless of the animation signal frequency properties, and yet preserve high-frequency transient motions. Besides, the proposed solution is fully automatic and does not require further manual painstaking fine-tuning.

We demonstrate the benefit of using recurrent structure, endowed with an internal memory, to approximate a realistic dynamic based on the current state relevantly. Besides, we demonstrate that our system refines MoCap-based animation, delivering natural motion patterns with realistic correlations between the different parts of the face. By learning the derivatives of the motion rather than the motion's absolute values, we free our system from framerate dependency, enabling it to process any input animation in real-time.

As any learning-based system, our system strongly depends on the data used during the training. Thus, our system tends to produce only motion patterns it has already seen, refining animation in the style of the animations it has learned on. It is very likely that the amount and the type of data (inevitably bearing the style of the

animators who made it), strongly affects the delivered animation. A more in-depth analysis on database dependency and style learning would be required to get full control of the animation quality that our system outputs. A solution could be to segment the face into regions [JTDP03], as it makes the system more generalizable, and reduces the style-dependency on the database.

We provide a real-time and accurate system to clean and refine a motion signal that is in addition compliant with the blendshape animation formulation. We hope this study will provide additional evidence that learning-based system have great potential for motion cleaning and restoration. The encouraging quality of the results, coupled with the system's ability to handle arbitrary framerates in real-time also show that this approach has potential for industrial applications.

CHAPTER 5

Controllable Facial Animation Editing

Contents

5.1 Introduction

In the previous chapter, we addressed the very common animation cleaning problem, when noise or local artifacts are present in the signal. This low-level editing aims at modifying the signal so as to remove inaccurate patterns, without explicit external constraints. In the following chapters, we consider a given animation which is "technically" perfect, but does not entirely fulfill the user intent. Instead of signal-level cleaning, this animation requires additional semantic level adjustments or corrections.

These modifications are manually done by an animator trained in manipulating low-level 3D animation parametrization such as blendshape coefficients. This editing pipeline implies technical and artistic skills, as well as a considerable amount of time to end up with a coherent and satisfactory final animation.

In this chapter, we investigate an editing algorithm that allows modifying an animation while preserving the dynamics of the face. We develop a machine-learning-based approach trained to produce natural-looking animation from a small set of input parameters. By training on natural animation space-time patterns, our system learns to preserve the temporal coherency of the motion and ensure smooth and continuous animation.

Previous data-driven methods [SLS+12, ASK+12] propose a temporal solution to propagate the edits across the surrounding frames by solving a movement matching equation or by using a bilinear spatiotemporal model. Although these methods provide smooth results, their temporal resolution depends on hyperparameters (energy weights or frequency coefficients) that need to be manually adjusted rendering the editing task more difficult to tune. On the contrary, recent works have shown very promising learning-based methods in related animation, providing efficient solutions to automatically handle the complexity of the facial movements while yielding consistent animation.

As contributions such as Seol et al. [SLS+12], our system is meant to cope with constraints akin to real-world applications as shown in Figure 5.1. However, unlike Seol et al. [SLS+12] who focuses on producing an efficient system dedicated to a professional use, our goal is to provide an alternative solution for non-specialist users. For example, in the case of exaggerated user constraints, the system of Seol et al. [SLS+12] method generates implausible animations. We specifically study designs that are robust to inadequate user edits, and handle exaggerate or conflicting inputs. Besides, instead of complicated facial control parameterizations, we propose to use intuitive high-level control parameters as input to the system, such as specifying the distance between the lips over time. The system runs at low latency, enabling us to create a graphical interface for users to interactively modify the output animation until getting a satisfying result. Moreover, we offer the flexibility to the user to iteratively modify its animation, either by editing a few frames or by imposing full-sequence constraints, until a satisfying result is produced (see Section 5.3.4).

This work takes inspiration from the seminal work of Holden et al. [HSK16], which tackles the same challenge of editing an animation using simple high-level parameters. Using a fully convolutional learning-based human motion editing system, they map high-level control parameters to a learned body motion manifold presented earlier by the same authors [HSKJ15]. Navigating this manifold of body motion allows to easily alter and control body animations, while preserving their plausibility. Unlike body motion, however, one challenge when dealing with facial animation is to preserve the high-frequency patterns of the motion, as they are responsible for important

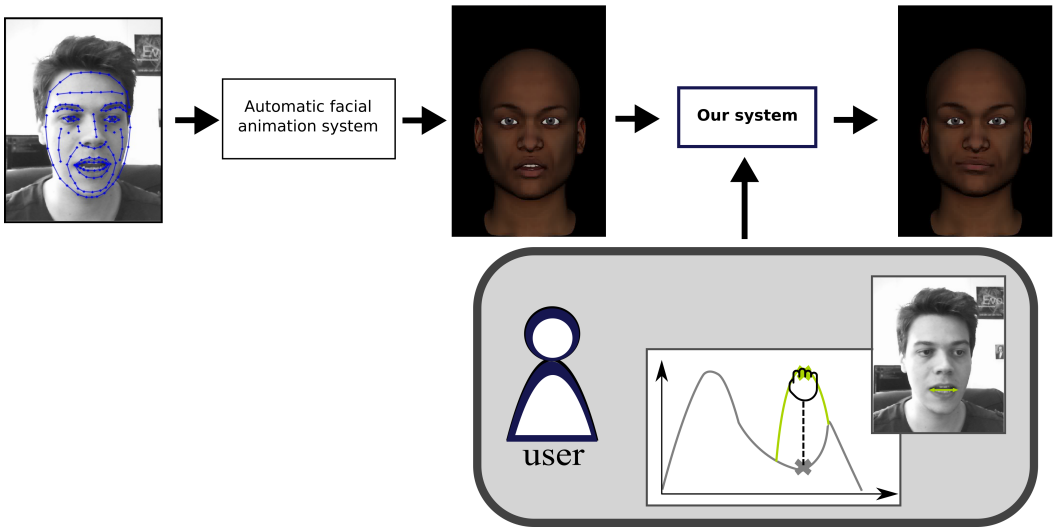


Figure 5.1 System overview. Our editing system allows a non-specialist user to easy and quickly interfere in the traditional facial animation pipeline to refine an animation.

communication cues (eye closures, lip contacts). Indeed, as pointed out by Seol et al. [SLS+12], ensuring an accurate mouth and eyelids closures at specific frames is of paramount importance to provide realistic facial animation. This is particularly true for learning-based solutions, that leverage large datasets of complex, possibly conflicting animation patterns [HSK16].

Although the system of [HSK16] demonstrated impressive results on body motion, we found that their architecture is not particularly suited to this particular aspect of facial animation. Among the shortcomings of these solutions is the ability to preserve the different frequency components of the animation and to adapt the behavior of the system to inconsistent inputs. Using it in our scenario leads to over-smoothed, unappealing facial animations, which we illustrate in Section 5.3.1.

Yet, Convolutional Neural Network models have genuine attractive properties including continuity and time invariance, making them reliable candidates to encode the temporal aspect of the motion. This structure progressively transforms local motion patterns through multiple layers, until producing the desired output animation and therefore, requires the full input signal to be specified to produce the desired output. When editing, the user often wants to see the entire motion at once before selecting motion segments to revise, and does not wish to modify early frames that will impact future motions. Hence, unlike in this previous chapter, user-driven editing only happens offline and thus, precludes a causal architecture. For these reasons, we adapt their convolutional-based approach for the purpose of facial motion, and tackle the high-frequency issue leveraging a resolution-preserving neural network. Exploiting the state-of-the-art architecture in high-resolution image processing, we build our

system on a one-dimensional fully convolutional network inspired by Ronnenberger and colleagues [RFB15], with skip connections between the down-sampling and the up-sampling parts designed specifically to preserve high-frequency details. Besides, we aim at a system that is resilient to coarse editing by non-specialist users. To that end, we propose to train an additional denoising autoencoder that we stack at the end of the network to ensure a natural-looking final animation output.

Previous learning-based motion editing works exploit a standard high-level body motion parameterization that includes joint positions and motion moment cues. However, there is no such counterpart motion representation consensus in the facial animation community. In this chapter, we suggest a new approach to code facial expressions through semantic high-level parameters for non-expert motion control.

In summary, this chapter presents the following investigations:

- A facial animation editing system based on convolutional neural networks, which enables to quickly edit a temporal talking facial animation with few intuitive control parameters. Based on a time resolution-preserving architecture, we experiment with an approach that can generate complex and plausible facial motion patterns. The proposed framework features a regressor designed to map low-dimensional control parameters to blendshape coefficients sequences. It is followed by an autoencoder meant to ensure the naturalness of the outputted animation sequences.

- A focus on robustifying produced animations, to be resilient to implausible inputs constraints. We use a denoising training strategy to improve the reliability of our system. The originality comes from the noise added to the indirect inputs used to train the stacked autoencoder, and an additional loss term encouraging mouth closure preservation during talking facial animations.

Chapter Overview The remainder of the chapter is organized as follows. We describe the input control parameters and detail our model in Section 5.2. We focus particularly on the benefit of the added autoencoder and the specific way of training it. We compared our system with related works in Section 5.3 and conducted several experiments highlighting the performance and benefits of our architecture. Finally, we demonstrate the usability of our framework in a realistic animation production pipeline before discussing the proposed approach in Section 5.4.

5.2 System Description

In this section, we describe our facial animation editing approach in more details. First, we discuss the choice of the control parameters which constitutes the input of our system in Section 5.2.1. Then, we elaborate on the structure of the neural network that forms the heart of our system. Our system shown in Figure 5.2 is composed of two neural-based parts. Both are fully convolutional, and operate on space-time signals, meaning they perform temporal convolutions on a time window

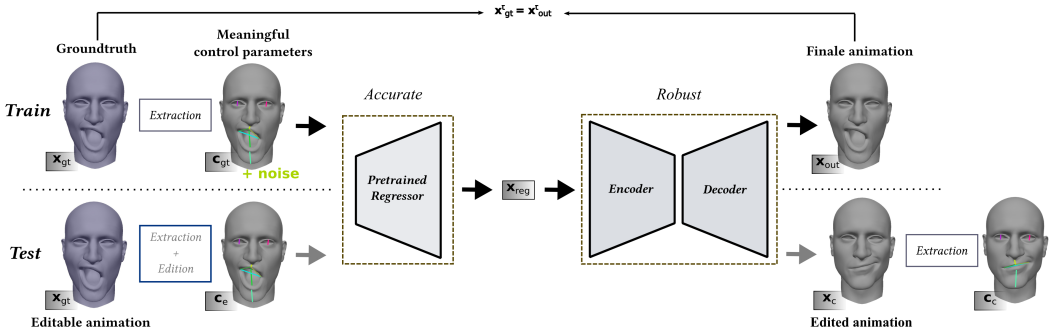


Figure 5.2 System description. (Top) At train time, fixing the parameters of the regressor, the autoencoder learns to reconstruct the initial blendshape weights from the noisy meaningful control parameters. (Bottom) At test time, the edited control parameters lead to an accurate blendshape weights sequence thank to the regressor. The stacked autoencoder is robust to inaccurate inputs and is trained to always output realistic edited animation.

of their input parameters. We briefly review this popular structure in Section 5.2.2. The first part of our system, presented Section 5.2.3, is a regressor which maps high-level inputs to a blendshape weights sequences. The second one is a stacked autoencoder that cleans the blendshape weight sequence to ensure a final realistic animation. We describe this second part in Section 5.2.4.

5.2.1 Meaningful high-level control parameters

Our investigations for controllable animation editing focuses intentionally on usability. The input of the proposed methods have to be meaningful, low-dimensional and easy to manipulate. From an application point of view, the user must be able to efficiently personalize a 3D animation without advanced animation skills. Thus, we consider a system that takes intuitive high-level parameters as input, with an explicit effect, so that users can easily perform their desired modifications into animation. Particularly important in facial animation is the rendition of speech, so we want the control parameters to be able to specify all plausible mouth shapes that occur during a natural speech.

Inspired by the work of Seol and colleagues [SSK+11], we choose inter-vertex distances with semantic meaning and a major effect on facial expressions, as control parameters. In their work, Seol and al. [SSK+11] propose to derive hierarchical GUI controllers from the original animation sliders. Classifying the blendshape according to their level of impact on the face into a three-layered layout, they define 12 *large* activation parameters shown in Figure 5.3: two for raising the eyebrows, two for the eyes openings, one nose-link blendshape, two for leading the mouth opening, one the mouth corners motion, and four controlling the chin and the jaw movement.

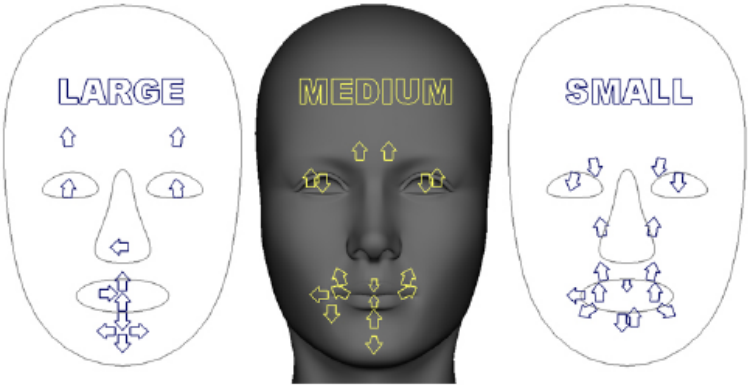


Figure 5.3 The various control parameters used by Seol and colleagues [SSK+11]. They create three layers of control, dividing the 44 blendshapes according to their activation levels.

However, these parameters remain blendshape-specific and cannot be derived on our template straightforwardly.

With particular attention paid to disambiguating the mouth movement and selecting a small number of controllers, we pick out eight control parameters shown in Figure 5.4. The horizontal and vertical inner-lips distances as well as the eyelids distances determine the state of the mouth/eye closure, two important expressive cues. To enable editing the emotional expressiveness of the animation such as modifying the smile intensity, we add the distance between the upper-lip center and the mouth corners. The lips protrusion, activated by pronouncing palate sounds such as "sh or ch" or doing a kiss shape are manipulated with the distances between the nose bridge and the upper-lip center and between the chin and the bottom-lip center. We found this to be a rather minimal set for our approach. Fewer parameters would result in ambiguous specifications for face shape, leading to a noisy regressor output.

To define their GUI controllers, Seol and coworkers [SSK+11] geometrically compute the optimal center of effect area and direction of the movement blendshapes. They design their arrow-shaped controller by fixing the z and use only the projection on the xy-plane to parameterize their controllers. In the same vein, we use the x or y projection of the vertices distances to give the user more intuition about the semantics of each control parameter. However, our control parameters are directly derived from the mesh of our template, and thus, do not depend on the blendshape model contrary to the mentioned work [SSK+11].

In this work, we always measure those distances on a blendshape-based character with fixed morphology $\mathcal{T}_W$. This ensures that the distance patterns we extract from the B3D(AC)^2 dataset's animations of the section 3.3 are actor-independent. In that way, any type and style of facial animation should be editable (see Section 5.3.2).

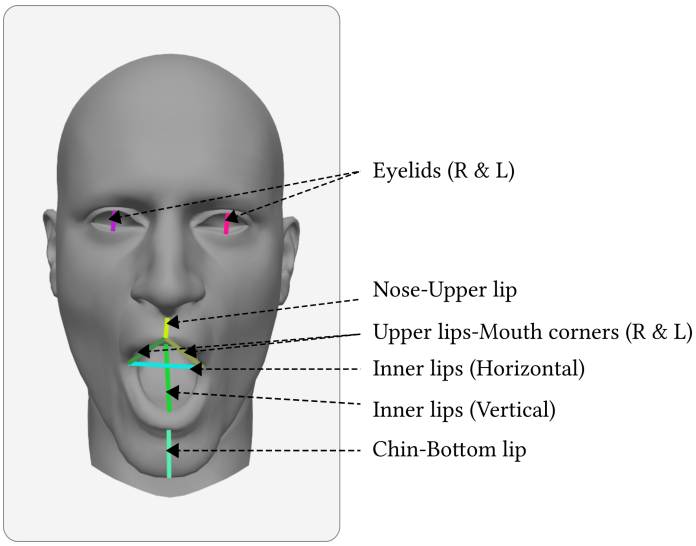


Figure 5.4 Eight inter-vertices distances, extracted from the mesh, used to compute our meaningful control parameters.

While our network can learn full-face motion patterns, we found that generalization of the results is improved if we split the facial controls in three groups that exhibit low motion correlation with each other in the database: lower-face, upper-face and eyelids. An independent network will be trained for each group, with its own relevant high-level control parameters as input, and appropriate blendshape coefficients as output. This splitting of the face is common in previous research works and practical applications [JTDP03, ZSCS04].

5.2.2 Convolutional Neural Networks for Motion Modeling

As explained in the introduction of this chapter (Section 5.1), this work builds upon state-of the-art motion editing work [HSK16]. In the domain of learning motion representation, an effective and proved technique to handle motion data is to use Convolutional Neural Network (CNN). The convolutional neural layer was initially introduced to process visual data [LBD+89], and basically consists of applying a convolving kernel, with learnable parameters, on local features, sliding over all the input data. The reduced number of parameters involved in this structure has allowed the processing of images with any size, achieving great success in the particular task of image classification [KSH12, SZ15]. CNN gets the attractive property to learn continuous high-level of reasoning from the input data, creating an informative representation of the motion data, such as a Motion Manifold [HSKJ15]. Although CNN was historically devoted to visual data, it has subsequently enabled effective time series modeling, demonstrating excellent results in speech recognition [GJ14] or natural language processing [CWB+11]. In Figure 5.5, we present the one-dimensional convolution principle used at the core of our neural architecture. By shifting along

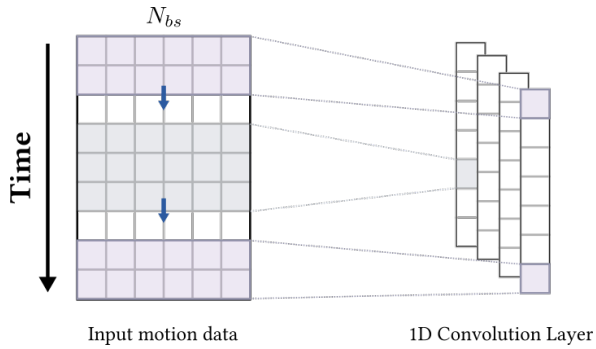


Figure 5.5 The one-dimensional convolutional layer. The convolving kernel is applied to temporal motion window to produce a new motion features and then moves along the time direction to process the next window. The convolution operation, followed by a nonlinear activation function, is computed from the beginning to the end to all the windows of the input motion sequence, leading to a more informative representation of the motion data. (Source [Kim14]).

the temporal axis only, a one-dimensional CNN presumes a temporal invariance. This assumption might not be valid in the spatial domain as the different parts of the face are strongly correlated and get their own temporal signatures. For instance, the underlying structure of the mouth strongly differs from the one of the eyes.

5.2.3 Regression from low dimension control parameters to blendshape weights

Motivated by the observation that facial animation is also composed of important high-frequency features, we moved away from previous motion-modeling network architectures and built a resolution-preserving neural network to regress the control parameters $\mathbf{c}_{gt} \in \mathbb{R}^{L \times N_{cp}}$ to blendshape weights $\mathbf{x}_{reg} \in \mathbb{R}^{L \times N_{feat}}$ as shown in Figure 5.2. L is the length of the input sequence, N_{cp} the number of control parameters, and $N_{feat} = 29$ is the size of the output blendshape vector. The control parameter coefficients have been calculated on a fixed morphology character, animated with the blendshape weights ($\mathbf{x}_{gt} \in \mathbb{R}^{L \times N_{feat}}$) extracted from the B3D(AC)ˆ2 database.

The regressor is a fully one-dimensional convolutional neural network with skip layers, a structure sometimes loosely described as U-net. Its architecture is depicted in Figure 5.6a. We use one-dimensional max-pooling layers and up-sampling layers to respectively down-sample and up-sample the temporal dimension. Each convolutional block in Figure 5.6 is composed of a batch normalization layer, a convolutional layer and the ELU activation function [CUH15]. As input to the regressor we use a time-window of 64 frames. We extract those windows from complete sequences with a time-overlap ratio of 0.75. As preprocessing, we subtract the mean controller values calculated on the whole trainset. All the filters in the network have a size of 3.

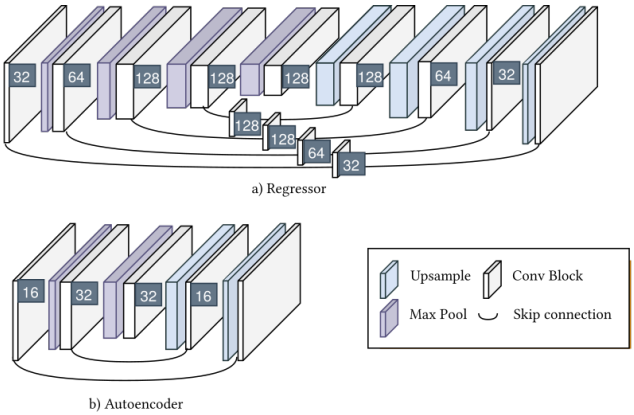


Figure 5.6 Architecture of the regressor (a) and the autoencoder (b)

Our loss function is composed of two terms [HSK16]: the mean square error (MSE) between the $\mathbf{x}_{gt}$ and $\mathbf{x}_{reg}$, $\mathcal{L}_{MSE}$, and a L2 regularization on the weights $\beta * \mathcal{L}_{reg}$.

We set the tradeoff parameter β equals to 1. We employ the Adam optimizer for training with a batch size of 128 and an initial learning rate of 0.001 with a decay ratio of 0.95 every five consecutive epochs with no validation loss improvement. We use sequences from 13 subjects of the dataset to train our network. This amounts to around 85 minutes of facial animation, which we split into a training set and a validation with a 0.95 ratio. We consider the final state of the network as the state at the epoch with the lowest validation loss.

5.2.4 Autoencoder for ensuring the naturalness of the animation

Our network features an animation autoencoder whose role is to clean-up the output of the regressor. Our regressor is a rather straightforward mapping network, so it will faithfully transcribe any user input, easily extrapolating to cases of unrealistic facial animation. On its own, such a regressor would not provide much value to non-expert users. It solely changes the parametrization with which animation is controlled, but it is not robust to exaggerations and inconsistencies. The added auto-encoder acts as a corrector that keeps the outputs in a realistic animation space. Its architecture is depicted in Figure 5.6b.

Ensuring that the network produces realistic animation is due to both the presence of the autoencoder and to the following denoising training strategy. Training autoencoders as denoisers -meaning feeding them with noised inputs and clean outputs- is common practice, but we found that the resulting autoencoder is very dependent on the noise characteristics. In our case, since the noise is supposed to mimic unrealistic inputs such as the one that a non-expert user might provide to the network, we found it difficult to find a good noise model. Instead we chose to train the whole end-to-end

system as a denoiser, while keeping the regressor weights constant (except the statistics of the batch normalization layers) and optimize the autoencoder's weights to reconstruct $\mathbf{x}_{gt}$. In practice, we modify around 20% of the control parameter inputs of the regressor $\mathbf{c}_{gt}$ (see Figure 5.2) with salt-and-pepper noise. We found that this creates noisy animation patterns that are closer to what the system would encounter in a real runtime scenario and therefore, forces the autoencoder to learn to clean-up inaccurate inputs. For the autoencoder to preserve the high-frequency features of the regressed output, we use a convolutional architecture similar to that of the regressor (Figure 5.6).

We train the autoencoder using the standard MSE loss, $\mathcal{L}_{MSE}$ applied on all blendshape coefficients. Such as our previous system presented in Section 4.2.3, we add a loss $\mathcal{L}_{distance}$, measuring the difference between some intervertices distances on the model animated with $\mathbf{x}_{gt}$ and $\mathbf{x}_{out}$. Finally, the autoencoder is trained to minimize the followed loss:

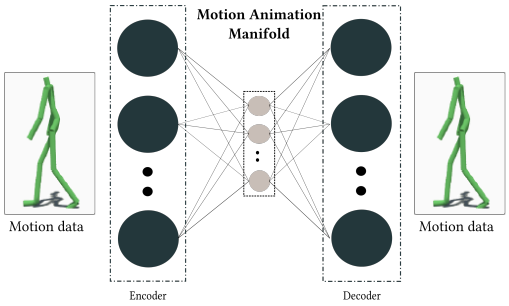
$$\mathcal{L} = \mathcal{L}_{MSE} + \alpha \mathcal{L}_{distance}. \tag{5.1}$$

Typically, $\mathcal{L}_{distance}$ measures the distances between the lips, and between the eyelids. This term helps ensure an accurate mouth closure during a talking facial animation [MD18]. For our experiments, the parameter α is set to 1. Training the model takes less than 2 hours on a NVIDIA GeForce GTX 1070 GPU.

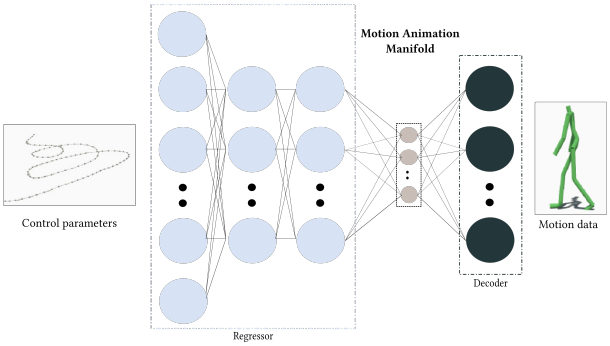
5.3 Experiments & Results

In this section, we present experimental results of our facial animation editing system. First, we evaluate our system by comparing its integrity to the recent related work of [HSK16], which addresses a similar set of requirements, albeit for body animation applications. We retained their system's architecture, adapting it for our specific inputs and outputs. The discrepancy between quantitative measures and qualitative look of the animations leads us to use special metrics for a more complete comparison (Section 5.3.1). This comparison confirms the suitability of the proposed neural network system for the purpose of facial animation, as well as its capability to create plausible facial animation preserving the complex dynamics of the facial movements.

To assess the data-dependency and reproducibility of our system, we apply it on a different recently released database (see Section 5.3.2) and measure quantitative performance. In Section 5.3.3, we study the robustness of our system to inaccurate user constraints, and analyze the role of the system's components. Finally, as our system runs with low latency, we demonstrate in Section 5.3.4 its potential as an interactive animation tool by showing examples of performance-based animation editing.



(a) Auto-encoder. [HSK16]. A manifold of realistic body-motion is learned through a fully convolutional auto-encoder.



(b) Regressor [HSK16]. After the creation of the manifold, the feedforward fully convolutional regressor is trained to map high-level control parameters to this motion manifold. Thereafter, the pre-trained decoder enables producing realistic animation from the control parameter code.

Figure 5.7 2-networks architecture introduced by Holden et al. [HSK16].

5.3.1 Comparison with state-of-the-art approach

Our system is designed for animation editing and control, but it will only be useful if its architecture can handle and represent sufficiently varied facial motion. Of particular interest is the ability to preserve the high-frequency components of facial animation, which are important for human communication. In practice, we evaluate how close the generated animation $\mathbf{w}_c$ is to ground-truth $\mathbf{w}_{gt}$ when the edited control parameters $\mathbf{c}_e$ are kept unchanged, equal to $\mathbf{c}_{gt}$ (see Figure 5.2). We evaluate this metric on the whole database using the leave-one-subject-out strategy.

To our knowledge, there is no work directly addressing the problem of high-level, temporal consistent manipulation of facial animation, precluding a direct comparison. In the broader field of animation research, Holden et al. [HSK16] set to tackle a similar set of goals for body animation editing and control. Part of their system is valid for facial animation and can be adapted to our inputs and outputs. Their 2-network architecture is presented in Figure 5.7.

To represent their approach, we first learn a time-convolutional autoencoder with

one layer to encode the sequence animation into a latent space and one layer to decode. Then, we learn a fully convolutional network to regress the control parameters to this latent space (see [HSK16] for details). The regressor is built with only 2 layers as it appeared to give better results in our case. To get a fair comparison, we train this system for each face area separately, using our same training strategy (see Section 5.2.1).

We evaluate the different systems by minimizing the mean square error (MSE) between the input and the output blendshape weights sequences. For our experiments, we use the regressor with the lowest MSE because the role of the regressor is to accurately regressed the control parameters to the blendshape weights.

Table 5.1 Quantitative comparison between the regressor and the full system on the test set.

	MSE (lower face)	MSE (eyelids)
Regressor only	0.0028	0.0064
Holden et al. [HSK16]	0.004	0.009
Our system	0.0082	0.0086

Interestingly, Table 5.1 shows that Holden et al. [HSK16] performs equally (eyelids) or better (lower face) than our complete system in term of MSE. However, by looking at the temporal curves of inner lips distance derived from $\mathbf{c}_{gt}$ and $\mathbf{c}_c$, we realize that their system smooths the motion signal and shows consequent loss of high-frequency components of the mouth and the eyes (Figure 5.8). While the reconstruction MSE is lower, the corresponding animation is qualitatively less appealing as it misses the key high-frequency communicational cues on the mouth and eyelids. Note that this behavior was probably less an issue in their original application on body animation, as high-frequency components carry less semantic weight in their case as it does for facial motion. In Figure 5.9, we display two frames extracted from sequences created from the same $\mathbf{c}_{gt}$ with the system of Holden et al. [HSK16] and our system. We can see that, while our system produces an animation with faithful mouth openings and closures, the animation resulting of their system misses these cues due to the smoothing nature of their architecture. More precisely, the blurriness of the output animation might be assigned to the simple depooling process used in their autoencoder [HSK16]. Examples of animations using both systems are shown in the supplementary video[1].

For a more representative quantitative comparison between our system and Holden et al. [HSK16], we propose to use a metric that highlights the capability to accurately retain facial animation cues such as mouth contacts, closures and eye

[1]Results presented in this chapter are shown in the supplementary video available at `https://elo-nsrb.github.io/homepage/publi_data/mig2019/video.mp4`

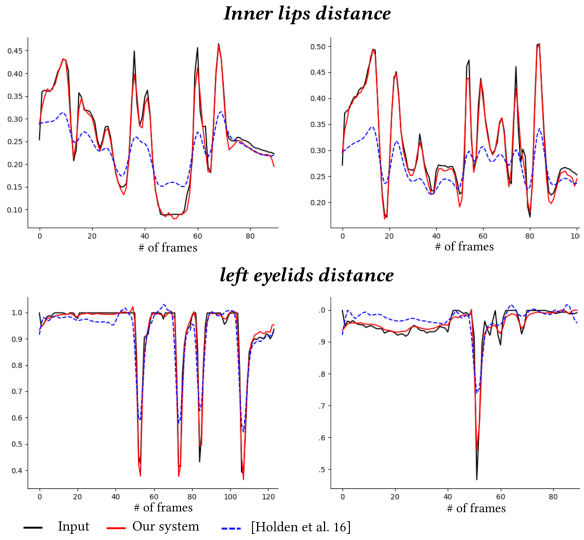


Figure 5.8 Comparison with Holden et al. [HSK16]: Curves of inner lips distance for different sequences. The body motion system [HSK16] smoothes the output signal loosing the high frequency components.

blinks. To our knowledge, there is no agreed-upon metric in the community for such semantic facial cues, so we suggest measuring a true positive rate (TPR), i.e. ratio of true positive mouth- (respectively eyelid-) closures to the number of actual mouth- (eyelid-) closures, and the false positive ratio (FPR) defined as the ratio of false positive mouth- (eyelid-) closures to the actual mouth- (eyelid-) closures. The TPR measures the capability of the system to accurately preserve the desired mouth- and eye-related conversational cues. The FPR controls that the system does not hallucinate undesired such movements. On Figure 5.10, we plot the TPR and the FPR for the mouth and right eyelid closures with respect to the threshold of detection. We can see that for lower thresholds, only our system creates consistent mouth/eye closures as its TPR is always the highest. The system of Holden et al. [HSK16] is not capable of producing eye closures so its FPR is zero for lower thresholds. Meanwhile, we control that our system does not hallucinate motion as its FPR remains low.

An interesting aspect to monitor is the ability to model immobility, that we observe here on the first curve plotting the inner lips distance in Figure 5.8. Between the 40^{th} frame and the 60^{th} frame, we can observe that our system can cope with no inner lips movements for multiple consecutive frames.

5.3.2 Data dependency: transfers on another database

As with all data-based approach, it is important to know how the approach depends on the size and content of the dataset. Thus, we test the validity of our model (trained with the B3D(AC)^2 dataset) on the recently released Vocaset database [CBL+19], using the inference scheme exhibited in Figure 5.11. This dataset is composed of

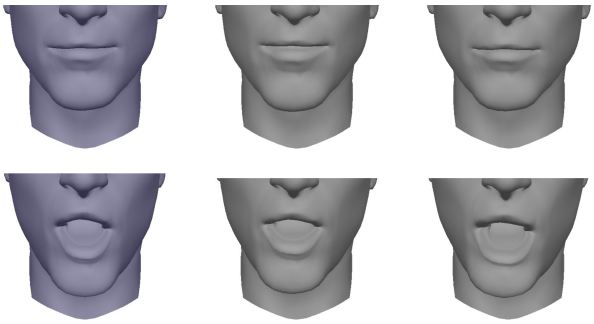


Figure 5.9 The ground-truth (left). Compare to [HSK16] (middle), our system (right) is able to generate an animation which faithfully respects the input mouth movements and its amplitude.

Table 5.2 Quantitative results of our system trained with the trainset of the B3D(AC)^2 dataset.

Trainset	Testset	MSE (mouth)	TPR	FPR
Vocaset	Vocaset	0.038	0.87	0.06
Vocaset	B3D(AC)^2	0.05	0.81	0.38
B3D(AC)^2	Vocaset	**0.004**	**0.98**	0.22
B3D(AC)^2	B3D(AC)^2	0.008	0.98	0.22
Both	Vocaset	**0.003**	0.95	0.22
Both	B3D(AC)^2	0.01	0.95	0.35

sequences of 12 subjects speaking sentences from the TIMIT corpus [Gar93]. We use the same processing pipeline to get the blendshape coefficients sequence as in Section 3.3 except that we do not use 2D information. We downsample the frame rate to 25 fps to match the frame rate of our dataset B3D(AC)^2.

As shown in Table 5.2, our system trained with only the trainset of the B3D(AC)^2 dataset and applied to the whole Vocaset gives a comparable MSE (0.004) as a one trained with both the Vocaset and B3D(AC)^2 dataset (0.003). The Vocaset content is less diversified, that is why the results obtained using only this dataset are the lowest. Indeed, there is no emotional sequence in this dataset unlike in the B3D(AC)^2 dataset which is one-half composed of emotional sequence. In such sequences, the amplitude of the movements is generally higher compared to neutral sequences. So, at test time, it is easier for a system trained with emotional content to render neutral speech content than the other way around. We can see on the supplementary material that our system is suitable to model any new subjects in the Vocaset.

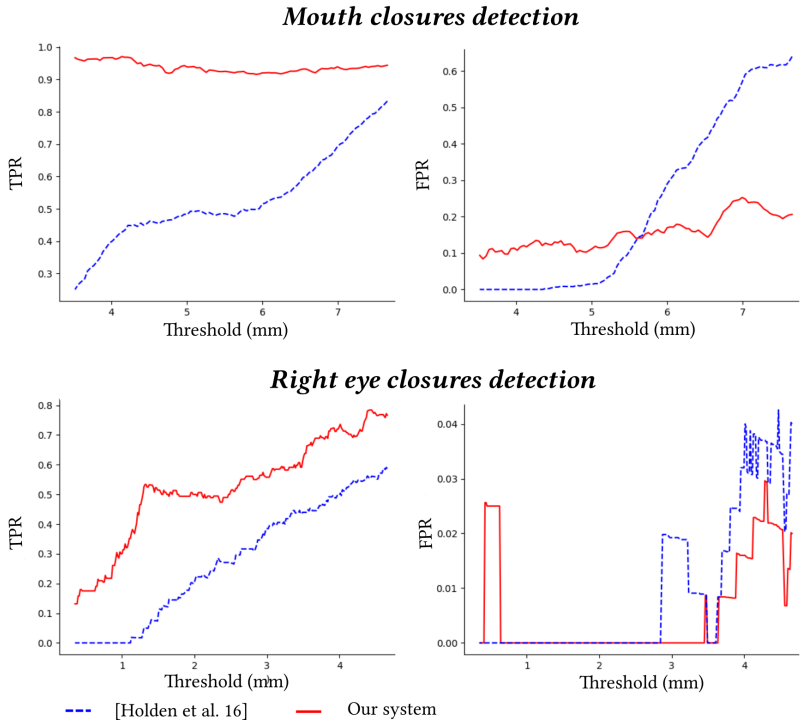


Figure 5.10 Comparison with [HSK16]: Curves of the TPR and the FPR of the mouth and eyes closures on the testset.

5.3.3 System Robutness: necessity of the autoencoder

Here we evaluate the robustness of our system by its ability to handle inaccurate input. We show that using the regressor alone would be more accurate than the full system in terms of MSE as shown in Table 5.1. However, without the autoencoder, the regressor alone would be too sensitive to user's inputs, leading to unrealistic animation output as soon as input control parameters do not match a realistic animation. The regressor handles the accuracy of mapping from control parameters to blendshape animation, while the subsequent autoencoder keeps the resulting animation inside the space of plausible animation. Both components are essential for a system aimed at non-specialist users. We show this by inputting different mouth-opening constraints and looking at inner-lips distance at output. One example of the edited curves is shown in Figure 5.12 and the corresponding output can be seen in Figure 5.13. We can see that the regressor is unstable; as soon as the input constraints constitute an unrealistic facial pattern, the output shapes are unrealistic. The autoencoder cleans up the output animation of the regressor, generating a natural animation. For instance, it projects unrealistic mouth openings to realistic ones when it is required. Note that this is not just a geometric projection operation but a temporal one as well, as our autoencoder models time-windows of animation. More results on full animations are provided in the supplementary video.

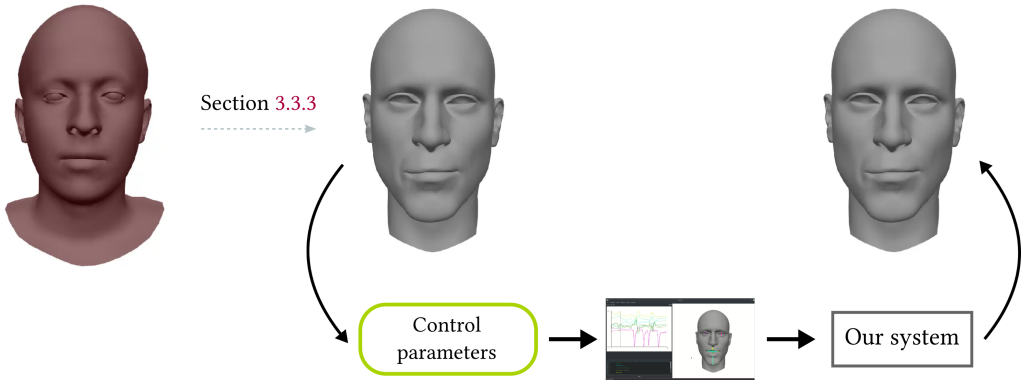


Figure 5.11 We evaluate our system on the recently released vocaset [CBL+19] using the following inference scheme: first, we extract the continuous control parameters from the tested animation and pass these temporal features through our system. Then, we observe the resulting animations and compare it to the original animation.

5.3.4 Usability: integration in a traditional facial animation pipeline

Even if our system processes whole sequences of animation, its architecture is light and performs network inference very quickly. This renders interactive uses of such a system imaginable. In this work, we propose an interactive editing tool that is meant to be easily integrated in a facial animation pipeline that would enable non-specialist users to generate quality facial animation. A common modern performance-based facial animation pipeline consists in acquiring sequences of actor performance, tracking his/her facial expressions, retargeting those to blendshape animation coefficients, and finally manually tuning the obtained animation. Today, real-time face tracking methods enable non-experts to get raw facial animation from simple video feeds, but the animation is often noisy. Moreover, as in professional pipelines, the animation must often be edited later on to match the artistic intent. Our tool finds its place at the editing stage of the pipeline. Through an interactive interface, the user can continuously refine the animation to produce the desired animation with low-latency. A screenshot of this interface is shown in Figure 5.14. Indeed, the inference time, time between the moment the user applies its new control parameters and the moment the new final animation is produced is on average less than 0.015s for a typical scene of 8 seconds (202 frames) on CPU.

To showcase this, we use an off-the-shelf real-time face tracking software [Dyn19a] that outputs blendshape coefficients. We developed a user interface that enables to visualize temporal curves for our control parameters and edit them via click-and-drag. Our network then runs inference to deliver the edited facial animation at an interactive rate. One can for instance change a neutral speech animation sequence by increasing the mouth corners distance, causing the character to smile while speaking. The figure 5.15 shows a frame with the 2D tracking landmarks, the corresponding

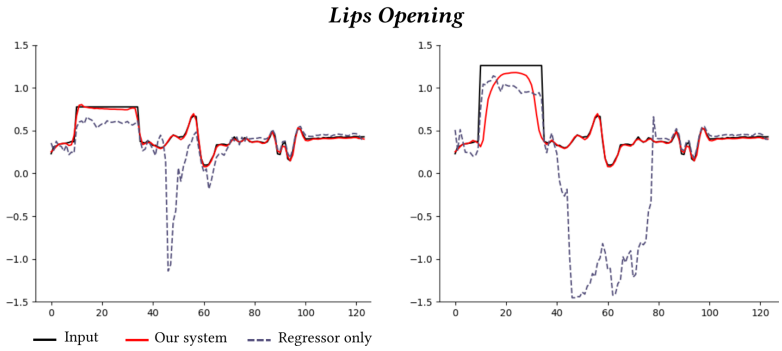


Figure 5.12 Realistic (left) and unrealistic (right) mouth opening input signal and the corresponding output with our system with and without the autoencoder. We can observe that the regressor alone is too sensitive to the input : unrealistic patterns appear as soon as a unseen input is given.

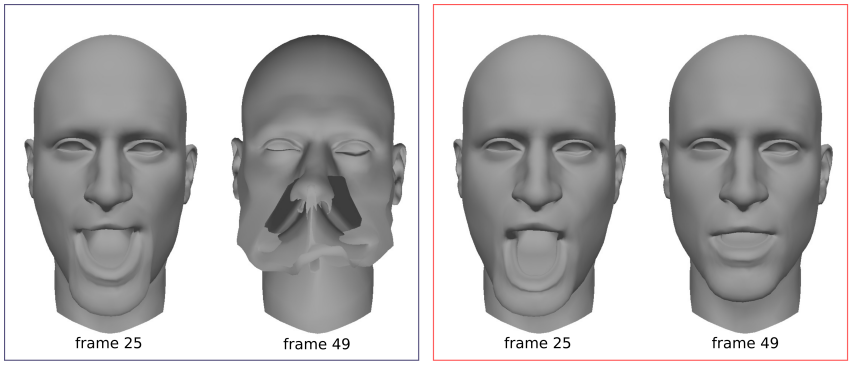


Figure 5.13 Output animation with an unrealistic mouth opening without (left) and with the autoencoder (right).

animation given by the tracking as well as the final edited animation with a smile. More isolated edits can be performed such as forcing a mouth closure or a blink by acting on the relevant local frames. Dynamic results of such edits are presented in the supplementary video.

5.4 Discussion

In this chapter, we present an interactive and robust approach to provide a convenient follow-up facial animation editing approach. Although data-driven approaches are not new in the field of motion editing [SLS+12, ASK+12], previous methods require manual intervention setting-up hyperparameters to enable accurate facial motion processing and handle the large range of facial motion frequencies. The study we have conducted leads to neural network proposition that takes care of the temporal consistency of the edited segment with the rest of the animation, ensuring natural transitions at its

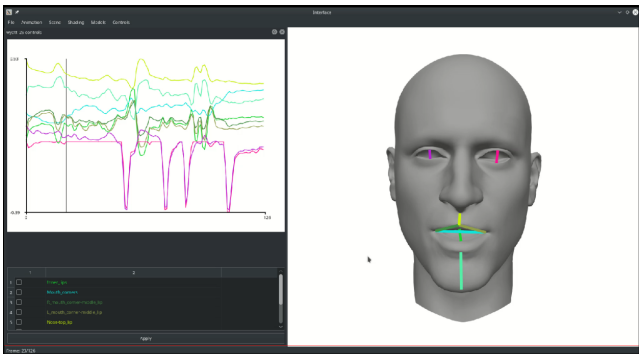


Figure 5.14 Screenchot of our interactive interface, enabling a non-expert user to continuously refine the animation and produce the desired animation with low-latency.

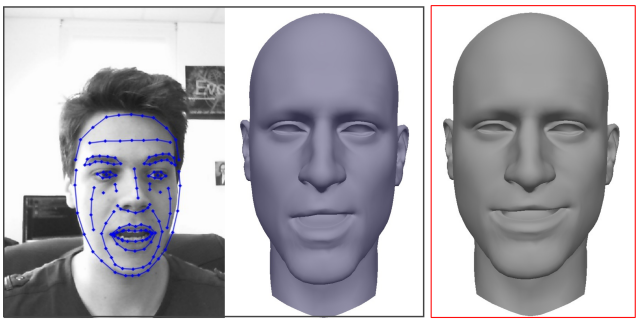


Figure 5.15 Editing of a performance-based animation: example of frame with 2D tracking landmarks, the animation given by the tracking software and the final edited animation. We change a neutral speech animation sequence by increasing the mouth corners distance, causing the character to smile while speaking.

boundaries. Using recent successful engineering deep learning tricks, our system upholds the essence of the facial animation throughout its processing. We have studied the behavior of our system by evaluating quantitatively on the error and semantic metrics versus relevant previous work, and have experimented with different datasets. In particular, we have demonstrated the necessity of using resolution-preserving architecture neural network to retain the temporal high-frequency information of facial motion, which architectures from previous work did not address.

This study also leads to the development of an experimental tool that can be used by non-specialist users to complete their facial animation pipeline, without manual data preprocessing such as labeling, alignment or motion segmentation. Indeed, the proposed design enables processing high abstractions of the animation signal, letting the users to quickly correct and modify portions of animation with any length, at any point in the timeline. The low latency at the inference allows instant

feedback on the resulting modified animation, increasing the speed and the efficiency of the editing task. Besides, our method is content-independent and emphasizes robustness, resulting in an editing tool that outputs plausible animation even when given imprecise or unrealistic user inputs.
We feel these results represent a step forward reducing the time-consuming task of facial animation editing while weighing up the usability of the solution.

A potential end goal this work wishes to address it to broaden the facial animation authoring to a less restrictive audience than actual editing tools. As our solution is devoted to large-scale facial editing only for non-expert usability purposes. A future direction could be to enhance it with a fine-scale details restoration solutions [BLB+08, BBA+07, MJC+08] or to integrate this system into a hierarchical or evolutionary process, starting from a coarse motion manipulation to precise motion refinement [RC19, SSK+11].

We note that the performance of our system strongly depends on the choice of the control parameters. More parameters result in a more accurate but less intuitive system that is harder to manipulate. Conversely, few parameters cause ambiguity in mapping controllers to facial shapes, resulting in less control over the produced animations. As an example, the sidewise motion of the chin is lost due to the lack of dedicated controllers (see Figure 5.16).

Rigging face model is a complex problem that leads to sophisticated structures involving shape deformers and blendshapes, enforcing multiple trial and error manipulations before achieving a satisfying one. Besides, a relevant low-dimensional face representation has to be informative enough to allow an efficient and unambiguous inference, producing realistic animation.

While body-motion can be intuitively and efficiently steered through dynamic high-level parameters such as the trajectory of the skeletal or the position and the angles between joints, such a relevant high-level and low-dimensional representation of the face remains an open question in the research community. Moreover, as previous motion editing architecture, the proposed framework is based on Euclidean frame-wise training metrics and relies on temporally dense inputs to guide motion signals explicitly at every frame.

The nature of the input and the definition of good training metrics that relevantly assess the quality of the edited animation, are hard problems that cannot be solved incrementally with current network architectures. In the next chapter, we explore alternative motion controllers and generative framework, in which the animation output, intended be human-like and plausible, is validated by the model. We study the possibility of providing discrete inputs, even semantic ones -such as phonemes- to control the generated animation leveraging the well-known GAN paradigm [GPAM+14].

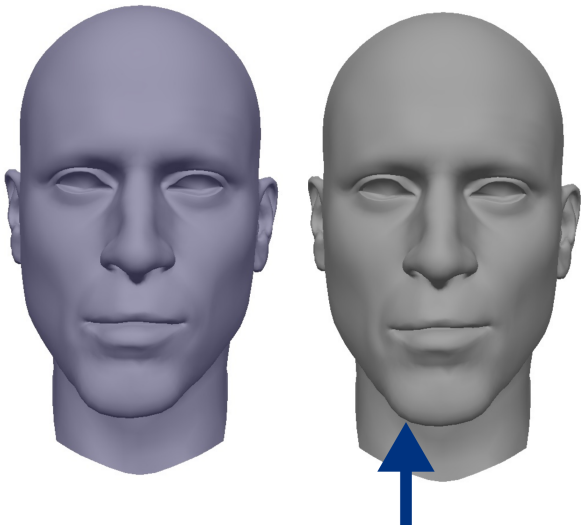


Figure 5.16 Limitation of our system: some motion such as the sidewise motion of the chin of the ground-truth (left) is lost at the output of our system (right).

CHAPTER 6

Generative Facial Animation Editing

Contents

6.1 Introduction

In the previous chapter, we presented a learning-based method to perform temporal animation editing, providing spatiotemporal constraints on meaningful vertices distances. However, this work suffers from several limitations, inherent to the chosen approach. The previous work relies on a regression model, imposing explicit and temporally dense constraints at each frame to guide the motion generation. This does not integrate well with the keyframing editing process, that is particularly cherished by artists in production. Moreover, a regressive model rests on a deterministic training strategy, assuming an available target motion sample and relevant metrics to weight the quality of the produced output at every frame.

Building upon our previous works, we propose in this chapter, a new point of view of the editing task through a generative framework. The perspective in viewing animation editing that we develop in this section originates from the parallel we draw between editing an animation and performing image inpainting (see Figure 6.1). Image inpainting aims at replacing unwanted/missing parts of an image with automatically generated pixel patterns, so that the edited image looks realistic. In animation editing, we pursue the same objective, substituting 2D spatial pixel patterns for 1D temporal motion signals. For a long time, image inpainting solutions have relied on low-level signal-based techniques such as image gradient preservation [PGB03]. While tremendously efficient, this approach reaches limitations when targeting larger image areas and rich image content. More elaborate neural-based generative models have recently shown their ability to fill-in gaps with complex image patterns and generate state-of-the-art results from little to no inputs. We share similar constraints and objectives as those recent studies, albeit in a different editing domain. We focus on the generation of complex believable motion patterns, given sparse, high-level inputs. We therefore investigate a machine-learning approach relying on GAN framework developed by [GPAM+14], that generates unsupervised plausible motion segments to replace missing, damaged, or unsatisfactory animation segments. Instead of comparing the output animation to a defined target one as in a regression set up, the adversarial learning strategy evaluates whether the produced animation is plausible and "real", allowing undeterministic motion generation.

This enables going beyond standard interpolation techniques, when no input animation signal is available to guide the result. Due to their simplicity and execution speed, several automatic motion completion systems have been developed based on simple interpolation between user-specified keyframes [Par72], usually with linear or cubic polynomials. Yet, while interpolation has proven efficient for short segments with dense sets of keyframes, the smooth and monotonous motion patterns they produce are far from realistic facial dynamics when used on longer segments. In many cases where the capture process has failed to produce an animation (occlusion, camera malfunctions), no input animation signal is available to guide the result. While standard interpolations are inefficient, the GAN paradigm enables producing realistic motion segments, filling gaps automatically without input guidance.

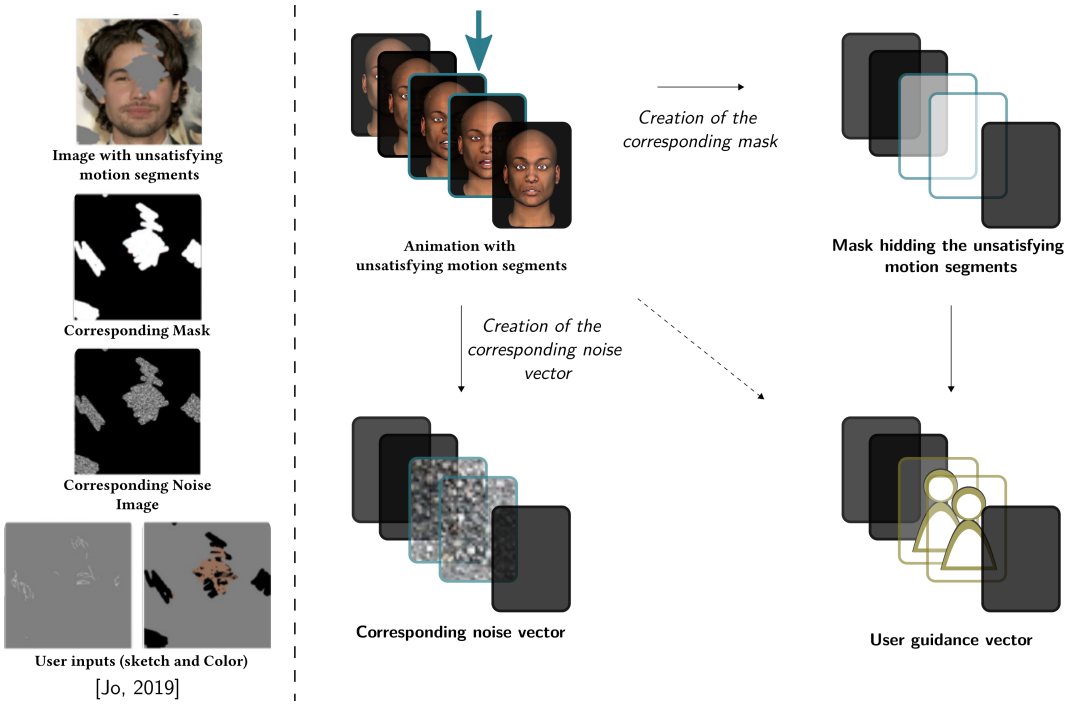


Figure 6.1 At the origin of this work is the parallel we draw between editing an animation and performing image inpainting. The goal is to replace unsatisfying parts with newly generated realistic ones. To do so, we derive a corresponding mask and noise vector from the input animation, in the same way as it is done in most image inpainting frameworks, and consider user's inputs to steer the new motion segment generation.

The GAN system consists of a generator intending to create plausible sequences in designated segments in the input animation, and a discriminator ensuring that the generated animation looks realistic. To cope with learning the complex temporal dynamics of the facial motion, we suggest designing our generator as a bidirectional recurrent architecture, ensuring both past and future motion consistency. Moreover, this architecture lends itself to the offline aspect of the user-driven editing. The proposed framework provides the user with control over the edited animation through high-level guidance, just as sketches enable semantic image manipulation in image inpainting scenarios [JP19], for intuitive and flexible animation editing. Leveraging the GAN improvement proposed by Mirza et al. [MO14], the standard GAN training is extended to allow different guiding signals including discrete or semantic user-guidance such as keyframes, noisy signals, or a sequence of visemes for supervised animation editing. Our approach reduces both the time and the manual work currently required to perform facial animation editing, while retaining the flexibility and the creativity properties of the current tools.

In summary, our primary investigations revolve around designing a multifunc-

tional framework that handles various high-level and semantic constraints to guide the editing process. We consider many editing use cases, such as long occlusions, expressions adding/changing, or viseme modifications.

Chapter Overview The remainder of the chapter is organized as follows. We describe the proposed methods in Section 6.2, motivating the original parametrization and the generative learning strategy that we use. We demonstrate the relevance of the developed framework to render realistic animations in various editing scenarios in Section 6.3 and conduct several quantitative evaluations highlighting the performance and benefits of this architecture in Section 6.4. Finally, we discuss the proposed approach in Section 6.5.

6.2 A Generative Editing Framework

Our goal herein is to train from data a generative neural network capable of generating plausible facial motions given different kinds of input constraints such as sparse keyframes, discrete semantic input, or coarse animation. In this section, we describe the parametrization of our system with different constraints, enabling supervised motion editing (Section 6.2.1). Then, we detail our system based on the well-established GAN minmax game (Section 6.2.3), briefly introduced beforehand in Section 6.2.2, as well as the training specifications. An overview of our system is depicted in Figure 6.2

6.2.1 Parametrization of our system

As motivated in the chapters 3, we parameterize facial animations with the highly popular blendshape representation, common throughout academia and the industry [LAR+14]. We develop a framework similar to the image inpainting ones [YLY+19, JP19]: more precisely, we consider an analogous training strategy for our networks. We feed our generator, G, with an incomplete animation, a noise vector, and a mask and optionally a discrete, noisy, or semantic input guiding the editing process. At training time, the incomplete animation $\mathbf{X}_i \in \mathbb{R}^{L\times N}$ corresponds to the original ground-truth animation $\mathbf{X}_{gt} \in \mathbb{R}^{L\times N}$ with randomly erased segments signaled by the mask. Both the original and the incomplete input animations consist of the concatenation of $L = 200$ frames of $N = 34$ blendshape coefficients. The mask $\mathbf{M} \in \mathbb{R}^{L\times N}$ encodes locations of erased segments (all blendshape coefficients) for a random number of consecutive frames. The input animation can be expressed as $\mathbf{X}_i = (\mathbf{1} - \mathbf{M}) \odot \mathbf{X}_{gt}$. $\mathbf{M}$ is a matrix with zeros everywhere and ones where blendshape coefficients are removed, and $\mathbf{1}$ is an all-ones matrix of size $L \times N$. The number and the length of masked segments in the input animation are chosen randomly, such as at test time our network can edit both short and very long sequences. At test time, masked segments are placed by the user to target the portions of the input sequence to edits. We note that our network can also generate an animation by

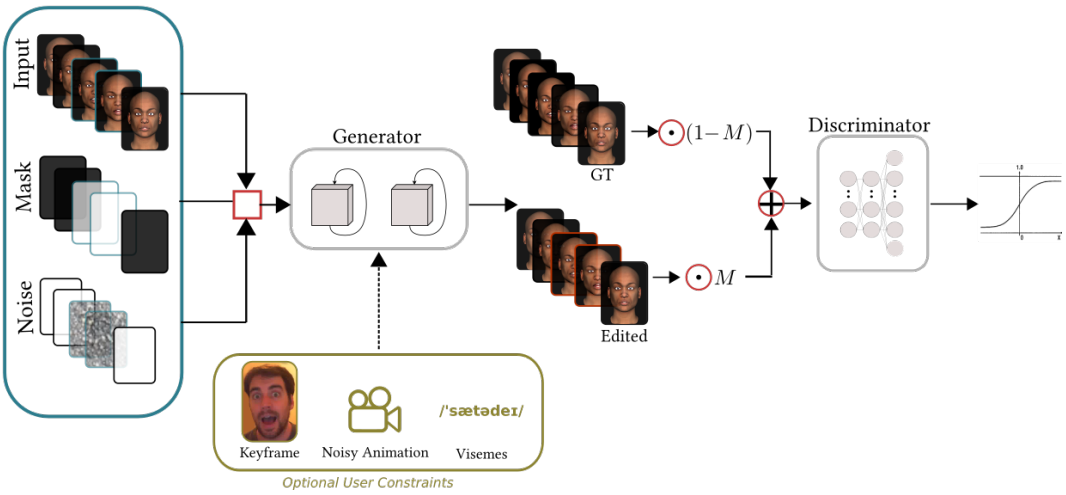


Figure 6.2 Framework overview. We build our editing tool upon a GAN scheme, using an approach similar to image inpainting. We feed the generator with a mask, an incomplete animation and a noise vector, eventually we add contraints such as sparse keyframes, a noisy animation or sequence of viseme. The generator ends-up with the completed animation. The discriminator has to distinguish between real animation and fake ones: it is supplied with the ground-truth animation and the generated one (the partial ground-truth sequence completed with the generate parts).

using a mask covering the full sequence. The vector of noise, $\mathbf{z} \in \mathbb{R}^{L\times 1}$, is composed of independent components drawn from Gaussian distribution, with 0 mean and a standard deviation of 1. We use the same framework for different editing scenarios and train a different network for each editing input type. Our framework can also perform unguided motion completion in missing segments, which is useful in the case of long occlusions for instance. In many cases though, the animator/user wants to guide the edit; so we focus on employing our framework for supervised motion editing. To achieve this, we leverage the conditional GAN (CGAN) [MO14] mechanism to add semantic guidance to our system. We concatenate a constraint matrix to the input, $\mathbf{C}_i = \tilde{\mathbf{M}}_i \odot \mathbf{C}_{gt,i}$, with non-zero components where animation has been erased. $\mathbf{C}_{gt,i} \in \mathbb{R}^{L\times N_{feat_i}}$ encodes the i^{th} constraint vector of N_{feat_i} features over time. $\tilde{\mathbf{M}}_i \in \mathbb{R}^{L\times N_{feat_i}}$ is the constraint-specific mask matrix, with zeros everywhere and ones at the same frame indices as $\mathbf{M}$. The constraints $\mathbf{C}_{gt,i}$ can be a sparse matrix with keyframes, a noisy animation signal, or one-hot vectors representing pronounced visemes at each frame. Each constraint conditions the training of the corresponding specific system. We consider three high-level constraint types enabling animation editing for several use cases:

- *Keyframes*: One main cause of animation editing is expression modifications, such as correcting the shape of the mouth or adding new expressions. Hence, we add sparse keyframes extracted from the ground-truth animation as constraints.

Table 6.1 Groups of phonemes.

Visemes	Phonemes	Visemes	Phonemes
sil		G + K + H	g, k, q, ɢ
AO + OY	a, ɔ	L + N + T + D	l, n, t, d, ʟ, ɬ, ɾ
AA + AE + AY	æ, ɑ	S + Z	s, z, ɣ
EH + EY	e, ɛ, eɪ	Sh + Ch + Zh	ʃ, ʧ, ʒ
IH + IY + EE + IX	i, ɪ, ɨ	TH + DH	θ, ð
OH + OW	o, ɒ	F + V	f, v
AH + ER	ʌ, ə, ɚ, ɝ	M + B + P	b, m, p
UW + AW + UH	u, ʊ, aʊ	W	w, ʍ
JH	j, ʤ	R	ɹ

The time between two keyframes is chosen randomly between 0 and 0.8 seconds.

- *Noisy Animation*: Our system enables the user to change the content of the animation and guide it with a coarse animation, such as one obtains from consumer-grade motion capture on consumer devices (webcam, mobile phone, ...).

- *Visemes*: We also consider a more semantic editing use case, such as speech corrections from audio. We use an audio-to-phoneme tool to obtain annotation in phonemes of each sequence in the database. In this work, we use the Montreal-Forced-Aligner [MSM+17], but any audio-to-phoneme tool can be used. We constrain our network with a one-hot vector representing the visemes at each time. A viseme is the visual facial representation of a group of phonemes. We group all phonemes in 18 classes of visemes presented in Table 6.1.

6.2.2 Generative Adversarial Network

We aim to realistically synthesize new motion segments that look like the real performance-based animations. Technically, we wish to generate one-dimensional animation vectors, with a temporal dynamic akin to real physical movements. This issue cannot be solved with regression-based learning techniques, that aim at finding the output data closed to well-defined target data, as explained in Section 3.1.1. Faithfully creating data, with a plausible underlying dynamic process lying in a high-dimensional space, is a challenging task that generative models wish to solve.

The purpose of generative algorithms is to generate sequences of numbers whose distribution is akin to a real (unknown) one. Among other techniques, one can find neural-based generative models, which learn the transformation function, mapping

a sequence of random numbers (following a known distribution) to a sequence of numbers with the desired properties. One early solution, the *direct* learning strategy, consists of explicitly comparing the generated distribution to the real one. Yet, it assumes a tractable and well-defined target process. This issue was ingenuously replaced by the *indirect* approach in the deep learning community, with the use of a new learning paradigm, the Generative Adversarial Network (GAN), introduced by Goodfellow and colleagues in 2014 [GPAM+14], that allows handling the case of intractable target distributions.

In that case, the output distribution is not directly compared to the real one, but is enforced to be close to the real one using an adversarial scheme. Its attractive generative property coupling with its striking success have motivated the investigation of the GAN principle for the animation editing purpose.

As in any GAN framework, our system is composed of two *adversarial* neural networks: the *generator*, designed to fill the timeline with realistic animation, and the *discriminator* intended to evaluate the quality of the generated animation, and to detect whether it is real. This game compels both networks to iteratively progress toward the optimal equilibrium in which the generator fools the discriminator, producing samples indistinguishable from the original ones; while the discriminator classifies as real the "true" and the "generated" samples with the same probability.

6.2.3 Framework details

Our generator, G, has to learn the temporal dynamics of facial motion. We use a recurrent architecture for our generator, as sharing parameters through time have demonstrated impressive results in modeling, correcting, and generating intricate temporal patterns. Our generator uses a Bidirectional Long Short-Term Memory (B-LSTM) architecture for its capability to adapt to quickly changing contexts yet also model long-term dependencies, properties well-demonstrated by the research community. Our generator consists of a sequence of N_{layers} B-LSTM layers (N_{layers}=2) with a stacked final dense output layer to get dimensions matching the output features. The recurrent layers consist of 128 hidden units. The main goal of the generator is to create plausible animations, i.e., to fill a given timeline segment with realistic motion signals that smoothly connects to the motion at the edge of the segment.

Our discriminator, D, has to learn to distinguish between a generated animation and a one produced by ground-truth motion capture. Because we want our generator to create an animation that blends well outside its segment, we supply our discriminator with the entire animation rather than only the generated segment, and choose a convolutional structure for D. Some elements have a higher impact on the quality perception of a facial animation. For instance, inaccuracies in mouth and eye closures during speech or blinks are naturally picked up as disturbing and unrealistic. Thus, we enrich the discriminator's score with relevant distance measurements over time that matches those salient elements. Our discriminator's structure is inspired by previous works [YLY+19, JP19]. It is a sequence of 4 convolutional layers, followed by spectral normalization [MKKY18], stabilizing the training of GANs. Over the

convolutional layers, we stack a fully connected layer predicting the plausibility of the input animation. The convolutional layers get a kernel of size 3, scanning their input with a stride of 2, and end up with respectively 64, 32, 16, 8 channels. We use the LeakyRelu activation function [XWCL15] after every layer except the last one.

6.2.4 Training methods

Classically, to train the proposed system we consider the minmax game between the generative and the discriminative losses. GAN frameworks have been reportedly hard to train, suffering from many pitfalls: the unsecured convergence of the system, the generator mode collapse, the unbalanced training between the two networks or even the hypersensitivity of the system to hyperparameters. Defining a functioning learning strategy is still an open question. Many solutions have been proposed since the initial work of Goodfellow and colleagues [GPAM+14], a comprehensive discussion on the different solutions can be found in [KAHK17].

In practice, we observe high quality results by leveraging state-of-the-art system developed in the Computer Vision field. Strongly inspired by [JP19, YLY+19], we design our generative loss as a sum of three terms. Our generator has to reproduce the input animation outside masked segments faithfully. Thus, we define a first loss ensuring accurate animation reconstruction:

$$\mathcal{L}_{feat} = \alpha_{gt}(\mathbf{1}-\mathbf{M}) \odot |G(\mathbf{X}_i) - \mathbf{X}_{gt}| + \mathbf{M} \odot |G(\mathbf{X}_i) - \mathbf{X}_{gt}| \tag{6.1}$$

The blendshape representation weights equally salient shapes such as shapes controlling eyelid closure and shapes with minor effect such as the one affecting the nose deformation. Therefore, as in our previous investigations, we add a loss $\mathcal{L}_{dis}$ (Equation (4.8)), to focus preservation of some key inter-vertices distances between the estimate and the ground truth animations. This loss encourages accurate mouth shape and eyelid closure, crucial ingredients for realistic facial animation. Finally, the generator is trained to minimize the following loss:

$$\mathcal{L}_G = \mathbb{E}[1 - D(G(\mathbf{X}_i))] + w_{feat}\mathcal{L}_{feat} + w_{dis}\mathcal{L}_{dis}. \tag{6.2}$$

At the same time, we train our discriminator to minimize the adversarial loss. We force the discriminator to focus on the edited part by feeding it with a recomposed animation $\mathbf{X}_{rec}$, which is the incomplete input animation completed with the generated animation, i.e, $\mathbf{X}_{rec} = (\mathbf{1}-\mathbf{M}) \odot \mathbf{X}_{gt} + \mathbf{M} \odot G(\mathbf{X}_i)$. We also influence the discriminator attention by providing it the key intervertices distances mentioned earlier. We add the WGAN-GP loss [GAA+17], $\mathcal{L}_{gp} = \mathbb{E}[||(\nabla_{\mathbf{U}} D(\mathbf{U}) \odot \mathbf{M}|| - 1)^2]$ to make the GAN training more stable. In this formula, $\mathbf{U}$ is a vector uniformly sampled along the line between discriminator inputs from $\mathbf{Y}_{gt}$ and $\mathbf{Y}_{rec}$, i.e, $\mathbf{U} = t\mathbf{Y}_{gt} + (1-t)\mathbf{Y}_{rec}$ with $0 \leq t \leq 1$. Hence, the loss of the discriminator is:

$$\mathcal{L}_D = \mathbb{E}[1 - D(\mathbf{Y}_{gt})] + \mathbb{E}[1 + D(\mathbf{Y}_{rec})] + w_{gp}\mathcal{L}_{gp}, \tag{6.3}$$

where $\mathbf{Y}$ refers to the concatenation of an animation and its corresponding intervertices distances.

For all our experiments, we set $w_{feat} = 1$, $\alpha_{gt} = 10$, $w_{gp} = 10$ and $w_{dis} = 1$. We set the initial learning rate of both the generator and the discriminator at 0.001. We use the Adam optimizer [KB14]. We add a dropout of 0.3 to regulate the generator. This system has been implemented using the Pytorch framework.

6.3 Results

In this section, we demonstrate our system's capability to render realistic animation with different types of editing constraints. First, we detail the data used for the training and the testing of our framework (Section 6.3.1). Then, we describe the different scenarios in which our framework might be useful, from unsupervised motion completion (Section 6.3.2), to constraint-based motion editing (Section 6.3.3).

6.3.1 Gathered Test Set

We use the two datasets presented in Chapter 3 for our experiments. We leverage the multimodal property of the B3D(AC)^2 to train our networks, especially the one requiring both facial animation and phoneme labels, and use the performance-based database alone (presented in Section 3.2) to train our "noisy-signal-based" editing system. As the noisy animations are obtained with an automatic face tracking solution [Dyn19a], noisy by nature, we do not need to add artificial noise to the input. For all our experiments, we resample every animation at 25 fps (the framerate of the B3D(AC)^2) and use the same blendshape model,the template $\mathcal{T}_l$ counting 34 blendshapes (see Figure 3.7) for every animation of each of our scenarios.

As with any learning-based methods, it is essential to know how the proposed approach depends on the training data. To test our framework, we record new sequences with a different subject, reciting new sentences, and performing different expressions to check if the model generalizes well. We derive both the original animations and the noisy ones using the same procedure as the one employed to derive the performance-based database (see Figure 3.1).

6.3.2 Unsupervised Motion Filling

First, we demonstrate the capability of our approach to generate plausible animation without any supervision. We validate it using animation of the test set by randomly removing some parts of them. We regenerate a complete sequence using our network, producing undirected motion filling. As we can see in the accompanying video[1], the generated parts (lasting 2.6s) are blended realistically with the animation preceding and following the edit. In this sequence, our generator produces "talking-style"

[1]Results presented in this chapter are shown in the supplementary video available at `https://elo-nsrb.github.io/homepage/publi_data/SCA2020/video.mp4`

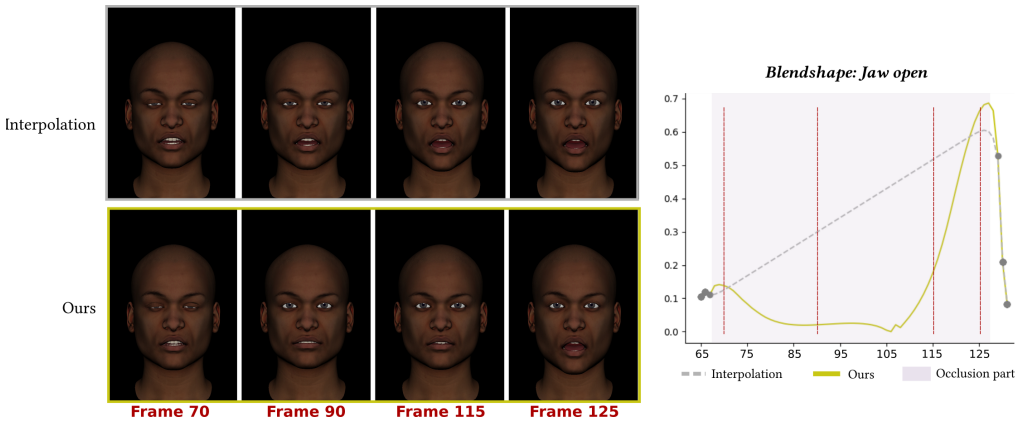


Figure 6.3 Occlusion motion completion. Compared to standard linear interpolation solving, our system generates realistic motion dynamics: in case of long occlusions, our system ensures that the mouth returns to the neutral pose. Moreover, as we use a bidirectional architecture, our system anticipates the wide opening of the mouth and smoothly re-open the mouth from the neutral pose.

motions and hallucinates eyebrows movements rendering the edited parts more plausible.

One potential application of our unsupervised animation generation system is its capability to generate more realistic sequences in case of long occlusions than simple interpolation methods. We use a new recorded sequence with occlusion of around 3 seconds (about 75 frames). Such occlusions often alter the quality of the final animation and require manual cleaning. We compare our generative method with a sequence resulting in interpolating the missing animation with boundaries and derivatives constraints. As we can see in Figure 6.3, filling the gap with interpolation leads to long oversmoothed motions, far from realistic motion patterns. Our system creates a more realistic sequence: the subject first returns to the neutral pose and anticipates the wide mouth opening by smoothly reopening the mouth. One might also observe the eyebrows activation, consistent with the mouth openings.

6.3.3 Guided Motion Editing

While unsupervised motion completion can be used to handle long occlusions, most relevant uses require users to steer the editing process. In the following, we present several use cases of guided facial animation editing. We test our system using both the test set, which is composed of sequences of unseen subjects, and new performance-based animations recorded outside the dataset.

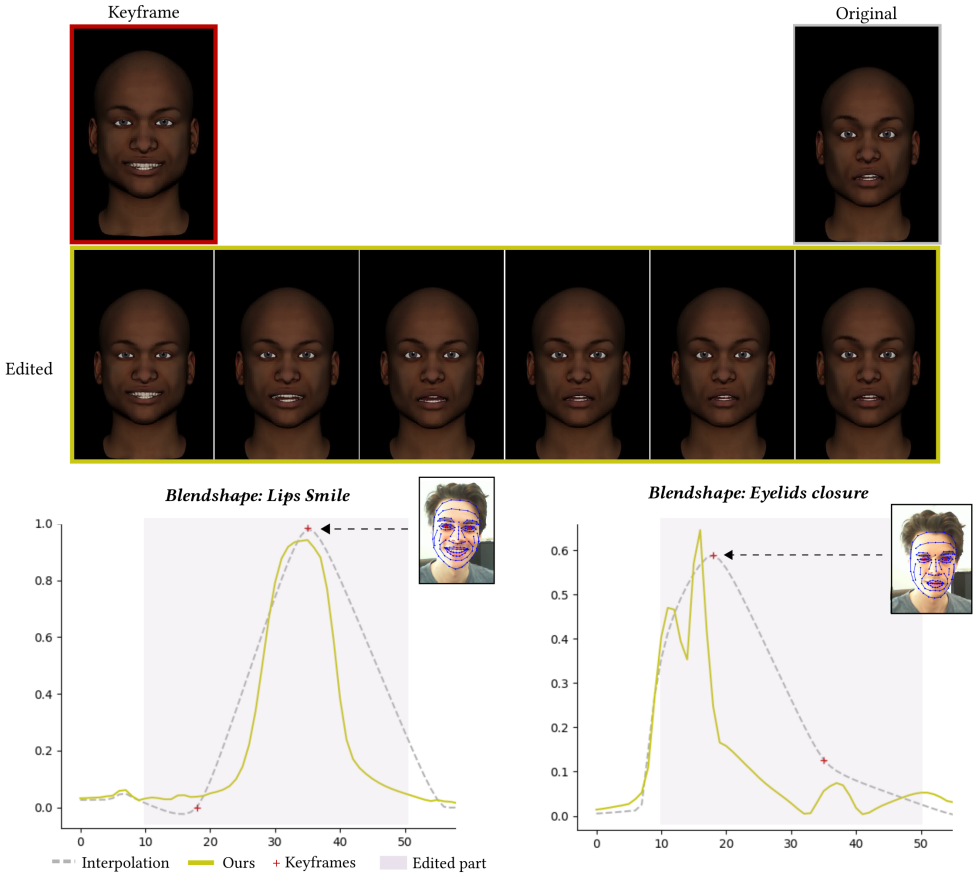


Figure 6.4 Validation of our keyframes-based constraint system on our testset with animation. Our system ensures natural coarticulation between key frames constraints and input signal.

6.3.3.1 Guiding with Keyframes

It is common for performance-based animation to require additional or localized corrections either due to technical or artistic considerations. Ideally, one would simply use new captured or manually-specified expressions to edit the animation and expect the editing tool to derive the right facial dynamics, reconstructing a realistic animation automatically. This use case has motivated the keyframe-based supervision of our editing system. We test our system's ability to handle this scenario by randomly removing parts of the input animation and inputting the network with sparse, closely- or widely-spaced, keyframe expressions. We observe that the system outputs natural and well-coarticulated motions between the keyframe constraints and the input signal: as we can see in Figure 6.4, our system generates non-linear blending around the smile keyframe expressions, and naturally reopens the mouth at the end of the edited segment. We can see in the video that our system generates a

more natural and organic facial dynamics than classic interpolation.

(a) Addition of one expression such as a wink. Our system naturally adds a key-expression: as we can observe, the mouth motion consistently moves to re-match the smiling expression.

(b) Modification of the mouth shape. Our system generates a more faithful shape of the mouth, given only one keyframe.

Figure 6.5 Keyframe-based Editing. Our system generates realistic motions with only a few keyframes as a constraint.

Another use case is adding an expression not present in the existing animation. For instance, in one of our videos, the performer forgot the final wink move at the end of the sequence (see 6.5a). We simply add it to the sequence by constraining the end of the sequence with a wink keyframe, which has been recorded later. We can observe in Figure 6.5a how naturally the mouth moves to combine the pre-existing smiling expression and the added wink request.

Finally, one recurrent shortcoming of performance-based animation is getting a mouth shape that does not match the audio speech. For instance, on a video outside the dataset, we observe that the face capture yields imprecise animation frames of

the mouth. As we can see in Figure 6.5b, the mouth should be almost closed, yet it remains wide open during a few frames. We fed the desired expression as a keyframe input to the system, and let the system generate the corrected mouth motion. The visual signature of labial consonants is a mouth closure. In the same editing spirit, our system can revise an inaccurate labial viseme by imposing mouth closure. We display an example of this correction in the accompanying video.

6.3.3.2 Guiding with Noisy Animation

Animation changes longer than a few seconds would require specifying many guiding keyframes. Instead, when long segments need to be substantially changed, one could guide animation editing with lower-quality facial tracking applications, using webcam or mobile phone feeds. In that case, the guiding animation is noisy and inaccurate, but is a simple and intuitive way to convey the animation intent. We test this configuration, feeding our system with noisy animations generated from a blendshape-based face tracking software as a guide for the animation segment to edit. As we can see in Figure 6.6, our system removes jitters and unrealistic temporal patterns but preserves natural high-frequency components such as the eyelids closures.

We compare the proposed approach to the one developed in Chapter 4. To do so, we mask-out the complete input (original) sequences included in the test set of the performance-based animation dataset (Section 3.2). Simultaneously, we feed the generator with the complete corresponding noisy animations and compute the MSE between the generated animation and the input one. We also provide the same sequences, which are been down-sampled at 25 fps, to the regression-based filtering system detailled earlier(see Section 4.2). The regression-based approach yields a lower MSE of 0.012 than the proposed GAN-based approach (MSE = 0.013). Qualitatively, we also observe that the output generated by the GAN-based system is less precise and includes motion patterns that are not present in the original sequence. On the contrary, animations produced by our previous filtering approach are more similar to the original ones. These results align with the general observations on the difference between traditional learning strategy and GAN framework.

We notice that the data used to train the GAN has been downsampled to a fixed and low framerate (25 fps), degrading the quality of animation samples. Hence, unlike the filtering system developed in Chapter 4 that is framerate independent, this network has not seen high resolution motion dynamics during training and therefore cannot retrieve it at run time. This might explain the loss of accuracy observed on animation produced by the GAN.

An interesting point to note is that the bidirectional recurrent architecture prevents from the drifting effect observed while processing long sequence (see Section 4.4.2).

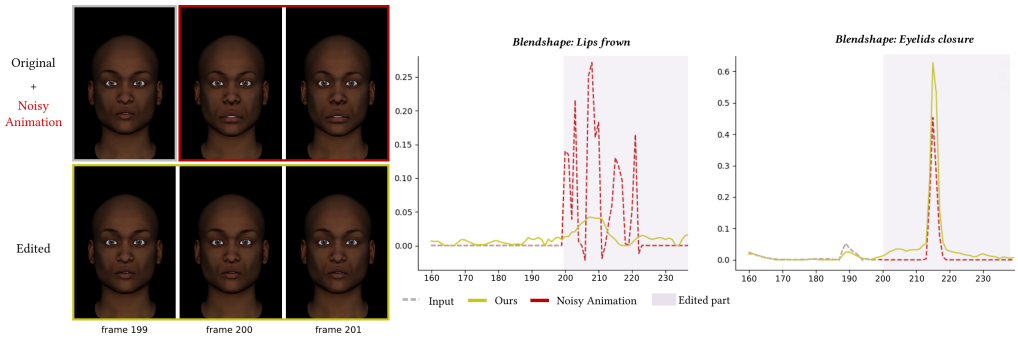


Figure 6.6 Noisy animation-based system. We mask half the original sequence and feed the network with the other half noisy animation. As we can see on the left, our system removes jitters and unnatural temporal patterns, generating a smooth animation at the boundary. We can see on the right, how the unrealistic lips frowning movements are filtered by our system, while the natural dynamic of the eyelids is preserved.

6.3.3.3 Guiding with Visemes

We demonstrate the capability of our system to edit an animation semantically. We use the initial sentence found in the test set *"Oh, I've missed you. I've been going completely doolally up here."*. We generate a new animation by substituting *"you"* with other nouns or noun phrases pronounced by the same subject in order to have consistent audio along with the animation. As we can see in Figure 6.7, our system generates new motions consistent with the input constraints, "our little brother": it adjusts the movements of the jaw to create a realistic bilabial viseme. We observe the closure of the mouth when pronouncing *"brother"* in Figure 6.7. It hallucinates consistent micro-motions, such as raising eyebrows at the same time, favoring natural-looking facial animation. Other examples are shown in the supplementary video.

We also perform viseme-based editing on a new subject reciting new sentences. For instance, we turn the initial sentence *"My favorite sport is skiing. I'm vacationing in Hawai this winter."* into *"My favorite sport is surfing. I'm vacationing in Hawai this winter."* The generated motion follows the new visemes sequence *"surfing"* in Figure 6.8. More precisely, we can see the bottom lip raising up to the bottom of the top teeth to generate the viseme *"f"*.

6.4 Evaluation

In this section, we present quantitative evaluations of our framework. First, we demonstrate the capability of our approach to reduce the manual effort required to edit facial animation. We then compare our methods with related ones in dealing with controllable animation editing. Finally, we assess the quality of our results by gathering user evaluation on a batch of edited sequences.

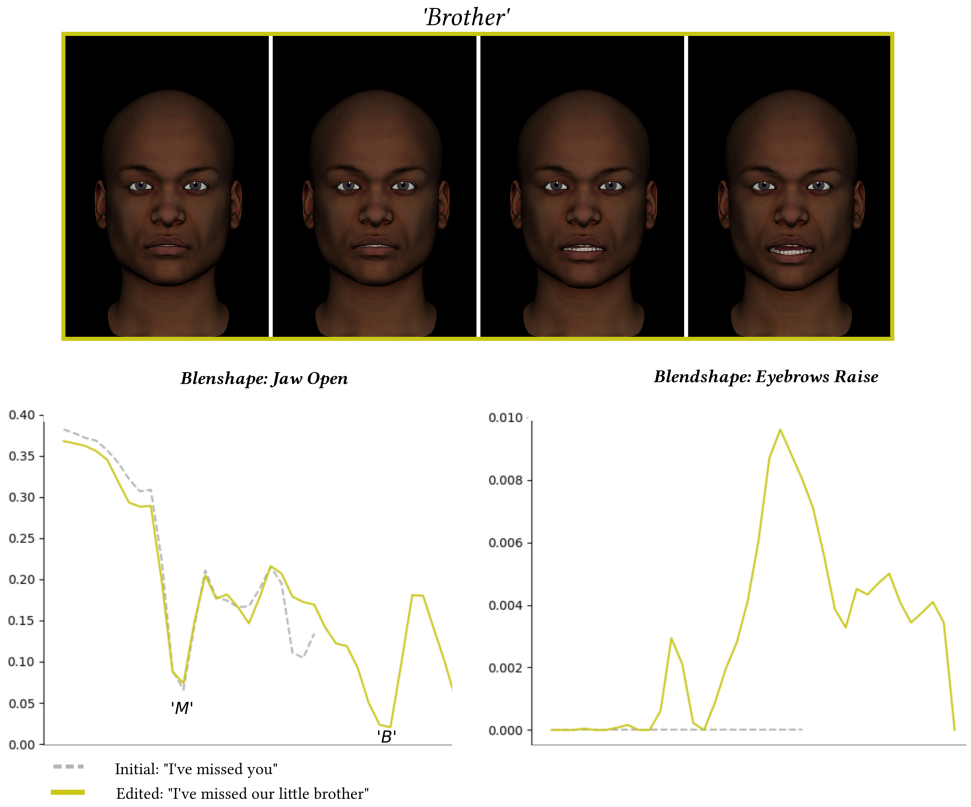


Figure 6.7 Our system modifies the jaw motion according to the input constraints such as adjusting the jaw opening to fit bilabial consonant constraints. It hallucinates micro-motions such as raising eyebrows to make the editing part more plausible. (Top) Generated frames given the input phonemes sequence *"brother"*.

6.4.1 Fast Animation Editing System

The principal objective of this work is to provide a system that accelerates the editing task. We timed two professional animators to measure the average time they need to create a sequence of 100 frames (see Table 6.2). From this experiment, we find that it takes between 20 and 50 minutes to create a 100-frames animation, depending on the complexity and the framerate of the animation. This amounts to an average individual keyframe setup time between 12 and 30 seconds. We note that this estimation is consistent with the study conducted by Seol et al. [SSK+11]. Now, for different sequences processed by our method, we estimate the number of keyframes that would be required to produce the same result manually. We proceed as follows: we automatically estimate the number of control points necessary for a cubic Bézier curve fitting algorithm to approximate the edited animation curves, within a tolerance threshold (set at 0.01). This process is repeated for each animation parameter independently. In manual facial animation, some complex motions require very dense keyframe layouts to look realistic, making our method all the more

Figure 6.8 Generated frames given the input phonemes sequence *"surfing"*. We can notice the bottom lip raises up to the bottom of the top teeth to generate the viseme *"f"*.

Table 6.2 Average time to create 100-frames animation.

	Handmade
Artist 1	$\sim$ 20 min
Artist 2	$\sim$ 50 min

appealing.

We compare in Table 6.3 the time to edit a few animations with our system and manual keyframing. From this experiment, we note that our system considerably reduces the time required to edit animation segments.

6.4.2 Comparison with Continuous Control Parameters Editing Systems

Recent controllable motion generation studies have an objective akin to animation editing, as they use regression neural networks to generate motion from high-level inputs. We compare our system to two previous works, closely related to motion editing: the seminal work of Holden et al. [HSK16] on controlled body motion generation and our previous work on facial animation editing (cf Chapter 5). For a fair comparison, we use the same control parameters as described in Chapter 5, but derived on the template $\mathcal{T}_l$ and train only one network for the different parts of

Table 6.3 Time performance evaluation. We compare the time to edit few animation with our system and manual keyframing. Our system considerably reduces the time of facial animation editing.

	# of frames	# of estimated Bézier Points	Average error by parameters	Manually created by an animator	Inference full sequence (CPU)
Occlusion completion	62	36	0.010	$\sim$ 12 min	0.14s
Viseme editing	19	15	0.012	$\sim$ 5 min	0.12 s
Noisy-based	116	93	0.010	$\sim$ 31 min	0.12 s

the face. We regress the corresponding blendshape weights using either the fully convolutional regressor and decoder of [HSK16], or the 2-network system proposed in Chapter 5. We quantitatively compare the reconstruction error between these methods and our system on the test set. Therefore, we mask-out the complete input animation and feed our network with the control parameter signals. We measure the mean square error between the original animation and the output one. As we can see in Table 6.4, our system achieves better performances than a regression network trained with MSE only.

Table 6.4 MSE between high level parameters and our network with 8 control parameters.

	MSE
[HSK16]	0.016
Chapter 5	0.018
Ours	**0.014**

We also observe qualitative differences between regressors [HSK16, BSBS19] and our current approach. We do so by feeding our generator with dense control parameter curves, as used by regressors. Even when stretching and deforming control curves to match sparse constraints, our system robustly continues to generate animation with realistic dynamics, preserving high-frequency motions as opposed to [HSK16].

As mentioned by Holden et al. [HSK16], the main issue with regression frameworks is the ambiguities of high-levels parameter inputs: the same set of high-level parameters can correspond to many different valid motion configurations. We test the behavior of our approach in such ambiguous cases, by using very few input control parameters (3): the mouth opening amplitude, the mouth's corners distance, and one eyelids closure distance. We indeed observe that a more ambiguous input signal leads to a noisier output animation for regression networks. With the same input, our system is able to hallucinate missing motion cues outputs, producing a more natural and smooth animation. We note that our system is even capable of creating plausible dynamics for the whole face in an unsupervised fashion (Section 6.3.2).

6.4.3 User Feedback

One widely recognized issue with animation generation methods is reliable evaluation of animation quality. Indeed, there is no quantitative metrics that reflect the naturalness and the realism of facial motions. Hence, we gather qualitative feedback on edited animation generated by our system in an informal study. A sample of 44 animation sequences -with different lengths and with or without audio- were presented to 21 subjects. Half the animations were edited with our system, using either visemes constraints, keyframes expression, noisy signals, or in an unsupervised fashion. Subjects were asked to assess whether the animation cames from original mocap or

was edited. In essence, participants were asked to play the role of the discriminator in distinguishing original from edited sequences. Most of the participants were not accustomed to close observation of 3D animation content. We gather the following user feedback among the 21 subjects: 54% of the original animations were classified as such (true positive), while 51% of edited sequences were also classified as original ones (false positive). We also show the sequences to 5 experienced subjects, that know the context of this work: even they detected only 58% of the edited sequences (true negative) and half of the original ones (true positive).

6.5 Conclusion

We have proposed a generative facial animation framework able to handle a wide range of animation editing scenarios. It enables unsupervised motion completion, semantic animation modifications, as well as animation editing from sparse keyframes or coarse noisy animation signals. The lack of high-quality animation data remains the major limitation in facial animation synthesis and editing research. While our system obtains good results, we note that the quality of produced animation can only be as good-looking and accurate as what the quality and diversity of our animation database covers. We present various results, testifying for the validity of the proposed framework, but the current state of our result calls for experimentation on more sophisticated blendshape models, more diverse facial motions, and possibly the addition of rigid head motion.

The presented method relies on a generative model and offers no guarantee as such to match input constraints exactly. Yet, ensuring an exact hit is a standard requirement for high-quality production. We note that a workaround solution in our case would be to post-process our system's animation to match sparse constraints exactly, following the interpolation of [SLS+12] for instance.

Beyond the proposed solution for offline facial animation editing, an interesting direction would be to enable facial animation modifications to occur in real-time. We plan on evaluating the performance of a forward-only recurrent network to assess the feasibility of real-time use cases. We leverage recurrence architecture that with its internal memory is conducive to approximating and extrapolating missing frames. A useful application of our work could be temporal super-resolution animation restoring that is decompressing animation with a low framerate by interpolating transitions between the given frames.

Our system aims to make facial animation editing more accessible to non-expert users, but also more time-efficient, to reduce the bottleneck of animation cleaning and editing. In terms of user interaction, our semantic editing framework requires isolating the animation segments to edit, and providing editing cues. An interesting future work would be to integrate our system within a user-oriented application, combining our network with a user interface and a recording framework, forming a complete, interactive, efficient animation editing tool.

Our experiments show that the proposed framework achieves good performance

on facial animation editing, demonstrating the potential and the versatility of our approach to animation modeling. This work opens up new possibilities in the future, in particular, it would be interesting to extent this work by considering other inputs including alternative controllers, audio signal or image-based features.

CHAPTER 7

Conclusion

Contents

7.1 Summary of Contributions

This thesis falls within the context of Motion Capture facial animation synthesis, a technology with the theoretical promise of having the ability to completely and flawlessly capture and retarget a human performance, from emotion down the most subtle motion of facial skin.

While Motion Capture has revolutionized the 3D animation field by providing a way to generate highly realistic animation, some pitfalls remain, limiting the access and the usability of this technique. Research and industry efforts have largely focused on improving the precision and robustness of motion capture. While this has undoubtedly lead performance-based animation to great results and success, It's very common that the animation obtained from the performance does not meet the quality requirement or the artistic intent. An animation editing step is thus necessary to correct or change the animation, implying going back to lengthy and tedious manual keyframing tasks.

Throughout this thesis, building upon previous work, we investigate recent machine learning techniques that have had striking success in other fields to extend various aspects of the facial animation editing. With the intuition that artificial intelligence is the next step in the 3D animation content production and could overcome these issues in the future, we report our investigation, results and limitations in this document with the hope to contribute to the improvement of 3D animation content editing.

In Chapter 4, we develop a real-time algorithm to automatically clean and refine an animation, leveraging from the recent progress made in recurrent neural architecture. Compared to standard low-level editing methods, the proposed system is causal and faithfully restores the natural dynamic of facial animation, without losing crucial cues included in the high-frequency motions or requiring painstaking manual fine-tuning. We suggest enhancing the standard recurrent structure, by feeding back the previous estimate state, to prevent our system from dying out or drifting away when processing long sequences. The proposed parameterization enables producing qualitative, pleasant, accurate results, outperforming standard signal processing techniques, and non-recurrent machine learning approaches. Once the training is done, our system, based on a causal architecture, is fast enough for a real-time application.

Cleaning and low-level editing is only one part of the animation editing concerns. More high-level control is needed. Thus, we investigate new algorithms to improve the constraint-based facial motion editing process.Presented in Chapter 5, we develop a solution for fast and efficient animation editing, that does not need any manual data pre-processing of the animation sequence. Following-up on recent advances in body-motion editing work, the proposed technique preserves and faithfully regenerates the specificity of the facial movements, delivering realistic modified animation. We propose a new facial animation representation to steer facial movements through temporally dense low-dimensional control parameters. Using an original interface, we demonstrate the suitability of our solution for a concrete animation editing application. This work represents the first step toward intelligent editing systems, which contributes to improving the efficiency of the facial animation manipulation. Besides, applying a machine-learning technique trained on real data ensures realistic animation production, while preventing the users from precisely defining all the temporal behaviors and every detail of the facial motion, making the editing task easier and more accessible. Yet, several limitations remains due to the regression-based approach including the nature of the input and the training strategy.

With the limitations of Chapter 5 in mind, we further explore intelligent facial editing in Chapter 6 and alternative facial motion representation to bypass these issues. We explore the GAN paradigm that has yielded very encouraging visual results, to provide a versatile solution allowing unsupervised, slightly supervised, or fully controllable motion editing. Notably, the proposed solution leverage adversarial learning to validate the quality of the motion by the model itself, and allow stochastic motion generation. This enables filling motion with plausible motion patterns, outperforming standard motion interpolation models. Notable attractive points include the flexibility and efficiency of the proposed technique. Suitable for the art direction of the standard keyframing workflow, the developed framework enables automatically dealing with motion occlusions or perform semantic animation

modifications, accelerating and supporting the facial editing process.

Recent learning-based algorithms, such as neural networks, have the ability to capture complex and diversified data distribution, including motion dynamics as some previous works have empirically proved. This makes them particularly well-suited tools to build systems that robustly generate realistic non-linear signals, even when driven by noisy, sparse or low-quality inputs. We demonstrate throughout this thesis their benefit to efficient and convenient dynamic facial animation editing solutions. By developing deep learning techniques devoted to the performance-based data manipulation, we hope to have contributed to the long-term objective of making efficient and manageable high-quality animation production tools more accessible and widespread. This thesis has led to publications in international conferences, which are listed below:

- E. Berson, C. Soladié, V. Barrielle, and N. Stoiber. A Robust Interactive Facial Animation Editing System. In *Proceedings of the 12th Annual International Conference on Motion, Interaction, and Games - MIG '19*, Newcastle-upon-Tyne, United Kingdom, 2019
- E. Berson, C. Soladié, and N. Stoiber. Real-Time Cleaning and Refinement of Facial Animation Signals. In *Proceedings of the 4th International Conference on Graphics and Signal Processing - ICGSP 2020*, Nagoya, Japan, 2020
- E. Berson, C. Soladié, and N. Stoiber. Intuitive facial animation editing based on a generative rnn framework. In *Proceedings of the 19th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '20*, Montreal, 2020

7.2 Perspectives and Future Work

The research progress outlined in this thesis is a first step in bringing forward the facial animation editing process. However, a number of problems remain, suggesting possible future research directions.

Creating a high-quality facial animation rests on various interconnected components, including motion-capture techniques settling the original animation's quality, a rigging step establishing the quality of animation parameterization and an editing step, whereby artists refine the animation. This thesis explores new perspectives improving the facial motion editing component. However, focusing on motion editing rests on the quality of the performance-based database, and assumes a given animation model.

Indeed, the main significant issue, shared among the deep learning community, remains the dependency on an appropriate large database of motion examples used for training. As observed in many studies, neural-based systems might fail if the input sample goes far beyond the span of the training set. While we provide an initial

validation of our solutions in this thesis, future works are necessary to test our solution to more complex facial animations. We start to overcome this issue in Chapter 3 by enhancing an available database [FGR+10]. Providing animated faces that climb up the right side of the uncanny valley is still challenging as humans are too good at noticing very small errors. Awe-inspiring results have been achieved in the facial animation field using neural-based approaches and high-quality data [LSSS18, WSS+19]. However, these data remain very rare as it implies using an expensive setup (up to 40 multicamera) for the creation. A further direction would be to use the last state-of-the-art 3D model reconstruction to extent this database. An interesting data augmentation that would ensure a larger diversity among motion examples would be to leverage the amount of available 2D videos. For instance, we could use the recent outstanding 3D differential neural-based rendering approaches [TZK+17], to automatically extract the 3D face mesh representation from a 2D images using the combination of a 3D differential rendered and a neural network. Combined with state-of-the-art speech detection/alignment systems, it would improve the talking motion coverage of our database.

Regarding the rigging foundation, we consider in this work the blendshape parametrization as it is a de-facto standard in both production and research communities. Yet, this model already limits the precision of the allowed temporal deformation [BSC16]. Interesting path could be explored to extend our system to other animation formats. One promising path would combine our work, with an interactive 3D face sculpting enhancement tool [GFZ+20] and/or to augment the proposed solution with automatic solutions restoring facial details, by leveraging the texture [LSSS18] or using the fast deep deformation system proposed by Bailey et al. [BODO20] mapping rig parameters to high-quality mesh.

In this thesis, we explore the facial motion controllability. The temporal curves proposed in Chapter 5 make explicit the temporal aspect of the motion, suggesting an overall user's authoring on the motion dynamics. Based on geometrical and semantic considerations, these parameters allow the user to coarsely edit the most salient features of the facial motion on a large scale. While leading to efficient motion editing, they fail to embody the whole range of spatiotemporal motion patterns, which constitute the intricate facial dynamic signature. Alternatively, in Chapter 6 we step toward more higher-level motion editing requiring less involvement of the users in supervising every instant in the timeline, while emphasizing the versatility and compactness of our solution with regard to actual editing application scenarios in real production pipeline. Yet, one could imagine new abstractions of the facial motions, encompassing both the geometrical and the temporal dimensions of this process. Guay and colleagues [GRGC15] propose a new paradigm, named *line of actions*, to drive body-motion through strokes in space-time fashion simultaneously, opening new motion metaphor perspectives [COS19]. While such representation is not directly transposable in the facial area, it is an interesting topic for future research.

With the booming success of (inverse) neural rendering [TZK+17, TFT+20], another direction would be to align 3D motion editing with 2D video editing, reforming the

spatiotemporal facial animation representation, and the user controllers. Combining 3D differential renders with neural networks, this technique has yielded tremendous results, outperforming standard computer graphics methods. Nevertheless, achieving fully motion controllability through neural approaches raises questions about the editability of the output animation. While neural networks have demonstrated great success in producing highly realistic animation, the prediction and control of failure cases are still open questions. Indeed, the main limitation of neural-based solutions remains its *black box* property. The growing prominence of neural networks does not imply a real understanding of their functioning. There is still no precise recipe for dealing with spatiotemporal data such as facial motion. Overall, the design of their architecture remains an empirical process. However, we notice that the technical choices and the scientific direction we have made, fall with the latest Computer Animation research [HYNP20, ZWR+20, ZLH+20, ZLB+20].

Although Motion Capture technology has become readily available for everyday consumers through affordable 3D capture technology, high-quality animations still require costly setups [LSSS18]. Bringing the gap between the accessibility and quality of 3D animation content is already in development, and we believe it will be unveiled in the near future. Going a step further, one could imagine the future of the facial animation in generative methods rather than Motion Capture technologies. As far as facial animation is concerned, we hope that this work will be the basis for further follow-up research, promoting its spread in a variety of new applications, reaching a larger audience.

Acronyms

AAM	Active Appearance Models
B3D(AC)^2	3D Audio-Visual Corpus of Affective Communication
CNN	Convolutional Neural Network
CRBM	Conditional Restricted Bolzmann Machine
DTW	Dynamic Time Warping
GAN	Generative Adversarial Network
GPLVM	Gaussian Process Latent Variable Model
HMM	Hidden Markov Models
ICA	Independent Component Analysis
IK	Inverse Kinematic
LDS	Linear Dynamical System
LLE	Locally Linear Embedding
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MoCap	Motion Capture
MSE	Mean Square Error
NN	Neural Network
PCA	Principal Component Analysis
RBF	Radial Basis Functions
RNN	Recurrent Neural Network
SLDS	Switching Linear Dynamical System

List of Figures

List of Tables

Bibliography

[ACOH+19] A. Aristidou, D. Cohen-Or, J. K. Hodgins, Y. Chrysanthou, and A. Shamir. Deep motifs and motion signatures. *ACM Transactions on Graphics*, 37(6):1–13, Jan. 2019. doi: 10.1145/3272127.3275038.

[ACP03] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)*, 22(3):587–594, 2003.

[AECOKC17] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics*, 36(6):1–13, Nov. 2017. doi: 10.1145/3130800.3130818.

[AF02] O. Arikan and D. A. Forsyth. Interactive Motion Generation from Examples. *ACM Transactions on Graphics (TOG)*, 21(3):483–490, 2002.

[AFO03] O. Arikan, D. A. Forsyth, and J. F. O'Brien. Motion synthesis from annotations. In *ACM SIGGRAPH 2003 Papers*, pages 402–408, 2003.

[AKA96] K. Arai, T. Kurihara, and K.-i. Anjyo. Bilinear interpolation for facial expression and metamorphosis in real-time animation. *The Visual Computer*, 12(3):105–116, 1996.

[ALCS18] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. Inverse Kinematics Techniques in Computer Graphics: A Survey: Inverse Kinematics Techniques in Computer Graphics. *Computer Graphics Forum*, 37(6):35–58, Sept. 2018. doi: 10.1111/cgf.13310.

[AM00] M. Alexa and W. Müller. Representing Animations by Principal Components. *Computer Graphics Forum*, 19(3):411–418, Sept. 2000. doi: 10.1111/1467-8659.00433.

[APM00] n. Allison, n. Puce, and n. McCarthy. Social perception from visual cues: role of the STS region. *Trends in Cognitive Sciences*, 4(7):267–278, July 2000. doi: 10.1016/s1364-6613(00)01501-1.

[ARV07] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, June 2007. IEEE. doi: 10.1109/CVPR.2007.383165.

[ASK+12] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):1–12, Apr. 2012. doi: 10.1145/2159516.2159523.

[ASWC13] R. Anderson, B. Stenger, V. Wan, and R. Cipolla. Expressive visual text-to-speech using active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3382–3389, 2013.

[ATL12] K. Anjyo, H. Todo, and J. Lewis. A Practical Approach to Direct Manipulation Blendshapes. *Journal of Graphics Tools*, 16(3):160–176, Aug. 2012. doi: 10.1080/2165347X.2012.689747.

[BBA+07] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics*, 26(99):33, July 2007. doi: 10.1145/1239451.1239484.

[BBEO03] G. Bailly, M. Berar, F. Elisei, and M. Odisio. Audiovisual speech synthesis. *International Journal of Speech Technology*, 6(4):331–346, 2003.

[BBG+14] A. H. Bermano, B. Bickel, M. Gross, D. Bradley, T. Beeler, F. Zund, D. Nowrouzezahrai, I. Baran, O. Sorkine-Hornung, H. Pfister, and R. W. Sumner. Facial performance enhancement using dynamic shape space analysis. *ACM Transactions on Graphics*, 33(2):1–12, Apr. 2014. doi: 10.1145/2546276.

[BBKK17] J. Bütepage, M. Black, D. Kragic, and H. Kjellström. Deep representation learning for human motion prediction and classification. *arXiv:1702.07486 [cs]*, Apr. 2017. URL `http://arxiv.org/abs/1702.07486`. arXiv: 1702.07486.

[BBPV03] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Computer graphics forum*, volume 22, pages 641–650. Wiley Online Library, 2003.

[BCHF19] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes. DReCon: data-driven responsive control of physics-based characters. *ACM Transactions on Graphics*, 38(6):1–11, Nov. 2019. doi: 10.1145/3355089.3356536.

[BCS97] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360. ACM Press/Addison-Wesley Publishing Co., 1997.

[BH00] M. Brand and A. Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, 2000.

[Bis05] A. Bissacco. Modeling and Learning Contact Dynamics in Human Motion. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 421–428, San Diego, CA, USA, 2005. IEEE. doi: 10.1109/CVPR.2005.225.

[BKL17] E. Barsoum, J. Kender, and Z. Liu. HP-GAN: Probabilistic 3D human motion prediction via GAN. *arXiv preprint arXiv:1711.09561*, 2017.

[BKMVG07] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106(1):116–129, 2007.

[Bla94] P. Blair. *Cartoon Animation*. How to Draw and Paint. Walter Foster Publishing, 1994.

[BLB+08] B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 57–66. Eurographics Association, 2008.

[BM92] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[BODO20] S. W. Bailey, D. Omens, P. Dilorenzo, and J. F. O'Brien. Fast and Deep Facial Deformations. *ACM Transactions on Graphics*, 39(4):94:1–15, Aug. 2020. doi: 10.1145/3386569.3392397.

[Bow00] R. Bowden. Learning Statistical Models of Human Motion. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, volume 2000, page 8, 2000.

[BP14] S. Bouaziz and M. Pauly. Semi-Supervised Facial Animation Retargeting. Technical report, 2014.

[Bra99] M. Brand. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.

[Bre97] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 568–574, San Juan, Puerto Rico, 1997. IEEE Comput. Soc. doi: 10.1109/CVPR.1997.609382.

[BS08] M. Botsch and O. Sorkine. On Linear Variational Surface Deformation Methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, Jan. 2008. doi: 10.1109/TVCG.2007.1054.

[BS09] A. Bissacco and S. Soatto. Hybrid Dynamical Models of Human Motion for the Recognition of Human Gaits. *International Journal of Computer Vision*, 85(1):101–114, Oct. 2009. doi: 10.1007/s11263-009-0248-7.

[BSBS19] E. Berson, C. Soladié, V. Barrielle, and N. Stoiber. A Robust Interactive Facial Animation Editing System. In *Proceedings of the 12th Annual International Conference on Motion, Interaction, and Games - MIG '19*, Newcastle-upon-Tyne, United Kingdom, 2019.

[BSC16] V. Barrielle, N. Stoiber, and C. Cagniart. BlendForces: A Dynamic Framework for Facial Animation. *Computer Graphics Forum*, 35(2):341–352, May 2016. doi: 10.1111/cgf.12836.

[BSS20a] E. Berson, C. Soladié, and N. Stoiber. Intuitive facial animation editing based on a generative rnn framework. In *Proceedings of the 19th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '20*, Montreal, 2020.

[BSS20b] E. Berson, C. Soladié, and N. Stoiber. Real-Time Cleaning and Refinement of Facial Animation Signals. In *Proceedings of the 4th International Conference on Graphics and Signal Processing - ICGSP 2020*, Nagoya, Japan, 2020.

[But15] M. Buttner. Motion Matching-The Road to Next-Gen Animation. In *Nucl. ai Conference*, 2015.

[BV99] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. doi: 10.1145/311535.311556.

[BVS16] M. Bessmeltsev, N. Vining, and A. Sheffer. Gesture3D: posing 3D characters via gesture drawings. *ACM Transactions on Graphics*, 35(6):1–13, Nov. 2016. doi: 10.1145/2980179.2980240.

[BW71] N. Burtnyk and M. Wein. Computer-generated key-frame animation. *Journal of the SMPTE*, 80(3):149–153, 1971.

[BW75] N. Burtnyk and M. Wein. Computer animation of free form images. In *Proceedings of the 2nd annual conference on Computer graphics and interactive techniques*, pages 78–80, 1975.

[BW95] A. Bruderlin and L. Williams. Motion signal processing. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, pages 97–104, Not Known, 1995. ACM Press. doi: 10.1145/218380.218421.

[CAW+19] H.-k. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles. Action-agnostic human pose forecasting. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1423–1432. IEEE, 2019.

[CBL+19] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black. Capture, Learning, and Synthesis of 3D Speaking Styles. *Computer Vision and Pattern Recognition (CVPR)*, page 11, 2019.

[CDB02] E. Chuang, F. Deshpande, and C. Bregler. Facial expression space learning. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 68–76, Beijing, China, 2002. IEEE Comput. Soc. doi: 10.1109/PCCGA.2002.1167840.

[CE05] Y.-J. Chang and T. Ezzat. Transferable videorealistic speech animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 143–151. ACM, 2005.

[CFCF16] L. Crnkovic-Friis and L. Crnkovic-Friis. Generative choreography using deep learning. *arXiv preprint arXiv:1605.06921*, 2016.

[CFKP04] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin. Real-time speech motion synthesis from recorded motions. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*, page 345, Grenoble, France, 2004. ACM Press. doi: 10.1145/1028523.1028570.

[CFP03] Y. Cao, P. Faloutsos, and F. Pighin. Unsupervised Learning for Speech Motion Editing. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 225–231, 2003.

[CG00] E. Cosatto and H. P. Graf. Photo-realistic talking-heads from image samples. *IEEE Transactions on multimedia*, 2(3):152–163, 2000.

[CGNS17] L. Ciccone, M. Guay, M. Nitti, and R. W. Sumner. Authoring motion cycles. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 1–9, Los Angeles California, July 2017. ACM. doi: 10.1145/3099564.3099570.

[CGZ17] J. Chi, S. Gao, and C. Zhang. Interactive facial expression editing based on spatio-temporal coherency. *The Visual Computer*, 33(6-8):981–991, 2017.

[CH05] J. Chai and J. K. Hodgins. Performance animation from low-dimensional control signals. *ACM SIGGRAPH 2005 Papers*, pages 686–696, 2005.

[CH07] J. Chai and J. K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. *ACM SIGGRAPH 2007 papers*, pages 8–es, 2007.

[CHZ14] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics*, 33(4):1–10, July 2014. doi: 10.1145/2601097.2601204.

[CiRL+16] B. Choi, R. B. i Ribera, J. P. Lewis, Y. Seol, S. Hong, H. Eom, S. Jung, and J. Noh. SketchiMo: sketch-based motion editing for articulated characters. *ACM Transactions on Graphics*, 35(4):1–12, July 2016. doi: 10.1145/2897824.2925970.

[CJ08] E. Chang and O. C. Jenkins. Sketching articulation and pose for facial animation. In *Data-Driven 3D Facial Animation*, pages 145–161. Springer, 2008.

[CKH11] D. Cosker, E. Krumhuber, and A. Hilton. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *2011 International Conference on Computer Vision*, pages 2296–2303, Barcelona, Spain, Nov. 2011. IEEE. doi: 10.1109/ICCV.2011.6126510.

[CLK01] B. Choe, H. Lee, and H.-S. Ko. Performance-Driven Muscle-Based Facial Animation. *The Journal of Visualization and Computer Animation*, 12:67–79, 2001.

[CLO17] O. Cetinaslan, J. Lewis, and V. Orvalho. Transposition Based Blendshape Direct Manipulation. In *VISIGRAPP (1: GRAPP)*, pages 105–115, 2017.

[CM92] Y. Chen and G. G. Medioni. Object modeling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[CM93] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. In *Models and techniques in computer animation*, pages 139–156. Springer, 1993.

[CO18] O. Cetinaslan and V. Orvalho. Direct manipulation of blendshapes using a sketch-based interface. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology - Web3D '18*, pages 1–10, Poznań, Poland, 2018. ACM Press. doi: 10.1145/3208806.3208811.

[COL15] O. Cetinaslan, V. Orvalho, and J. P. Lewis. Sketch-Based Controllers for Blendshape Facial Animation. *Eurographics (Short Papers)*, pages 25–28, 2015.

[COS19] L. Ciccone, C. Öztireli, and R. W. Sumner. Tangent-space optimization for interactive animation control. *ACM Transactions on Graphics*, 38(4):1–10, July 2019. doi: 10.1145/3306346.3322938.

[CPAMN20] E. Corona, A. Pumarola, G. Alenyà, and F. Moreno-Noguer. Context-aware Human Motion Prediction. *arXiv:1904.03419 [cs]*, Mar. 2020. URL `http://arxiv.org/abs/1904.03419`. arXiv: 1904.03419.

[CSLK19] Q. Cui, H. Sun, Y. Li, and Y. Kong. A Deep Bi-directional Attention Network for Human Motion Recovery. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 701–707, Macao, China, Aug. 2019. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2019/99.

[CTFP05] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin. Expressive speech-driven facial animation. *ACM Transactions on Graphics (TOG)*, 24(4):1283–1302, 2005.

[CUH15] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289 [cs]*, Nov. 2015. URL `http://arxiv.org/abs/1511.07289`. arXiv: 1511.07289.

[CVTV05] Y. Chang, M. Vieira, M. Turk, and L. Velho. Automatic 3D facial expression analysis in videos. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 293–307. Springer, 2005.

[CWB+11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[CWLZ13] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics*, 32(4):1, July 2013. doi: 10.1145/2461912.2462012.

[CWW+16] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics*, 35(4):1–12, July 2016. doi: 10.1145/2897824.2925873.

[CWZ+14] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[CXH03] J.-x. Chai, J. Xiao, and J. Hodgins. Vision-based Control of 3D Facial Animation. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 193–206, 2003.

[DBB+18] D. Dinev, T. Beeler, D. Bradley, M. Bächer, H. Xu, and L. Kavan. User-Guided Lip Correction for Facial Performance Capture. *Computer Graphics Forum*, 37(8):93–101, Dec. 2018. doi: 10.1111/cgf.13515.

[DBNN04] Z. Deng, M. Bulut, U. Neumann, and S. Narayanan. Automatic Dynamic Expression Synthesis For Speech Animation. In *IEEE Computer Animation and Social Agents*, pages 267–274, 2004.

[DBR20] K. Deng, A. Bansal, and D. Ramanan. Unsupervised Any-to-Many Audiovisual Synthesis via Exemplar Autoencoders. *arXiv:2001.04463 [cs, eess]*, Jan. 2020. URL `http://arxiv.org/abs/2001.04463`. arXiv: 2001.04463.

[DBS18] K. Dobs, I. Bülthoff, and J. Schultz. Use and Usefulness of Dynamic Face Stimuli for Face Perception Studies—a Review of Behavioral Findings and Methodology. *Frontiers in Psychology*, 9:1355, Aug. 2018. doi: 10.3389/fpsyg.2018.01355.

[DDFG01] A. Doucet, N. De Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[DHG13] S. Deena, S. Hou, and A. Galata. Visual speech synthesis using a variable-order switching shared Gaussian process dynamical model. *IEEE Transactions on Multimedia*, 15(8):1755–1768, 2013.

[DM08] Z. Deng and X. Ma. Perceptually guided expressive facial animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 67–76. Eurographics Association, 2008.

[DN06] Z. Deng and U. Neumann. eFASE: expressive facial animation synthesis and editing with phoneme-isomap controls. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 251–260. Eurographics Association, 2006.

[DN07] Z. Deng and J. Noh. Computer Facial Animation: A Survey. In Z. Deng and U. Neumann, editors, *Data-Driven 3D Facial Animation*, pages 1–28. Springer London, London, 2007. doi: 10.1007/978-1-84628-907-1_1.

[DRvdP15] B. Dalstein, R. Ronfard, and M. van de Panne. Vector graphics animation with time-varying topology. *ACM Transactions on Graphics*, 34(4):1–12, July 2015. doi: 10.1145/2766913.

[DSJ+11] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011.

[Dyn19a] Dynamixyz. *Live-Instant*. 2019. URL `http://www.dynamixyz.com/performer-single-view/`.

[Dyn19b] Dynamixyz. *Performer*. 2019. URL `http://www.dynamixyz.com/performer-single-view/`.

[ECPC18] M. Engilberge, L. Chevallier, P. Perez, and M. Cord. Finding Beans in Burgers: Deep Semantic-Visual Embedding with Localization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3984–3993, Salt Lake City, UT, June 2018. IEEE. doi: 10.1109/CVPR.2018.00419.

[ECR08] G. Englebienne, T. Cootes, and M. Rattray. A probabilistic model for generating realistic lip movements from speech. In *Advances in neural information processing systems*, pages 401–408, 2008.

[EGP02] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. *ACM Transactions on Graphics (TOG)*, 21(3):388–398, 2002. Publisher: ACM New York, NY, USA.

[Ekm97] R. Ekman. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[EL99] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1033–1038 vol.2, Kerkyra, Greece, 1999. IEEE. doi: 10.1109/ICCV.1999.790383.

[ELFS16] P. Edwards, C. Landreth, E. Fiume, and K. Singh. JALI: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG)*, 35(4):127, 2016.

[EST+19] B. Egger, W. A. P. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter. 3D Morphable Face Models – Past, Present and Future. *arXiv:1909.01815 [cs]*, Sept. 2019. URL http://arxiv.org/abs/1909.01815. arXiv: 1909.01815.

[FGR+10] G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool. A 3-D Audio-Visual Corpus of Affective Communication. *IEEE Transactions on Multimedia*, 12(6):591–598, Oct. 2010. doi: 10.1109/TMM.2010.2052239.

[FHKS12] A. Feng, Y. Huang, M. Kallmann, and A. Shapiro. An analysis of motion blending techniques. In *International Conference on Motion in Games*, pages 232–243. Springer, 2012.

[FJX+15] Y. Feng, M. Ji, J. Xiao, X. Yang, J. J. Zhang, Y. Zhuang, and X. Li. Mining Spatial-Temporal Patterns and Structural Sparsity for Human Motion Data Denoising. *IEEE Transactions on Cybernetics*, 45(12):2693–2706, Dec. 2015. doi: 10.1109/TCYB.2014.2381659.

[FKY08] W.-W. Feng, B.-U. Kim, and Y. Yu. Real-time data driven deformation using kernel canonical correlation analysis. *ACM Transactions on Graphics*, 27(3):1, Aug. 2008. doi: 10.1145/1360612.1360690.

[FLFM15] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[Flu11] B. Flueckiger. Computer-Generated Characters in Avatar and Benjamin Button. page 28, 2011.

[FRQL00] M. J. Farah, C. Rabinowitz, G. E. Quinn, and G. T. Liu. Early commitment of neural substrates for face recognition. *Cognitive Neuropsychology*, 17(1):117–123, Feb. 2000. doi: 10.1080/026432900380526.

[FTZ+19] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala. Text-based editing of talking-head video. *ACM Transactions on Graphics*, 38(4):1–14, July 2019. doi: 10.1145/3306346.3323028.

[GAA+17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. *arXiv:1704.00028 [cs, stat]*, Dec. 2017. URL http://arxiv.org/abs/1704.00028. arXiv: 1704.00028.

[Gar93] J. S. Garofolo. TIMIT acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[GBBB06] O. Govokhina, G. Bailly, G. Breton, and P. Bagshaw. TDA: A new trainable trajectory formation system for facial animation. *International Conference on Spoken Language Processing*, page 5, 2006.

[GCR13] M. Guay, M.-P. Cani, and R. Ronfard. The line of action: an intuitive interface for expressive character posing. *ACM Transactions on Graphics*, 32(6):1–8, Nov. 2013. doi: 10.1145/2508363.2508397.

[GF16] Ø. Gløersen and P. Federolf. Predicting Missing Marker Trajectories in Human Motion Data Using Marker Intercorrelations. *PLOS ONE*, 11(3):e0152616, Mar. 2016. doi: 10.1371/journal.pone.0152616.

[GFZ+20] A. Gruber, M. Fratarcangeli, G. Zoss, R. Cattaneo, T. Beeler, M. Gross, and D. Bradley. Interactive Sculpting of Digital Faces Using an Anatomical Modeling Paradigm. In *Computer Graphics Forum*, volume 39, page 2, 2020.

[GJ14] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

[Gle97] M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*, pages 139–ff., Providence, Rhode Island, United States, 1997. ACM Press. doi: 10.1145/253284.253321.

[Gle99] M. Gleicher. Animation from observation: Motion capture and motion editing. *ACM SIGGRAPH Computer Graphics*, 33(4):51–54, Nov. 1999. doi: 10.1145/345370.345409.

[Gle01] M. Gleicher. Comparing Constraint-Based Motion Editing Methods. *Graphical Models*, 63(2):107–134, Mar. 2001. doi: 10.1006/gmod.2001.0549.

[Gle08] M. L. Gleicher. Graph-based motion synthesis: an annotated bibliography. In *ACM SIGGRAPH 2008 classes on - SIGGRAPH '08*, page 1, Los Angeles, California, 2008. ACM Press. doi: 10.1145/1401132.1401200.

[GLSR19] F. Gaisbauer, J. Lehwald, J. Sprenger, and E. Rukzio. Natural Posture Blending Using Deep Neural Networks. In *Motion, Interaction and Games*, pages 1–6, Newcastle upon Tyne United Kingdom, Oct. 2019. ACM. doi: 10.1145/3359566.3360052.

[GM10] O. Gunnarsson and S. Maddock. Sketch-Based Posing of 3D Faces for Facial Animation. *Theory and Practice of Computer Graphics*, page 8 pages, 2010. doi: 10.2312/LOCALCHAPTEREVENTS/TPCG/TPCG10/223-230.

[GMHP04] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM SIGGRAPH 2004 Papers*, pages 522–531, 2004.

[GMK+19] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019.

[GP03] M. A. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4(3):179–192, Mar. 2003. doi: 10.1038/nrn1057.

[GPAM+14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. URL `http://arxiv.org/abs/1406.2661`. arXiv: 1406.2661.

[GRBS10] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and Filtering for Human Motion Capture: A Multi-Layer Framework. *International Journal of Computer Vision*, 87(1-2):75–92, Mar. 2010. doi: 10.1007/s11263-008-0173-1.

[GRGC15] M. Guay, R. Ronfard, M. Gleicher, and M.-P. Cani. Space-time sketching of character animation. *ACM Transactions on Graphics*, 34(4):1–10, July 2015. doi: 10.1145/2766893.

[GS05] S. J. Gaffney and P. Smyth. Joint probabilistic curve clustering and alignment. In *Advances in neural information processing systems*, pages 473–480, 2005.

[GSAH17] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466. IEEE, 2017.

[GSKJ08] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen. Snap-together motion: assembling run-time animations. *ACM SIGGRAPH 2008 classes*, pages 1–9, 2008.

[GVR+14] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormaehlen, P. Perez, and C. Theobalt. Automatic Face Reenactment. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4217–4224, June 2014. doi: 10.1109/CVPR.2014.537. arXiv: 1602.02651.

[GVS+15] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez, and C. Theobalt. VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track. *Computer Graphics Forum*, 34(2):193–204, May 2015. doi: 10.1111/cgf.12552.

[GVWT13] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics*, 32(6):1–10, Nov. 2013. doi: 10.1145/2508363.2508380.

[GWLM18] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018.

[Hav06] P. Havaldar. Sony Pictures Imageworks. In *ACM SIGGRAPH 2006 Courses on - SIGGRAPH '06*, page 5, Boston, Massachusetts, 2006. ACM Press. doi: 10.1145/1185657.1185845.

[HDF12] C. Huang, X. Ding, and C. Fang. Pose robust face tracking by combining view-based AAMs and temporal filters. *Computer Vision and Image Understanding*, 116(7):777–792, 2012.

[HdSP07] E. Hsu, M. da Silva, and J. Popović. Guided time warping for motion editing. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 45–52. Eurographics Association, 2007.

[HG07] R. Heck and M. Gleicher. Parametric motion graphs. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games - I3D '07*, page 129, Seattle, Washington, 2007. ACM Press. doi: 10.1145/1230100.1230123.

[HHC+19] S. Hong, D. Han, K. Cho, J. S. Shin, and J. Noh. Physics-based full-body soccer motion control for dribbling and shooting. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. Publisher: ACM New York, NY, USA.

[HHS+17] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, T. Komura, J. Saito, I. Kusajima, X. Zhao, M.-G. Choi, and R. Hu. A Recurrent Variational Autoencoder for Human Motion Synthesis. *IEEE Computer Graphics and Applications*, 37:4, 2017.

[HJ10] Hui Lou and Jinxiang Chai. Example-Based Human Motion Denoising. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):870–879, Sept. 2010. doi: 10.1109/TVCG.2010.23.

[HK08] C.-C. Hsieh and P.-L. Kuo. An impulsive noise reduction agent for rigid body motion data using B-spline wavelets. *Expert Systems with Applications*, 34(3):1733–1741, 2008. Publisher: Elsevier.

[HK10] Y. Huang and M. Kallmann. Motion parameterization with inverse blending. In *International Conference on Motion in Games*, pages 242–253. Springer, 2010.

[HKPP20] D. Holden, O. Kanoun, M. Perepichka, and T. Popa. Learned motion matching. *ACM Transactions on Graphics*, 39(4), July 2020. doi: 10.1145/3386569.3392440.

[HKS17] D. Holden, T. Komura, and J. Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics*, 36(4):1–13, July 2017. doi: 10.1145/3072959.3073663.

[HLL16] K. Hyun, K. Lee, and J. Lee. Motion Grammars for Character Animation. *Computer Graphics Forum*, 35(2):103–113, May 2016. doi: 10.1111/cgf.12815.

[HMYL15] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1675–1683, 2015.

[HP18] F. G. Harvey and C. Pal. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs on - SA '18*, pages 1–4, Tokyo, Japan, 2018. ACM Press. doi: 10.1145/3283254.3283277.

[HS97] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSK16] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics*, 35(4):1–11, July 2016. doi: 10.1145/2897824.2925975.

[HSKJ15] D. Holden, J. Saito, T. Komura, and T. Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, pages 1–4. ACM Press, 2015. doi: 10.1145/2820903.2820918.

[Hti17] K. K. Htike. A review on data-driven learning of a talking head model. *International Journal of Intelligent Systems Technologies and Applications*, 16(2):169–190, 2017.

[HYNP20] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal. Robust Motion In-betweening. 39(4):12, 2020.

[IKKP17] A.-E. Ichim, P. Kadleček, L. Kavan, and M. Pauly. Phace: physics-based face modeling and animation. *ACM Transactions on Graphics*, 36(4):1–14, July 2017. doi: 10.1145/3072959.3073664.

[IMH06] T. Igarashi, T. Moscovich, and J. F. Hughes. Spatial keyframing for performance-driven animation. *ACM SIGGRAPH 2006 Courses*, pages 17–es, 2006.

[IZZE17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, Honolulu, HI, July 2017. IEEE. doi: 10.1109/CVPR.2017.632.

[JCZ19] A. Jamaludin, J. S. Chung, and A. Zisserman. You Said That?: Synthesising Talking Faces from Audio. *International Journal of Computer Vision*, 127(11-12):1767–1779, Dec. 2019. doi: 10.1007/s11263-019-01150-y.

[JGGN15] M. Jin, D. Gopstein, Y. Gingold, and A. Nealen. AniMesh: interleaved animation, modeling, and editing. *ACM Transactions on Graphics*, 34(6):1–8, Nov. 2015. doi: 10.1145/2816795.2818114.

[JM04] O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to Isomap nonlinear dimension reduction. In *Twenty-first international conference on Machine learning - ICML '04*, page 56, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015357.

[JM08] D. Jurafsky and J. H. Martin. Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*, 2008.

[Jol86] I. T. Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[JP19] Y. Jo and J. Park. SC-FEGAN: Face Editing Generative Adversarial Network with User's Sketch and Color. *arXiv:1902.06838 [cs]*, Feb. 2019. URL `http://arxiv.org/abs/1902.06838`. arXiv: 1902.06838.

[JT81] O. Johnston and F. Thomas. *The illusion of life: Disney animation*. Disney Editions New York, 1981.

[JTDP03] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin. Learning Controls for Blend Shape Based Realistic Facial Animation. *SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 187–192, 2003.

[JZSS16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[KAHK17] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[KAL+17] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven Facial Animation by Joint End-to-end Learning of Pose and Emotion. *ACM Trans. Graph.*, 36(4):94:1–94:12, July 2017. doi: 10.1145/3072959.3073658.

[Kan10] N. Kanwisher. Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences of the United States of America*, 107(25):11163–11170, June 2010. doi: 10.1073/pnas.1005062107.

[KB84] D. H. Kochanek and R. H. Bartels. Interpolating splines with local tension, continuity, and bias control. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 33–41, 1984.

[KB14] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Dec. 2014. URL `http://arxiv.org/abs/1412.6980`. arXiv: 1412.6980.

[KBG+13] M. Kapadia, A. Beacco, F. Garcia, V. Reddy, N. Pelechano, and N. I. Badler. Multi-domain real-time planning in dynamic environments. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '13*, page 115, Anaheim, California, 2013. ACM Press. doi: 10.1145/2485895.2485909.

[KBK18] T. Kucherenko, J. Beskow, and H. Kjellström. A Neural Network Approach to Missing Marker Reconstruction in Human Motion Capture. *arXiv:1803.02665 [cs]*, Sept. 2018. URL `http://arxiv.org/abs/1803.02665`. arXiv: 1803.02665.

[KEG16] J. F. P. Kooij, G. Englebienne, and D. M. Gavrila. Mixture of Switching Linear Dynamics to Discover Behavior Patterns in Object Tracks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):322–334, Feb. 2016. doi: 10.1109/TPAMI.2015.2443801.

[KEZ+19] H. Kim, M. Elgharib, M. Zollhöfer, H.-P. Seidel, T. Beeler, C. Richardt, and C. Theobalt. Neural Style-Preserving Visual Dubbing. *ACM Transactions on Graphics*, 38(6):1–13, Nov. 2019. doi: 10.1145/3355089.3356500. arXiv: 1909.02518.

[KF01] Kiam Choo and D. Fleet. People tracking using hybrid Monte Carlo filtering. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 321–328, Vancouver,

BC, Canada, 2001. IEEE Comput. Soc. doi: 10.1109/ICCV.2001.937643.

[KG03] L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 214–224. Eurographics Association, 2003.

[KG04] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (ToG)*, 23(3):559–568, 2004.

[KG18] Y. Koyama and M. Goto. OptiMo: Optimization-Guided Motion Editing for Keyframe Character Animation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, Montreal QC, Canada, 2018. ACM Press. doi: 10.1145/3173574.3173735.

[KGP02] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *ACM SIGGRAPH*, page 482. ACM, 2002.

[Kim14] Y. Kim. Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882 [cs]*, Sept. 2014. URL `http://arxiv.org/abs/1408.5882`. arXiv: 1408.5882.

[KM04] T. Kurihara and N. Miyata. Modeling deformable human hands from medical images. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*, page 355, Grenoble, France, 2004. ACM Press. doi: 10.1145/1028523.1028571.

[KML18] M. A. Kiasari, D. S. Moirangthem, and M. Lee. Human Action Generation with Generative Adversarial Networks. *arXiv:1805.10416 [cs]*, May 2018. URL `http://arxiv.org/abs/1805.10416`. arXiv: 1805.10416.

[KMT03] S. Kshirsagar and N. Magnenat-Thalmann. Visyllable Based Speech Animation. *Computer Graphics Forum*, 22(3):631–639, Sept. 2003. doi: 10.1111/1467-8659.t01-2-00711.

[KN14] I. Kapsouras and N. Nikolaidis. Action recognition on motion capture data using a dynemes and forward differences representation. *Journal of Visual Communication and Image Representation*, 25(6):1432–1445, Aug. 2014. doi: 10.1016/j.jvcir.2014.04.007.

[Kor02] A. Kort. Computer aided inbetweening. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 125–132, 2002.

[KPS03] T.-h. Kim, S. I. Park, and S. Y. Shin. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (TOG)*, 22(3):392–401, 2003.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KTC+18] H. Kim, C. Theobalt, P. Carrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, and M. Zollhöfer. Deep video portraits. *ACM Transactions on Graphics*, 37(4):1–14, July 2018. doi: 10.1145/3197517.3201283.

[KYTM15] T. Kim, Y. Yue, S. Taylor, and I. Matthews. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 577–586. ACM, 2015.

[LA10] J. P. Lewis and K.-i. Anjyo. Direct Manipulation Blendshapes. *IEEE Computer Graphics and Applications*, 30(4):42–50, July 2010. doi: 10.1109/MCG.2010.41.

[LAGP09] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust Single-view Geometry and Motion Reconstruction. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 175:1–175:10, New York, NY, USA, 2009. ACM. doi: 10.1145/1661412.1618521.

[LAR+14] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng. Practice and Theory of Blendshape Facial Models. *Eurographics (State of the Art Reports)*, 1(8):2, 2014.

[Law04] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

[LBB+17] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6):1–17, Nov. 2017. doi: 10.1145/3130800.3130813.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. Publisher: MIT Press.

[LCR+02] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics*

and interactive techniques - SIGGRAPH '02, page 491, San Antonio, Texas, 2002. ACM Press. doi: 10.1145/566570.566607.

[LCXS09] M. Lau, J. Chai, Y.-Q. Xu, and H.-Y. Shum. Face poser: Interactive modeling of 3D facial expressions using facial priors. *ACM Transactions on Graphics*, 29(1):1–17, Dec. 2009. doi: 10.1145/1640443.1640446.

[LD08] Q. Li and Z. Deng. Orthogonal-Blendshape-Based Editing System for Facial Motion Capture Data. *IEEE Computer Graphics and Applications*, 28(6):76–82, Nov. 2008. doi: 10.1109/MCG.2008.120.

[LGN14] A. M. Lehrmann, P. V. Gehler, and S. 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. doi: 10.1109/CVPR.2014.171.

[LHP+17] X. Liu, G.-F. He, S.-J. Peng, Y.-m. Cheung, and Y. Y. Tang. Efficient Human Motion Retrieval via Temporal Adjacent Bag of Words and Discriminative Neighborhood Preserving Dictionary Learning. *IEEE Transactions on Human-Machine Systems*, 47(6):763–776, Dec. 2017. doi: 10.1109/THMS.2017.2675959.

[Li10] H. Li. *Animation reconstruction of deformable surfaces*. PhD thesis, ETH Zurich, 2010.

[LL04] J. Lee and K. H. Lee. Precomputing avatar behavior from human motion data. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 79–87, 2004.

[LLKP11] S. Levine, Y. Lee, V. Koltun, and Z. Popović. Space-time planning with parameterized locomotion controllers. *ACM Transactions on Graphics*, 30(3):1–11, May 2011. doi: 10.1145/1966394.1966402.

[LLL19] K. Lee, S. Lee, and J. Lee. Interactive character animation by learning multi-objective control. *ACM Transactions on Graphics*, 37(6):1–10, Jan. 2019. doi: 10.1145/3272127.3275071.

[LLX+01] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG)*, 20(3):127–150, July 2001. doi: 10.1145/501786.501787.

[LMPF10] L. Li, J. McCann, N. Pollard, and C. Faloutsos. BoLeRO: A Principled Technique for Including Bone Length Constraints in Motion Capture Occlusion Filling. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, page 10, 2010.

[LNLN06] M. W. Lee, R. Nevatia, M. W. Lee, and R. Nevatia. Human Pose Tracking Using Multi-level Structured Models. In J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3953, pages 368–381, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi: 10.1007/11744078_29.

[LO11] K. Liu and J. Ostermann. Realistic facial expression synthesis for an image-based talking head. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[LP02] C. K. Liu and Z. Popović. Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics (TOG)*, 21(3):408–416, 2002.

[LS99] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pages 39–48, Not Known, 1999. ACM Press. doi: 10.1145/311535.311539.

[LS02] J. Lee and S. Y. Shin. General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):119–128, 2002.

[LSP08] H. Li, R. W. Sumner, and M. Pauly. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum*, 27(5):1421–1430, July 2008. doi: 10.1111/j.1467-8659.2008.01282.x.

[LSSS18] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep Appearance Models for Face Rendering. *ACM Transactions on Graphics*, 37(4):1–13, July 2018. doi: 10.1145/3197517.3201401. arXiv: 1808.00362.

[LWB+10] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović. Motion fields for interactive character locomotion. *ACM SIGGRAPH Asia 2010 papers*, pages 1–8, 2010.

[LWH+12] S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics*, 31(4):1–10, Aug. 2012. doi: 10.1145/2185520.2185524.

[LWJ+19] Z. Liu, S. Wu, S. Jin, Q. Liu, S. Lu, R. Zimmermann, and L. 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. doi: 10.1109/CVPR.2019.01024.

[LWW+18] C. Lv, Z. Wu, X. Wang, D. Zhang, X. Liu, and M. Zhou. Facial Expression Editing in Face Sketch Using Shape Space Theory. In *2018 International Conference on Cyberworlds (CW)*, pages 33–40, Singapore, Oct. 2018. IEEE. doi: 10.1109/CW.2018.00019.

[LYYB13] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime Facial Animation with On-the-fly Correctives. *ACM Trans. Graph.*, 32(4):42:1–42:10, 2013. doi: 10.1145/2461912.2462019.

[LZWM05] G. Liu, J. Zhang, W. Wang, and L. McMillan. A system for analyzing and indexing human-motion databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05*, page 924, Baltimore, Maryland, 2005. ACM Press. doi: 10.1145/1066157.1066290.

[LZX+18] Z. Li, Y. Zhou, S. Xiao, C. He, Z. Huang, and H. Li. Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis. *arXiv:1707.05363 [cs]*, July 2018. URL `http://arxiv.org/abs/1707.05363`. arXiv: 1707.05363.

[LZZ+19] S. Li, Y. Zhou, H. Zhu, W. Xie, Y. Zhao, and X. Liu. Bidirectional recurrent autoencoder for 3D skeleton motion data refinement. *Computers & Graphics*, 81:92–103, 2019. Publisher: Elsevier.

[LZZL20] S.-J. Li, H.-S. Zhu, L.-P. Zheng, and L. Li. A Perceptual-Based Noise-Agnostic 3D Skeleton Motion Data Refinement Network. *IEEE Access*, 8:52927–52940, 2020. doi: 10.1109/ACCESS.2020.2980316. Conference Name: IEEE Access.

[MAO+11] J. C. Miranda, X. Alvarez, J. Orvalho, D. Gutierrez, A. A. Sousa, and V. Orvalho. Sketch express: facial expressions made easy. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling - SBIM '11*, page 87, Vancouver, British Columbia, Canada, 2011. ACM Press. doi: 10.1145/2021164.2021180.

[MAO+12] J. C. Miranda, X. Alvarez, J. Orvalho, D. Gutierrez, A. Augusto Sousa, and V. Orvalho. Sketch express: A sketching interface for facial animation. *Computers & Graphics*, 36(6):585–595, Oct. 2012. doi: 10.1016/j.cag.2012.03.002.

[MBR17] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. *arXiv:1705.02445 [cs]*, May 2017. URL `http://arxiv.org/abs/1705.02445`. arXiv: 1705.02445.

[MC95] D. W. Massaro and M. M. Cohen. Perceiving talking faces. *Current Directions in Psychological Science*, 4(4):104–109, 1995. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

[MC12] J. Min and J. Chai. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics*, 31(6):1–12, Nov. 2012. doi: 10.1145/2366145.2366172.

[MCC09] J. Min, Y.-L. Chen, and J. Chai. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics*, 29(1):1–12, Dec. 2009. doi: 10.1145/1640443.1640452.

[MCP+06] J. Ma, R. Cole, B. Pellom, W. Ward, and B. Wise. Accurate visible speech synthesis based on concatenating variable length motion capture data. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):266–276, 2006.

[MD18] L. Ma and Z. Deng. Real-Time Facial Expression Transformation for Monocular RGB Video: Real-Time Facial Expression Transformation for Monocular RGB Video. *Computer Graphics Forum*, Oct. 2018. doi: 10.1111/cgf.13586.

[MJC+08] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics*, 27(5):1–10, Dec. 2008. doi: 10.1145/1409060.1409074.

[MK05] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. *ACM SIGGRAPH 2005 Papers*, pages 1062–1070, 2005.

[MK14] K. Miki and R. Kakigi. Magnetoencephalographic study on facial movements. *Frontiers in Human Neuroscience*, 8, 2014. doi: 10.3389/fnhum.2014.00550.

[MKKY18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral Normalization for Generative Adversarial Networks. *arXiv:1802.05957 [cs, stat]*, Feb. 2018. URL `http://arxiv.org/abs/1802.05957`. arXiv: 1802.05957.

[MLCC17] U. Mall, G. R. Lal, S. Chaudhuri, and P. Chaudhuri. A Deep Recurrent Framework for Cleaning Motion Capture Data. *arXiv:1712.03380 [cs]*, Dec. 2017. URL `http://arxiv.org/abs/1712.03380`. arXiv: 1712.03380.

[MLD09] X. Ma, B. H. Le, and Z. Deng. Style learning and transferring for facial animation editing. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 123–132. ACM, 2009.

[MM76] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, 1976.

[MMK12] M. Mori, K. MacDorman, and N. Kageki. The Uncanny Valley [From the Field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100, June 2012. doi: 10.1109/MRA.2012.2192811.

[MO14] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, Nov. 2014. URL `http://arxiv.org/abs/1411.1784`. arXiv: 1411.1784.

[Mor70] M. Mori. The uncanny valley. *Energy*, 7:33–35, 1970.

[MP07] J. McCann and N. Pollard. Responsive characters from motion fragments. *ACM SIGGRAPH 2007 papers*, pages 6–es, 2007.

[MSM+17] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Interspeech 2017*, pages 498–502. ISCA, Aug. 2017. doi: 10.21437/Interspeech.2017-1386.

[MV15] W. Mattheyses and W. Verhelst. Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication*, 66:182–217, 2015.

[Neb99] J.-C. Nebel. Keyframe interpolation with self-collision avoidance. In W. Hansmann, W. Purgathofer, F. Sillion, N. Magnenat-Thalmann, and D. Thalmann, editors, *Computer Animation and Simulation '99*, pages 77–86. Springer Vienna, Vienna, 1999. doi: 10.1007/978-3-7091-6423-5_8.

[NF06] G. Nataneli and P. Faloutsos. Sketch-Based Facial Animation. *Eurographics / ACM SIGGRAPH Symposium on Computer Animation*, page 2, 2006.

[NFN00] J.-y. Noh, D. Fidaleo, and U. Neumann. Animated deformations with radial basis functions. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 166–174, 2000.

[NN98] J.-y. Noh and U. Neumann. A survey of facial modeling and animation techniques. Technical report, USC Technical Report, 99–705, 1998.

[NSX+19] K. Nagano, J. Seo, J. Xing, L. Wei, Z. Li, S. Saito, A. Agarwal, J. Fursund, and H. Li. paGAN: real-time avatars using dynamic textures. *ACM Transactions on Graphics*, 37(6):1–12, Jan. 2019. doi: 10.1145/3272127.3275075.

[NVW+13] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics*, 32(6):1–10, Nov. 2013. doi: 10.1145/2508363.2508417.

[OBP+12] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez. A Facial Rigging Survey. *Eurographics 2012 - State of the Art Reports*, page 22 pages, 2012. doi: 10.2312/CONF/EG2012/STARS/183-204.

[OCIM07] S. Ouni, M. M. Cohen, H. Ishak, and D. W. Massaro. Visual Contribution to Speech Perception: Measuring the Intelligibility of Animated Talking Heads. *EURASIP Journal on Audio, Speech, and Music Processing*, 2007:1–12, 2007. doi: 10.1155/2007/47891.

[ORBD05] M. S. Oh, J. Rehg, T. Balch, and F. Dellaert. Learning and inference in parametric switching linear dynamic systems. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 1161–1168 Vol. 2, Beijing, China, 2005. IEEE. doi: 10.1109/ICCV.2005.135.

[Par72] F. I. Parke. Computer generated animation of faces. In *Proceedings of the ACM annual conference - Volume 1*, ACM '72, pages 451–457, Boston, Massachusetts, USA, Aug. 1972. Association for Computing Machinery. doi: 10.1145/800193.569955.

[Par74] F. I. Parke. A parametric model for human faces. Technical report, UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE, 1974.

[PASH13] H. Y. Ping, L. N. Abdullah, P. S. Sulaiman, and A. A. Halin. Computer Facial Animation: A Review. *International Journal of Computer Theory and Engineering*, pages 658–662, 2013. doi: 10.7763/IJCTE.2013.V5.770.

[PCP17] H. X. Pham, S. Cheung, and V. Pavlovic. Speech-Driven 3D Facial Animation with Implicit Emotional Awareness: A Deep Learning Approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2328–2336, Honolulu, HI, USA, July 2017. IEEE. doi: 10.1109/CVPRW.2017.287.

[PGB03] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[PHMP19] M. Perepichka, D. Holden, S. P. Mudur, and T. Popa. Robust Marker Trajectory Repair for MOCAP using Kinematic Reference. In *Motion, Interaction and Games on - MIG '19*, pages 1–10, Newcastle upon Tyne, United Kingdom, 2019. ACM Press. doi: 10.1145/3359566.3360060.

[PKA+09] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Genova, Italy, Sept. 2009. IEEE. doi: 10.1109/AVSS.2009.58.

[Ple03] Pless. Image spaces and video trajectories: using Isomap to explore video sequences. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1433–1440 vol.2, Nice, France, 2003. IEEE. doi: 10.1109/ICCV.2003.1238658.

[PLH04] H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without ICP. *Computer Vision and Image Understanding*, 95(1):54–71, July 2004. doi: 10.1016/j.cviu.2004.04.002.

[Pol64] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, Jan. 1964. doi: 10.1016/0041-5553(64)90137-5.

[PRM01] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Advances in neural information processing systems*, pages 981–987, 2001.

[PSS99] F. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3D model-based tracking. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 143–150 vol.1, Kerkyra, Greece, 1999. IEEE. doi: 10.1109/ICCV.1999.791210.

[PSS02] S. I. Park, H. J. Shin, and S. Y. Shin. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 105–111, 2002.

[RAAL19] R. A. Roberts, R. K. D. Anjos, K. Anjyo, and J. P. Lewis. Piku Piku Interpolation. In *SIGGRAPH Asia 2019 Technical Briefs on - SA '19*, pages 53–56, Brisbane, QLD, Australia, 2019. ACM Press. doi: 10.1145/3355088.3365156.

[RAY+16] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. *33rd International Conference on Machine Learning*, pages 1060–1069, 2016.

[RB05] G. Roussos and B. J. C. Baxter. Rapid evaluation of radial basis functions. *Journal of Computational and Applied Mathematics*, 180(1):51–70, Aug. 2005. doi: 10.1016/j.cam.2004.10.002.

[RBSB18] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3D faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018.

[RC19] K. Reed and D. Cosker. User Guided Facial Animation through an Evolutionary Interface. *Computer Graphics Forum*, Mar. 2019.

[RCB98] C. Rose, M. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, Oct. 1998. doi: 10.1109/38.708559.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 9351:234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

[RGMN19] A. H. Ruiz, J. Gall, and F. Moreno-Noguer. Human Motion Prediction via Spatio-Temporal Inpainting. *arXiv:1812.05478 [cs]*, Oct. 2019. URL `http://arxiv.org/abs/1812.05478`. arXiv: 1812.05478.

[RHD12] B. Rossion, B. Hanseeuw, and L. Dricot. Defining face perception areas in the human brain: A large-scale factorial fMRI face localizer analysis. *Brain and Cognition*, 79(2):138–157, July 2012. doi: 10.1016/j.bandc.2012.01.001.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[RISC01] C. F. Rose III, P.-P. J. Sloan, and M. F. Cohen. Artist-Directed Inverse-Kinematics Using Radial Basis Function Interpolation. *Computer Graphics Forum*, 20(3):239–250, Sept. 2001. doi: 10.1111/1467-8659.00516.

[RJ86] L. Rabiner and B. Juang. An introduction to hidden Markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, Quebec City, Que., Canada, 2001. IEEE Comput. Soc. doi: 10.1109/IM.2001.924423.

[RS00] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[RZL+17] R. B. i. Ribera, E. Zell, J. P. Lewis, J. Noh, and M. Botsch. Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics*, 36(4):1–12, July 2017. doi: 10.1145/3072959.3073674.

[RZS10] C. Ren, L. Zhao, and A. Safonova. Human Motion Synthesis with Optimization-based Graphs. *Computer Graphics Forum*, 29(2):545–554, May 2010. doi: 10.1111/j.1467-8659.2009.01624.x.

[SAD+08] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56. Springer, 2008.

[SBS10] N. Stoiber, G. Breton, and R. Seguier. Modeling Short-Term Dynamics and Variability for Realistic Interactive Facial Animation. *IEEE Computer Graphics and Applications*, 30(4):51–61, July 2010. doi: 10.1109/MCG.2010.40.

[Sca04] R. A. Scarborough. *Coarticulation and the structure of the lexicon*. University of California, Los Angeles, 2004.

[Seo11] J. Seo. Compression and Direct Manipulation of Complex Blendshape Models. *ACM Trans. Graph*, 30(6):164:1–164:10, Dec. 2011.

[SG15] F. Simion and E. D. Giorgio. Face perception and processing in early infancy: inborn predispositions and developmental changes. *Frontiers in Psychology*, 6, July 2015. doi: 10.3389/fpsyg.2015.00969.

[SHK+14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. Publisher: JMLR. org.

[SHP04] A. Safonova, J. K. Hodgins, and N. S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics*, 23(3):514, Aug. 2004. doi: 10.1145/1015706.1015754.

[SHT09] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608, 2009.

[SKY12] H. P. Shum, T. Komura, and S. Yamazaki. Simulating Multiple Character Interactions with Collaborative and Adversarial Goals. *IEEE*

Transactions on Visualization and Computer Graphics, 18(5):741–752, May 2012. doi: 10.1109/TVCG.2010.257.

[SL14] Y. Seol and J. P. Lewis. Tuning facial animation in a mocap pipeline. In *ACM SIGGRAPH 2014 Talks*, page 13. ACM, 2014.

[SLS+12] Y. Seol, J. P. Lewis, J. Seo, B. Choi, K. Anjyo, and J. Noh. Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG)*, 31(2):14, 2012.

[SLSG01] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Transactions on Graphics*, 20(2):67–94, Apr. 2001. doi: 10.1145/502122.502123.

[SMDH13] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[SMH11] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024, 2011.

[SMM05] M. Srinivasan, R. A. Metoyer, and E. N. Mortensen. Controllable real-time locomotion using mobility maps. In *Proceedings of Graphics Interface 2005*, pages 51–59, 2005.

[SMND08] T. Sucontphunt, Z. Mo, U. Neumann, and Z. Deng. Interactive 3D facial expression posing through 2D portrait manipulation. In *Graphics interface*, pages 177–184. Citeseer, 2008.

[SNF05] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM SIGGRAPH 2005 Papers*, pages 417–425, 2005.

[SO06] H. J. Shin and H. S. Oh. Fat graphs: constructing an interactive character with continuous controls. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 291–298. Eurographics Association, 2006.

[SP54] W. H. Sumby and I. Pollack. Visual Contribution to Speech Intelligibility in Noise. *The Journal of the Acoustical Society of America*, 26(2):212–215, Mar. 1954. doi: 10.1121/1.1907309.

[SP04] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, 2004. doi: 10.1145/1015706.1015736.

[SSB08] N. Stoiber, R. Seguier, and G. Breton. Automatic design of a control interface for a synthetic face. In *Proceedingsc of the 13th international conference on Intelligent user interfaces - IUI '09*, page 207, Sanibel Island, Florida, USA, 2008. ACM Press. doi: 10.1145/1502650.1502681.

[SSK+11] Y. Seol, J. Seo, P. H. Kim, J. P. Lewis, and J. Noh. Artist friendly facial animation retargeting. *ACM Transactions on Graphics*, 30(6):162, Dec. 2011. doi: 10.1145/2070781.2024196.

[SSK+12] Y. Seol, J. Seo, P. H. Kim, J. P. Lewis, and J. Noh. Weighted pose space editing for facial animation. *The Visual Computer*, 28(3):319–327, Mar. 2012. doi: 10.1007/s00371-011-0641-4.

[SSKS17] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics*, 36(4):1–13, July 2017. doi: 10.1145/3072959.3073640.

[ST03] V. D. Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728, 2003.

[SWQ+20] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy. Everybody's Talkin': Let Me Talk as You Want. *arXiv:2001.05201 [cs]*, Jan. 2020. URL http://arxiv.org/abs/2001.05201. arXiv: 2001.05201.

[SZ15] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Apr. 2015. URL http://arxiv.org/abs/1409.1556. arXiv: 1409.1556.

[SZKS19] S. Starke, H. Zhang, T. Komura, and J. Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6):1–14, Nov. 2019. doi: 10.1145/3355089.3356505.

[TDlTM11] J. R. Tena, F. De la Torre, and I. Matthews. Interactive Region-based Linear 3D Face Models. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 76:1–76:10, New York, NY, USA, 2011. ACM. doi: 10.1145/1964921.1964971. event-place: Vancouver, British Columbia, Canada.

[TDSL00] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[TET+19] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner. Neural Voice Puppetry: Audio-driven Facial Reenactment. *arXiv:1912.05566 [cs]*, Dec. 2019. URL http://arxiv.org/abs/1912.05566. arXiv: 1912.05566.

[TFT+20] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B. Goldman, and M. Zollhöfer. State of the Art on Neural Rendering. *arXiv:2004.03805 [cs]*, Apr. 2020. URL `http://arxiv.org/abs/2004.03805`. arXiv: 2004.03805.

[TH09] G. W. Taylor and G. E. Hinton. Factored conditional restricted Boltzmann Machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada, 2009. ACM Press. doi: 10.1145/1553374.1553505.

[THR07] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2007.

[THR11] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Two distributed-state models for generating high-dimensional time series. *Journal of Machine Learning Research*, 12(Mar):1025–1068, 2011.

[TKMK99] M. Tamura, S. Kondo, T. Masuko, and T. Kobayashi. Text-to-audio-visual speech synthesis based on parameter generation from HMM. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[TKY+17] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics*, 36(4):1–11, July 2017. doi: 10.1145/3072959.3073699.

[TLP07] A. Treuille, Y. Lee, and Z. Popović. Near-optimal character animation with continuous control. *ACM SIGGRAPH 2007 papers*, pages 7–es, 2007.

[TMLZ18] Y. Tang, L. Ma, W. Liu, and W.-S. Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 935–941, 2018.

[TMTM12] S. L. Taylor, M. Mahler, B.-J. Theobald, and I. Matthews. Dynamic Units of Visual Speech. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, page 10 pages, 2012. doi: 10.2312/SCA/SCA12/275-284.

[Tov95] M. J. Tovée. Face Recognition: What are faces for? *Current Biology*, 5(5):480–482, May 1995. doi: 10.1016/S0960-9822(95)00096-0.

[TS06] T. Tangkuampien and D. Suter. Human Motion De-noising via Greedy Kernel Principal Component Analysis Filtering. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 457–460, Hong Kong, China, 2006. IEEE. doi: 10.1109/ICPR.2006.639.

[TZK+17] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. *arXiv:1703.10580 [cs]*, Mar. 2017. URL `http://arxiv.org/abs/1703.10580`. arXiv: 1703.10580.

[TZN+15] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics*, 34(6):1–14, Nov. 2015. doi: 10.1145/2816795.2818056.

[TZS+16] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nie\s sner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[UAT95] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, pages 91–96, Not Known, 1995. ACM Press. doi: 10.1145/218380.218419.

[UFG+08] R. Urtasun, D. J. Fleet, A. Geiger, J. Popović, T. J. Darrell, and N. D. Lawrence. Topologically-constrained latent variable models. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 1080–1087, Helsinki, Finland, 2008. ACM Press. doi: 10.1145/1390156.1390292.

[VBPP05] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In *ACM transactions on graphics (TOG)*, volume 24, pages 426–433. ACM, 2005.

[VKK14] A. Vögele, B. Krüger, and R. Klein. Efficient Unsupervised Temporal Segmentation of Human Motion. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, page 10 pages, 2014. doi: 10.2312/SCA.20141135.

[VLL+10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[VPP18] K. Vougioukas, S. Petridis, and M. Pantic. End-to-End Speech-Driven Facial Animation with Temporal GANs. *BMVC*, July 2018. URL http://arxiv.org/abs/1805.09313. arXiv: 1805.09313.

[VPP19] K. Vougioukas, S. Petridis, and M. Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, pages 1–16, 2019.

[VWB+12] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics*, 31(6):1, Nov. 2012. doi: 10.1145/2366145.2366206.

[WBLP11] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM transactions on graphics (TOG)*, volume 30, page 77. ACM, 2011.

[WCP10] W. Wang and M. A. Carreira-Perpinan. Manifold blurring mean shift algorithms for manifold denoising. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1759–1766, San Francisco, CA, USA, June 2010. IEEE. doi: 10.1109/CVPR.2010.5539845.

[WCX19] Z. Wang, J. Chai, and S. Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. *IEEE transactions on visualization and computer graphics*, 2019.

[Wer88] P. J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, Jan. 1988. doi: 10.1016/0893-6080(88)90007-X.

[WFH08] J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, Feb. 2008. doi: 10.1109/TPAMI.2007.1167.

[WH97] D. Wiley and J. Hahn. Interpolation synthesis for articulated figure motion. In *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality*, pages 156–160, Albuquerque, NM, USA, 1997. IEEE Comput. Soc. Press. doi: 10.1109/VRAIS.1997.583065.

[WHL+04] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. *Computer Graphics Forum*, 23(3):677–686, Sept. 2004. doi: 10.1111/j.1467-8659.2004.00800.x.

[Wil06] L. Williams. Performance-driven facial animation. *Acm SIGGRAPH 2006 Courses*, pages 16–es, 2006.

[WK88] A. Witkin and M. Kass. Spacetime constraints. *ACM Siggraph Computer Graphics*, 22(4):159–168, 1988.

[WKT06] S.-K. Weng, C.-M. Kuo, and S.-K. Tu. Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208, 2006.

[WKZ+18] O. Wiles, A. S. Koepke, A. Zisserman, O. Wiles, A. S. Koepke, and A. Zisserman. X2Face: A Network for Controlling Face Generation Using Images, Audio, and Pose Codes. In Y. Weiss, editor, *Computer Vision – ECCV 2018*, volume 11217, pages 690–706, Cham, 2018. Springer International Publishing. doi: 10.1007/978-3-030-01261-8_41.

[WLQ+16] Z. Wang, S. Liu, R. Qian, T. Jiang, X. Yang, and J. J. Zhang. Human motion data refinement unitizing structural sparsity and spatial-temporal information. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 975–982, Chengdu, China, Nov. 2016. IEEE. doi: 10.1109/ICSP.2016.7877975.

[WMC11] X. Wei, J. Min, and J. Chai. Physically valid statistical models for human motion generation. *ACM Transactions on Graphics*, 30(3):1–10, May 2011. doi: 10.1145/1966394.1966398.

[WMK05] S. Watanabe, K. Miki, and R. Kakigi. Mechanisms of face perception in humans: A magneto- and electro-encephalographic study. *Neuropathology*, 25(1):8–20, 2005. doi: 10.1111/j.1440-1789.2004.00603.x.

[WNS+10] B. Whited, G. Noris, M. Simmons, R. W. Sumner, M. Gross, and J. Rossignac. BetweenIT: An Interactive Tool for Tight Inbetweening. *Computer Graphics Forum*, 29(2):605–614, May 2010. doi: 10.1111/j.1467-8659.2009.01630.x.

[WP95] A. Witkin and Z. Popovic. Motion warping. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 105–108, 1995.

[WSS+19] S.-E. Wei, J. Saragih, T. Simon, A. W. Harley, S. Lombardi, M. Perdoch, A. Hypes, D. Wang, H. Badino, and Y. Sheikh. VR facial animation via multiview image translation. *ACM Transactions on Graphics (TOG)*, 38(4):67, 2019.

[WV18] K. Westbrook and M. Varacallo. Anatomy, Head and Neck, Facial Muscles. *StatPearls Publishing*, 2018. Publisher: StatPearls Publishing, Treasure Island (FL).

[XCLT14] F. Xu, J. Chai, Y. Liu, and X. Tong. Controllable high-fidelity facial performance transfer. *ACM Transactions on Graphics*, 33(4):1–11, July 2014. doi: 10.1145/2601097.2601210.

[XFJ+15] J. Xiao, Y. Feng, M. Ji, X. Yang, J. J. Zhang, and Y. Zhuang. Sparse motion bases selection for human motion denoising. *Signal Processing*, 110:108–122, May 2015. doi: 10.1016/j.sigpro.2014.08.017.

[XWCL15] B. Xu, N. Wang, T. Chen, and M. Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv:1505.00853 [cs, stat]*, Nov. 2015. URL http://arxiv.org/abs/1505.00853. arXiv: 1505.00853.

[YAMZ+15] I. Yoo, M. Abdul Massih, I. Ziamtsov, R. Hassan, and B. Benes. Motion retiming by using bilateral time control surfaces. *Computers & Graphics*, 47:59–67, Apr. 2015. doi: 10.1016/j.cag.2014.11.001.

[YL10] Y. Ye and C. K. Liu. Synthesis of responsive motion using a dynamic model. In *Computer Graphics Forum*, volume 29, pages 555–562. Wiley Online Library, 2010.

[YLY+19] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang. Free-Form Image Inpainting with Gated Convolution. *arXiv:1806.03589 [cs]*, Oct. 2019. URL http://arxiv.org/abs/1806.03589. arXiv: 1806.03589.

[YN03] K. Yamane and Y. Nakamura. Dynamics filter-concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, June 2003. doi: 10.1109/TRA.2003.810579.

[YYZ+20] R. Yi, Z. Ye, J. Zhang, H. Bao, and Y.-J. Liu. Audio-driven Talking Face Video Generation with Learning-based Personalized Head Pose. *arXiv:2002.10137 [cs]*, Mar. 2020. URL http://arxiv.org/abs/2002.10137. arXiv: 2002.10137.

[ZHK+17] L. Zhao, F. Han, M. Kapadia, V. Pavlovic, and D. Metaxas. Sketch-based Face Editing in Video Using Identity Deformation Transfer. *arXiv preprint arXiv:1703.08738*, 2017.

[ZHP+19] L. Zhao, F. Han, X. Peng, X. Zhang, M. Kapadia, V. Pavlovic, and D. N. Metaxas. Cartoonish sketch-based face editing in videos using identity deformation transfer. *Computers & Graphics*, 79:58–68, Apr. 2019. doi: 10.1016/j.cag.2019.01.004. arXiv: 1703.08738.

[ZHT06] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, June 2006. doi: 10.1198/106186006X113430.

[ZJS15] Y. Zhao, D. Jiang, and H. Sahli. 3D emotional facial animation synthesis with factored conditional Restricted Boltzmann Machines. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 797–803. IEEE, 2015.

[ZLB+20] Y. Zhou, J. Lu, C. Barnes, J. Yang, S. Xiang, and H. li. Generative Tweening: Long-term Inbetweening of 3D Human Motions. *arXiv:2005.08891 [cs]*, May 2020. URL http://arxiv.org/abs/2005.08891. arXiv: 2005.08891.

[ZLH03] Q. Zhang, Liu ,Zicheng, and Heung-Yeung Shum. Geometry-driven photorealistic facial expression synthesis. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–186, 2003. doi: 10.1109/TVCG.2006.9.

[ZLH+20] Y. Zhou, D. Li, X. Han, E. Kalogerakis, E. Shechtman, and J. Echevarria. MakeItTalk: Speaker-Aware Talking Head Animation. *arXiv:2004.12992 [cs]*, Apr. 2020. URL http://arxiv.org/abs/2004.12992. arXiv: 2004.12992.

[ZLL+19] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, 2019.

[ZPW+20] X. Zeng, Y. Pan, M. Wang, J. Zhang, and Y. Liu. Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose. *arXiv:2003.12957 [cs]*, Mar. 2020. URL http://arxiv.org/abs/2003.12957. arXiv: 2003.12957.

[ZSBL19] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9459–9468, 2019.

[ZSCS04] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime Faces: High Resolution Capture for Modeling and Animation. *ACM Trans. Graph.*, 23:548–558, 2004.

[ZSKS18] H. Zhang, S. Starke, T. Komura, and J. Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics*, 37(4):1–11, July 2018. doi: 10.1145/3197517.3201366.

[ZTG+18] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications.

Computer Graphics Forum, 37(2):523–550, May 2018. doi: 10.1111/cgf.13382.

[ZvdP18] X. Zhang and M. van de Panne. Data-driven autocompletion for keyframe animation. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, pages 1–11, Limassol Cyprus, Nov. 2018. ACM. doi: 10.1145/3274247.3274502.

[ZWL+13] X. Zhang, L. Wang, G. Li, F. Seide, and F. K. Soong. A New Language Independent, Photo-Realistic Talking Head Driven by Voice Only. *Interspeech*, pages 2743–2747, 2013.

[ZWR+20] W. Zhuang, Y. Wang, J. Robinson, C. Wang, M. Shao, Y. Fu, and S. Xia. Towards 3D Dance Motion Synthesis and Control. *arXiv:2006.05743 [cs]*, June 2020. URL http://arxiv.org/abs/2006.05743. arXiv: 2006.05743.

[ZXL+18] Y. Zhou, Z. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh. Visemenet: Audio-driven animator-centric speech animation. *ACM Transactions on Graphics (TOG)*, 37(4):161, 2018.

[ZYC+13] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and Peng Liu. A high-resolution spontaneous 3D dynamic facial expression database. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6, Shanghai, China, Apr. 2013. IEEE. doi: 10.1109/FG.2013.6553788.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

CentraleSupélec

Titre : L'Utilisation des Méthodes d'Apprentissage Profond au service de l'Édition d'Animation Faciale 3D

Mots-clés : Animation Faciale, Edition d'Animation, Deep Learning

Résumé : L'animation faciale vise à mouvoir des personnages virtuels afin de leur donner vie. Produire une animation faciale satisfaisante est une tâche difficile, notamment parce que l'œil humain est affûté à analyser les visages, et très sensible aux moindres défauts. Les technologies de capture de mouvements ont ouvert de nouvelles possibilités, facilitant la génération d'animations réalistes. La promesse de ces technologies est de parfaitement capturer et retranscrire toute la subtilité des mouvements faciaux, et du signal expressif et émotionnel qu'il diffuse. En pratique, même les systèmes les plus aboutis échouent à accomplir parfaitement ce rôle. L'édition d'animation est alors une étape cruciale dans la production d'animations, mais demeure une étape très couteuse. En effet, manipuler une animation requiert un certain savoir-faire et une grande quantité de travail pour répondre aux exigences de qualités attendues. Nous proposons d'améliorer le processus d'édition en mettant à profit les récentes méthodes d'apprentissage, qui permettent de produire en grande quantité, des animations convaincantes. Ces méthodes apprennent les caractéristiques et les spécificités des dynamiques du visage, afin de synthétiser de nouvelles animations réalistes. Nous proposons une nouvelle méthode de filtrage d'animation qui apprend à nettoyer et même restaurer les dynamiques des différents paramètres d'animation. Cette thèse étudie également de nouvelles possibilités de modifier une animation et de diriger la synthèse d'animation à partir de paramètres de contrôle originaux, intuitifs et sémantiques. Enfin, cette thèse explore les méthodes d'apprentissage génératives, permettant la synthèse et l'édition, non ou faiblement supervisée, d'une animation.

Title: Leveraging Neural Networks for 3D Facial Animation Editing

Keywords: Facial Animation, Motion Editing, Neural Networks

Abstract: Facial animation consists of breathing life into computer graphic characters. The major challenge of facial animation is to fool the human's eyes, very acute at recognizing natural motions. The emergence and increasing availability of motion capture (MoCap) technologies have opened a new era, where realistic animation generation is more deterministic and repeatable. The theoretical promise of MoCap is the ability to capture and retarget human performance completely and flawlessly, from emotion down to the most subtle motion of facial skin. In reality, even professional motion capture setups often fall short of a perfect animation. Animation editing is therefore unavoidable and frequently the bottleneck of modern performance-based animation pipelines, requiring a considerable amount of time and special skills. We propose to improve the facial animation signal editing process by leveraging recent deep learning techniques. These methods enable producing realistic motion sequences, endowed with all the specificities included in the real animation used for training. We adopt this technology to enhance the cleaning facial animation editing process, providing a solution preserving and even restoring the facial motion dynamics. We also explore new possibilities to synthesize facial animations from alternative high-level control inputs, more semantic and intuitive than the current standard animation parameters. Pushing forward the motion editing toward user controllability, we provide a robust and interactive regressive neural-based method to modify a facial animation efficiently, from few temporally dense parameters. To bypass the limitations of regression algorithms, we explore generative methods suitable for unsupervised and partially supervised motion editing.