Annotation et synthèse basée données des expressions faciales de la Langue des Signes Française

Clément Reverdy

UNIVERSITE
BRETAGNE
LOIRE / MATHSTIC

THÈSEDE DOCTORA

L'UNIVERSITEDEBRETAGNESUD
COMUE**U**NIVERSITE**B**RETAGNE**L**OIR
E

EcoleDoctoraleN°601
MathSTIC:MathématiquesetSciencesetd el'InformationetdelaCommunication
Spécialité:InformatiquePar

«ClémentREVER D

«Annotationetsynthèsebasée LanguedesSignesFrançaise»

«»

ThèseprésentéeetsoutenueàVANNES,le
Unitéderecherche:IRISA,UMR6074Thèse N°:Ordre550

Rapporteursavantsoutenance:
CatherinePELACHAUD,Directricede RechercheJoaquimJORGE,Professeurdes universités

Compositiondujury:

Président:DamienROHMER,Professeur

Dir.dethèse:SylvieGIBET,Professeure

Co-enc.dethèse:CarolineLARBOULETTE,

ACKNOWLEDGEMENT

Je remercie naturellement mes deux directrices de thèse, Sylvie Gibet et Caroline Larboulette qui m'ont soutenu au cours des années (en particulier les dernières) qu'a duré cette thèse et aussi Pierre-François Marteau qui m'a introduit dans l'univers de la recherche au travers de deux stages durant mon master.

Je tiens également à remercier tous les collègues de l'IRISA-UBS et en particulier les doctorants et autres qui ont partagé mon quotidien et dont la fréquentation a été un enrichissement, Lucie, Mathieu, Pamela, Maël, Armel, Kader, Raounak, Lei, Nan, Delphine, Lionel, Fadhlallah, Romain, Thibaut, Rémi, Romain, Jean-Christophe, Tiago, Elyes, ... J'oublie des noms. Je ne suis pas particulièrement expansif mais j'ai eu beaucoup de plaisir en votre compagnie. Bon courage à ceux qui n'ont pas encore soutenu.

Je remercie encore Lucie et Mathieu. Ainsi qu'Alan.

Enfin je remercie ma famille et mes cercles d'amis de Vannes et d'ailleurs.

TABLE OF CONTENTS

CHAPITRE 1

INTRODUCTION

La Langue des Signes Française (LSF), utilisée en France par la communauté des sourds et leur entourage, représente une part majeure de la culture et de l'identité de cette dernière. Le développement d'outils informatiques éducatifs, de divertissement et de travail tenant compte de leur langue naturelle est apprécié des 100000 locuteurs français (environ 169000 dans le monde) [1]. Outre la vidéo, l'un des moyens permettant la prise en charge de cette langue qui ne connaît à l'heure actuelle pas de forme écrite, est la génération de contenu par le biais de personnages virtuels animés appelés avatars signeurs.

En LSF, l'expressivité faciale revêt une importance particulière [HLR11] [SWMT13] car elle est le vecteur de nombreuses informations, parmi lesquelles des informations affectives (liées à l'état émotionnel), adjectivales (liées par exemple à l'emphase associée à des adjectifs ou des adverbes) ou clausales (liées aux clauses interrogatives, négatives ou impératives), et sans lesquelles la compréhension de la langue ne peut être que partielle. Dans cette thèse nous nous sommes intéressés aux expressions faciales affectives en considérant plus spécifiquement les émotions d'Eckman [EF78].

Le système d'analyse / synthèse d'expressions faciales que nous proposons s'intègre dans un projet plus général de synthèse gestuelle de la LSF par concaténation qui permet d'éditer des morceaux de mouvements capturés, c'est-à-dire de les couper / coller / transformer / mixer afin de construire de nouvelles phrases [GCDLN11, LAGT$^+$13]. Ce système nécessite d'avoir à disposition une base de données annotée qui comprend à la fois des mouvements et des informations sémantiques qui étiquettent ces mouvements [ACD$^+$], de façon à produire de nouvelles animations à partir de phrases énoncées comme une combinaison d'annotations. Cependant, son utilisation nécessite de disposer d'un corpus conséquent de données annotées, ce qui représente un travail humain important et fastidieux.

Nos corpus, que ce soit pour les mouvements corporels ou faciaux, sont constitués à partir de données de capture de mouvement (*mocap*) qui présentent l'avantage de fournir une grande précision tant spatiale que temporelle (fréquence d'acquisition > 200 fps) ainsi que l'exploita-

1. https ://www.ethnologue.com/language/fsl

tion d'une technologie unique et synchrone pour les deux types de mouvements.

Dans cette thèse, nous proposons un système complet de synthèse faciale qui s'appuie sur des données capturées et annotées automatiquement. Ce système permet (i) de générer les paramètres nécessaires pour l'animation faciale, et (ii) de produire de manière semi-automatique les annotations correspondantes à partir de ces mêmes paramètres.

En matière d'animation faciale, les traitements peuvent varier suivant la nature des données dont on dispose initialement [RGL15a]. Dans le cadre des méthodes employées avec des données issues de la capture du mouvement basée marqueurs, nous évoquerons deux principales familles de méthodes. Tout d'abord, nous retrouvons les méthodes basées Laplacien comme le modèle *thin-shell* [BS08, BBA$^+$07a, LZD13] où un sous-ensemble de sommets du maillage cible est contraint et suit les mouvement des marqueurs qui leur sont associés, alors que les autres sommets sont modifiés de façon à minimiser la déformation du maillage. Ce type de méthode est jusqu'à un certain point compatible avec du temps-réel, cela dépend du nombre total de sommets du maillage cible. Le second type principalement utilisé est celui basé *blendshapes* [LAR$^+$, SLS$^+$12a, DCFN06a]. Il s'agit d'une représentation constituée d'une expression neutre du maillage et d'un ensemble d'expressions unitaires (dites bases) représentant chacune une expression faciale particulière exprimée différentiellement par rapport à l'expression neutre. Une expression E quelconque est alors représentée comme l'expression neutre B_{neutre} à laquelle on ajoute une combinaison linéaire de l'ensemble des bases B_i : $E = B_{neutre} + \sum_{i=0}^{b} w_i B_i$. Cette représentation par *blendshapes* est intéressante pour plusieurs raisons. Elle permet un stockage relativement compact (dans notre cas une cinquantaine de coefficients par *frame*). De plus il s'agit d'une représentation avec un haut niveau d'abstraction pouvant constituer un descripteur efficace pour un module de reconnaissance visant à annoter automatiquement nos corpus. Enfin, le vecteur de coefficients représente une couche d'abstraction entre les données brutes capturées et le module d'annotation car il ne dépend pas de la morphologie de l'acteur mais uniquement du modèle (maillage + bases *blendshapes*) ce qui facilitera la tâche lors de la constitution de corpus multi-acteurs.

1.1 Contributions et méthodologie

Dans cette thèse, nous nous intéressons à l'annotation et à la synthèse des expressions faciales. L'objectif final est d'intégrer l'expressivité faciale à un système de génération de phrases en Langue des Signes Française par synthèse concaténative.

Notre objectif est dual. D'une part nous souhaitons obtenir automatiquement les animations

correspondant à nos données *mocap* ; d'autre part nous voulons annoter ces données aussi automatiquement que possible.

Dans cette optique, notre première contribution consiste à adopter une démarche globale cohérente afin de résoudre ce double-objectif. Formulée de façon générale, l'idée de base est de considérer que pour obtenir les paramètres d'animation faciale à partir des données de mouvement enregistrées, des descripteurs précis sont extraits de ces données brutes et nous faisons l'hypothèse que ces mêmes descripteurs constituent une bonne entrée du système d'annotation automatique de l'expression faciale. Dans notre cas nous avons choisi d'utiliser une représentation par *blendshapes* (voir figure 1.1). Il s'agit d'une méthode couramment utilisée en animation faciale qui présente certaines caractéristiques que nous jugeons intéressantes à la fois pour la synthèse, mais aussi pour la reconnaissance automatique d'expressions faciales. En effet, la représentation des expressions faciales par *blendshapes* présente l'avantage d'être relativement compacte (dimension 51 dans notre cas) et signifiante car chaque forme de base représente un mouvement unitaire, ces formes de bases étant inspirées du FACS [EF78]. Un autre intérêt de reprendre la même représentation en synthèse qu'en reconnaissance est que cela permet de créer une couche d'abstraction permettant de passer outre les différences morphologiques entre acteurs. En effet, l'avatar sur lequel sont transférées les expressions faciales des différents humains peut faire office de "proxy". Cet avantage trouvera son intérêt dans des travaux futurs visant à exploiter simultanément les données de divers acteurs.

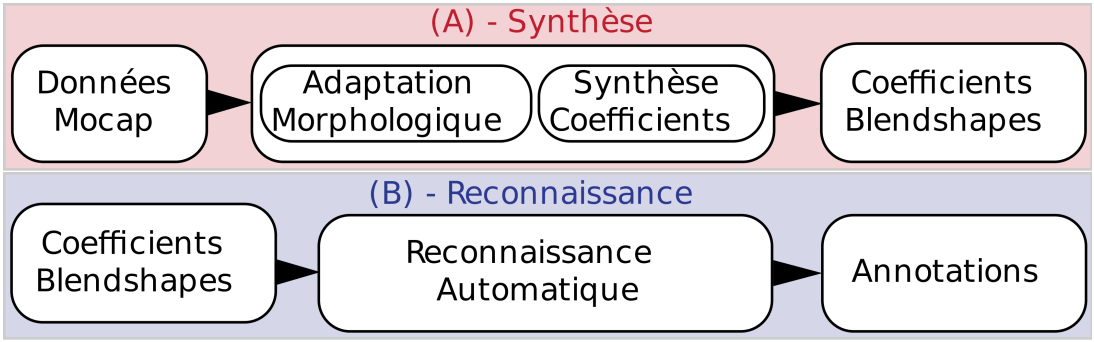


FIGURE 1.1 – Schéma d'ensemble : à partir de données *mocap*, nous calculons des coefficients de *blendshapes* afin de faire de la synthèse d'expressions faciales. Ces mêmes coefficients sont utilisés pour faire la reconnaissance automatique de ces expressions et annoter nos corpus.

Notre seconde contribution concerne le développement d'une approche d'annotation automatique qui s'appuie sur la reconnaissance d'expressions faciales émotionnelles par des tech-

niques d'apprentissage automatique. Les travaux réalisés dans ce domaine ont fait l'objet d'une publication [NRLG18]. L'annotation est réalisée en deux étapes : tout d'abord, nous effectuons une segmentation temporelle pour dissocier les phases montrant des émotions des phases de transitions. Puis, nous reconnaissons automatiquement ces segments par des techniques d'apprentissage automatique. La segmentation est réalisée à partir de caractéristiques dynamiques telles que la vitesse et l'accélération de la valeur des coefficients *blendshapes*. Pour l'étape de reconnaissance automatique, nous avons testé plusieurs algorithmes d'apprentissage automatique dont les *Support Vector Machines* (SVM), les techniques de plus proches voisins (KNN) et les *Random Forests*. Les résultats encourageants de ces travaux tendent à valider notre démarche.

Une autre contribution réside dans la méthode de synthèse d'animation que nous avons mise au point. Celle-ci est réalisée en deux étapes. La première étape, que nous appelons adaptation morphologique, est la suivante : nous appliquons une régression RBF aux positions de marqueurs originales afin d'obtenir les positions qu'auraient ces marqueurs s'ils étaient posés sur la surface de l'avatar. La seconde étape consiste à déterminer par optimisation la combinaison des coefficients de *blendshapes* minimisant la distance entre les positions de marqueurs adaptés et les sommets correspondants à chacun de ces marqueurs. Pour obtenir une combinaison optimale sans avoir d'artefacts nuisibles, il est nécessaire d'adjoindre à la fonction de coût des énergies de régularisation afin de pénaliser les combinaisons incorrectes. L'innovation consiste en l'adaptation que nous avons faite de l'énergie *thin-shell* (une énergie basée Laplacien quantifiant la déformation d'une surface) afin de l'adapter aux *blendshapes* et son incorporation comme énergie de régularisation à notre problème d'optimisation.

Enfin nos travaux préliminaires à la capture de données (en utilisant un dispositif de *motion capture* basé marqueurs) nous ont amené à une étude sur la disposition des marqueurs et la détermination d'un nombre suffisant pour une capture de qualité. Cette réflexion et la méthode que nous avons adoptée pour trouver des réponses ont fait l'objet d'une publication [RGL15b].

1.2 Organisation du document

Ce manuscrit est organisé en sept chapitres. Les trois premiers chapitres suivant cette introduction sont issus de travaux publiés. Les deux chapitres suivants présentent des travaux qui seront publiés après la soutenance.

Le chapitre 2 a été publié dans [RGL15a]. Il s'agit d'un état de l'art balayant les méthodes d'animation faciale basées données existantes en 2015. Il décrit les principales étapes permettant d'animer les visages de personnages virtuels à partir de données réelles. Les thèmes abordés sont les notions de bases du domaine de la modélisation et de l'animation $3D$, les différents dispositifs de capture de mouvement existants (*mocap* basée marqueurs, caméra rgbd basée stéréovision ou lumière structurée), les principales méthodes de l'état de l'art permettant d'animer des maillages 3D et les principales techniques permettant de calculer les paramètres d'animation à partir de données de mouvement réelles.

Le chapitre 3 a été publié dans [RGL15b]. Il s'agit d'un travail méthodologique préparatoire pour capturer des données via un dispositif de *mocap* basé marqueurs afin de répondre aux interrogations suivantes : i. combien de marqueurs sont suffisants et ii. comment les disposer. Une méthode de *clustering* (*Kmeans*) a été utilisée afin de faire varier facilement le nombre de marqueurs. Différentes dispositions de jeux de marqueurs proposées dans l'état de l'art ainsi que d'autres dispositions ad-hoc dérivées ont ainsi été testées. Les résultats de cette étude nous ont permis d'établir notre propre disposition pour la suite de nos travaux.

Le chapitre 4 a été publié dans [NRLG18]. Ce chapitre présente l'approche adoptée pour l'annotation automatique des expressions faciales affectives. Après une description du corpus de données capturé dans le cadre de nos travaux, nous présentons la méthode à proprement parler – segmentation, reconnaissance, puis annotation – que nous avons utilisée pour annoter et étiqueter automatiquement ces données.

Le chapitre 5 présente la méthode de synthèse basée *blendshapes* qui a été mise en oeuvre afin de générer des animations faciales à partir des données capturées : de l'adaptation morphologique à l'incorporation d'énergies de régularisation dans un système d'optimisation. Des résultats sont présentés pour différentes combinaisons de ces énergies.

Deux études perceptuelles présentées au chapitre 6 analysent les résultats obtenus par notre système de synthèse puis le comparent à l'état de l'art, ce qui nous a permis de valider le modèle de synthèse réalisé.

Enfin, nous concluons sur un bilan des différents apports de cette thèse et sur les perspectives qu'elle ouvre pour des travaux futurs.

CHAPITRE 2

ANIMATION FACIALE BASÉE DONNÉES : NOTIONS DE BASE

Ce chapitre, ayant fait l'objet d'une publication aux Journées Françaises d'Informatique Graphique [RGL15a] en 2015, dresse un panorama des différentes problématiques liées à l'animation faciale basée données. Nous y avons apporté quelques modifications afin d'y intégrer les avancées récentes du domaine.

Le but de l'animation basée données est d'animer des personnages virtuels reproduisant les actions effectuées par des acteurs humains. Dans ce contexte, le visage joue un rôle prépondérant puisqu'il est l'un des principaux vecteurs de l'émotion et de la communication chez l'humain. Par ailleurs, contrairement au reste du corps dont les mouvements sont contraints par des articulations et des os, les déformations du visage suivent une autre forme de dynamique ce qui en fait un cas d'étude à part. Les applications sont diverses, par exemple, la création d'avatars virtuels ou l'animation de personnages présentant un comportement naturel. Dans ce papier, nous aborderons la question depuis la capture des données faciales (les différents dispositifs et méthodes de capture) jusqu'aux méthodes de synthèse exploitant ces données.

2.1 Introduction

L'animation faciale basée données est un sujet qui a gagné en intérêt au cours des dernières années. Il s'agit d'animer des avatars virtuels à partir de données réelles. Les enjeux sont divers. Les industries audiovisuelles et vidéoludiques notamment sont demandeuses de ce type de technologies car elles permettent de générer des animations d'autant plus crédibles aux yeux du public qu'elles sont produites à partir de comportements humains réels. Parmi les applications possibles on peut citer la communication en temps réel d'humains à humains anonymisée par le biais de ces avatars, ou bien encore la création d'agents virtuels présentant un comportement humain pour des interfaces humains-machine.

Les quinze dernières années ont vu se développer successivement deux principaux types de

technologies de capture de données $3D$: (i) La capture de mouvement (*mocap*) qui consiste à suivre par triangulation les positions $3D$ d'un nombre limité de marqueurs disposés sur le corps d'un acteur (et, dans le cas qui nous intéresse, le visage) ; (ii) et plus récemment, les technologies de types *RGB-D* (ou caméras à capteurs de profondeur) reposant sur la vision binoculaire ou l'émission de lumière structurée.

Parmi les méthodes d'animation faciale, celles basées *blendshapes* restent très populaires de part leur facilité d'utilisation. Elles s'appuient sur des formes de base qui étaient initialement définies par des graphistes. Aujourd'hui, comme nous le verrons dans ce chapitre, des méthodes permettent d'automatiser cette étape. Plus récemment, d'autres méthodes (notamment les modèles à couches fines) ont vu le jour et se sont développées. Elles permettent d'appliquer au maillage des déformations avec davantage de degrés de libertés que ceux permis par les modèles de type *blendshapes*. Quel que soit le choix du type de modèle, le lien entre les données sources et les modèles est essentiellement assuré par des techniques d'optimisation.

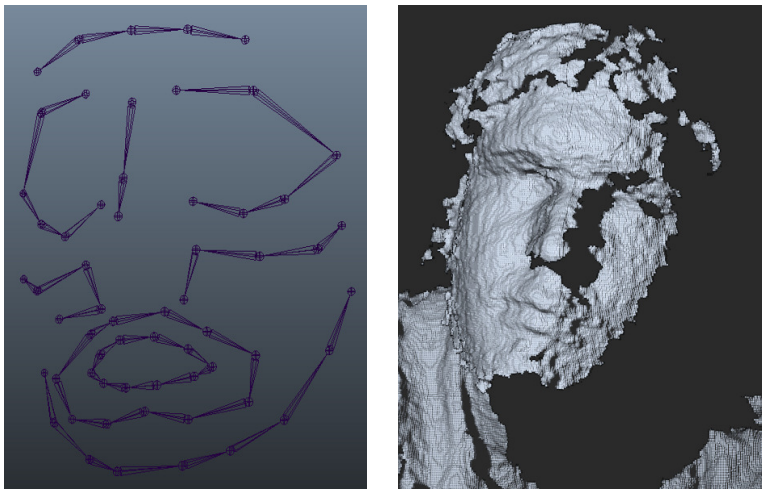
En dehors des modèles strictement géométriques, des méthodes dites physiques (e.g., reposant sur des systèmes masse-ressort) ainsi que des techniques (aussi bien géométriques que physiques) ayant vocation à représenter des modèles anatomiques [SNF05, HWHM15, LTW95] ont aussi été développées. Néanmoins, ce type de méthode n'est pas le plus utilisé.

En dépit du fait que l'animation faciale basée données ait gagné en maturité notamment grâce à ces dernières innovations, il n'existe que très peu d'états de l'art traitant spécifiquement de ce sujet. À notre connaissance, l'étude la plus complète et pertinente est proposée par [DN08a]. Néanmoins, les dernières avancées technologiques (en particulier les dispositifs de type *RGB-D*) et les possibilités qu'elles offrent n'y sont pas évoquées.

Cet article vise à fournir une introduction à l'animation faciale basée données ainsi qu'une source de références vers les dernières tendances du domaine. Dans un premier temps nous présenterons les principaux dispositifs de capture de type *mocap* et *RGB-D*, les principes généraux sur lesquels ils reposent et leurs principaux problèmes (section 2.2). Dans un second temps, les principales méthodes d'animation à partir de données sources capturées sont présentées (section 2.3). Enfin une discussion concernant les avantages et inconvénients liés à ces différentes techniques est proposée (section 2.4).

2.2 Acquisition des données sources

Dans cette section, nous discuterons de deux principales méthodes de capture (figure 2.1) : la capture de mouvement basée marqueurs (*mocap*) et la capture sans marqueur via des dispositifs



(1) (2)

FIGURE 2.1 – Représentation des données capturées via (1) *mocap* ou (2) dispositif *RGB-D* (nuage de points uniquement).

de type *RGB-D*.

2.2.1 Capture de mouvement $3D$ basée marqueurs

La *mocap* est une technique couramment utilisée (tant pour la capture du visage que du corps de l'acteur) par l'industrie audiovisuelle. Il s'agit de suivre les déplacements de marqueurs placés sur le corps d'un acteur à l'aide d'un ensemble de caméras ; chaque marqueur doit être visible par au moins deux caméras pour que sa position exacte puisse être connue à un instant donné. Cette technologie permet de capturer le mouvement de ces marqueurs avec une précision spatiale et une fréquence temporelle élevées (respectivement ≤ 1 mm et ≥ 120 Hz). Néanmoins la résolution spatiale de ce type de capture est faible dans la mesure où le nombre de marqueurs que l'on peut disposer sur le visage de l'acteur est limité. Les données obtenues sont donc décrites par un nuage de K points évoluant au cours du temps.

En général, pour la capture de mouvement non-faciale, le positionnement des marqueurs est choisi afin de pouvoir inférer les paramètres de rotation et les positions des différentes articulations du squelette selon des modèles biomécaniques. En revanche, en ce qui concerne le visage, le choix du nombre de marqueurs et surtout leurs positionnements n'ont à ce jour fait l'objet que de très peu d'études [LZD13, RGL15b]. Dans les faits, le nombre de marqueurs varie globalement d'une trentaine de marqueurs (disposition *Face Robot* du logiciel commercial Softimage d'Autodesk) à une centaine [DCFN06b, HCTW11]. Augmenter le nombre de

marqueurs aura pour effet d'augmenter la quantité d'information spatiale capturée concernant la déformation du visage de l'acteur à chaque *frame* et donc de rendre plus efficace et plus fidèle la synthèse des expressions faciales. Néanmoins cela a un coût, que ce soit en ce qui concerne le post-traitement des données (e.g., inversion de marqueurs dans le suivi des trajectoires) ou la préparation des séances de capture (e.g., risque d'oubli / de chute / mauvais placement d'un marqueur). Il faut également tenir compte du fait que l'information n'est pas uniformément répartie sur l'ensemble du visage et que celle portée par certains points du visage peut-être redondante avec celle portée par d'autres points ; une répartition intelligente des marqueurs est donc susceptible d'améliorer significativement la qualité des données capturées. À cela s'ajoute le besoin réel de ces informations, par exemple, si le modèle d'animation ciblé est relativement simple, s'il n'a qu'un nombre de degrés de liberté limité comme c'est le cas pour un modèle basé *blendshapes* (voir section : 2.3) où l'espace de représentation est celui des combinaisons linéaires de ses bases (une cinquantaine dans le cas où ses bases sont calquées sur le FACS), alors un nombre restreint de marqueurs bien placés peut suffire.

Par ailleurs, si la *mocap* permet de capturer efficacement les déformations à large échelle liées aux expressions faciales et aux mouvements labiaux, du fait de sa faible résolution spatiale, elle échoue à capturer à elle-seule les déformations plus fines telles que celles liées aux rides. Il est néanmoins possible de contourner ce problème en faisant appel à des solutions hybrides. Dans [BBA$^+$07b] les auteurs ont, d'une part, incorporé à leur modèle facial cible un modèle permettant de représenter les déformations liées aux rides et, d'autre part, ajouté à leur dispositif de capture des caméras haute-résolution afin de détecter les rides et en inférer les paramètres pour leur modèle. Dans [HCTW11], les détails fins sont aussi pris en charge par le modèle (basé *blendshapes*) dont les bases ont été générées à partir de captures haute résolution effectuées via un scanner laser.

2.2.2 Capture de mouvement $3D$ sans marqueur type *RGB-D*

Nous parlerons ici des méthodes de capture $3D$ sans marqueur qui reposent principalement sur deux technologies, la lumière structurée et la stéréovision. Les enjeux de ce type de capture sont d'une part de capturer un nuage de points autour de la surface du visage, d'autre part d'acquérir un maillage $3D$ représentant le visage de l'acteur avec une haute résolution à partir de ce nuage de points à un instant donné, et enfin de suivre les déformations de ce maillage au cours du temps de façon à préserver une structure (ensemble de sommets + liens de connexité) commune au cours du temps.

Obtention d'un nuage dense de points

La lumière structurée [RHHL02][ZH04] consiste à projeter un motif lumineux ayant des caractéristiques connues, l'analyse de la déformation de ce motif sur la surface des objets de la scène permet d'obtenir des informations sur la géométrie de ces objets. La figure 2.2 illustre un exemple de ce type de dispositif. Un état de l'art sur les différents motifs pouvant être utilisés a été réalisé par Salvie et al. [SFPL10], Zhang Song [Zha10] a réalisé un état de l'art spécifique aux motifs de franges (e.g., sinusoïdes).

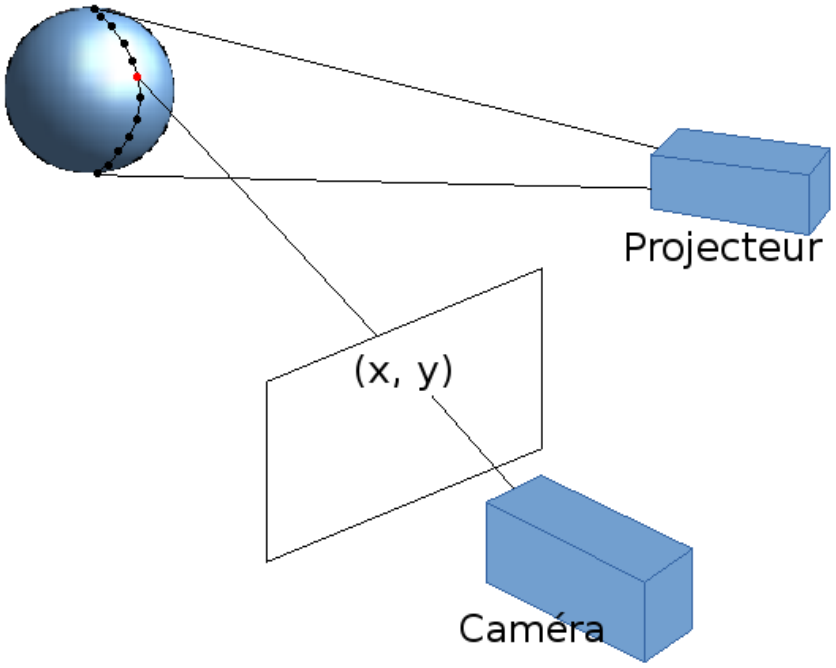


FIGURE 2.2 – Illustration de l'estimation de la position d'un point dans l'espace par projection d'une lumière structurée (ici le motif est une bande). La position du point sera déterminée par l'intersection entre la droite partant de la caméra et le plan de la bande de lumière projetée.

La stéréovision consiste à capturer la scène sous plusieurs angles de vue simultanément. La figure 2.3 illustre ce type de dispositif, sachant qu'un couple de pixels P_1 et P_2 sont les projections d'un même point X sur deux caméras différentes dont on connait les centres de projections O_1 et O_2. La position $3D$ du point X est à l'intersection des droites O_1P_1 et O_2P_2. Pour un pixel P_1 donné, l'intégralité des pixels P_2 pouvant être un projeté de X sont situés sur la droite épipolaire, il s'agit alors de trouver le pixel le plus similaire parmi ces derniers. Un exemple de mesure de similarité pouvant être employée pour évaluer la correspondance possible entre deux pixels est la corrélation croisée des pixels P_1 et P_2 sur leurs voisinages. Scharstein

et Szeliski [SS02] ont réalisé une taxonomie complète des différentes stratégies pour la mise en correspondance de couples d'images.

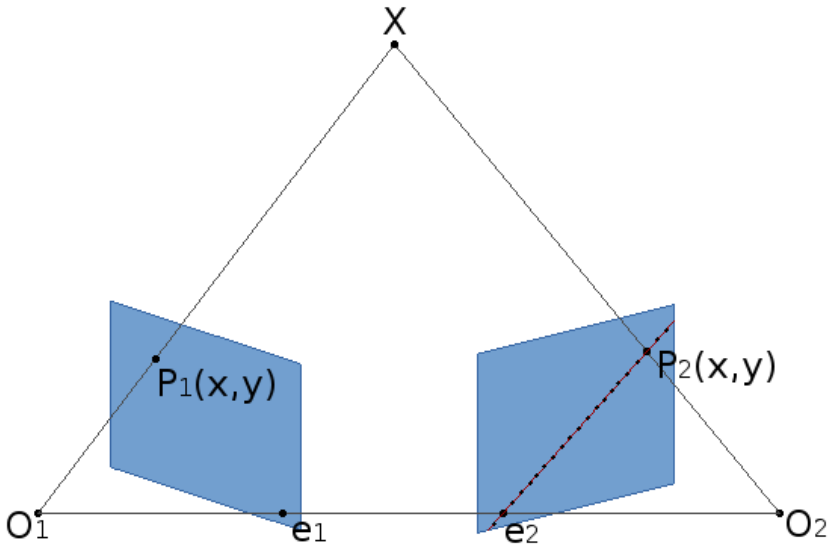


FIGURE 2.3 – Illustration de l'estimation de la distance d'un point par vision binoculaire. O_1 et O_2 sont les centres de projection des deux caméras, e_1 et e_2 sont les intersections entre la droite O_1O_2 et les plans de projection de chacune des caméras. Le point P_1 correspond à la projection du point X sur le plan de la caméra 1. La droite e_2P_2 (appelée ligne épipolaire) est l'intersection entre le plan XO_1O_2 et le plan de projection de la caméra 2. Le point P_2 est le pixel le plus similaire à P_1 situé sur la ligne épipolaire. Les droites O_1P_1 et O_2P_2 sont situées sur un plan commun et s'intersectent au point X.

Les techniques basées sur la projection de lumière structurée ont longtemps permis d'obtenir des résultats plus robustes (l'éclairage permet d'injecter de l'information connue) que les méthodes basées strictement sur la stéréovision. Néanmoins les travaux récents de Bradley et al. en 2008 [BBH08] puis de Beeler et al. [BBB+10] ont montré qu'il est possible d'obtenir des résultats comparables avec des dispositifs de capture totalement passifs (c.à.d. sans émission de lumière). L'un des points forts de [BBB+10] est sa robustesse ; le système est conçu selon un mode pyramidal, la mise en correspondance des pixels est réalisée de façon progressive, d'abord à basse résolution puis en l'augmentant à chaque niveau, des contraintes de cohérence sont utilisées à chacun de ces étages ainsi qu'un système de raffinement progressif.

D'autres approches hybrides ont été proposées. Dans Zhang et al. [ZSCS04] la projection de motifs sert essentiellement à ajouter des variations de couleurs sur le visage et donc faciliter l'évaluation de la correspondance entre les pixels de chacune des caméras. Weise et al. [WLVG07] ont proposé une méthode d'acquisition tirant parti à la fois de la projection de mo-

tifs lumineux et de la stéréovision.

Post-traitements

L'inférence des coordonnées $3D$ associées à chaque pixel peut être sujet à des erreurs. Un certain nombre de traitements peuvent être effectués afin d'améliorer la qualité des résultats, réduire la redondance des informations, filtrer les erreurs aberrantes.

Aligner différents nuages de points est une tâche qui peut s'avérer nécessaire. Pour une modélisation complète du visage, il est utile de pouvoir le capturer sous plusieurs angles de vue. Soit en ayant plusieurs dispositifs capturant simultanément la scène sous différents angles de vue [BBB+10], soit en ayant un appareil unique capturant le visage sous différentes poses [RHHL02][WBLP11a]. Les différences (angle et translation) entre chacune de ces prises de vue ou poses ne sont pas forcément connues. L'algorithme le plus fréquemment utilisé pour retrouver ces paramètres de transformation rigide est l'ICP (*Iterative Closest Points*). Brièvement, il s'agit d'un algorithme itératif visant à aligner un nuage de point de référence avec un autre nuage de points en estimant successivement pour chaque point du nuage de référence le plus proche voisin dans l'autre nuage connaissant une estimation des paramètres de transformation, puis à réestimer ces paramètres afin d'aligner ces couples jusqu'à convergence. Dans [RL01], les auteurs proposent différentes variantes de cet algorithme.

Le filtrage des anomalies est une tâche qui revient régulièrement dans la littérature. Suivant le dispositif de capture et la méthode de détermination de la profondeur choisie, la génération de points aberrants peut avoir des origines différentes. Par exemple avec une méthode par décalage de phase dans un motif sinusoïdal projeté, on peut avoir des erreurs générées par l'ambiguïté liée à la périodicité [WLVG07] [ZH04]. Dans le cas d'une reconstitution par stéréovision, les erreurs peuvent être dues à un calcul de disparité erronné à cause d'une mauvaise mise en correspondance de pixels.

Avant même de filtrer le nuage de points, le nombre de points aberrants peut être fortement réduit par l'ajout de contraintes lors de sa génération assurant ainsi sa cohérence.

Les anomalies restantes doivent être détectées, pour cela il convient de définir ce qui caractérise un point aberrant et définir des contraintes afin de les discriminer. Dans [WLVG08a] et [MAW+07], les auteurs tirent parti des informations collectées depuis les différents points de vue afin de détecter des incohérences. Par exemple une erreur survient lorsqu'une caméra détecte un point qui devrait normalement avoir été masqué par un autre point détecté par une autre caméra. Dans [BBH08] un critère basé sur la distance par rapport au plan approximant ses voisins (estimable par la méthode des moindres carrés) est utilisé afin de détecter les anomalies

(dépassant un certain seuil), d'autres critères sont proposés dans [WPK+04].

Sous-échantillonner le nuage points peut parfois être utile, afin de réduire la redondance d'informations [BBH08] ou pour accélérer les temps de traitement (pour permettre un rendu temps réel par exemple) [RHHL02]. Dans le premier cas une structure arborescente tirée de [BHGS06] a été utilisée, dans le second cas les points redondants ont été fusionnés via une grille de voxels.

2.2.3 Quelques bases de données existantes

La grande majorité des bases de données disponibles ont été capturées via des caméras $2D$ classiques [LCK+10, KCT00] [APD10a] ou via des caméras rgb+d [CWZ+14] [ZYC+13]. Les base de données capturées via des dispositifs de *mocap* basée marqueurs sont en revanche plus rares (dispositifs plus coûteux et compliqués à mettre en place). À notre connaissance le seul existant est [BBL+08]. Dans sa thèse [Web17] dresse un excellent panorama des bases de données existantes.

2.3 Animation

Dans cette section, nous explicitons le formalisme général utilisé dans l'article puis nous décrivons les principales méthodes utilisées dans la littérature.

2.3.1 Formalismes

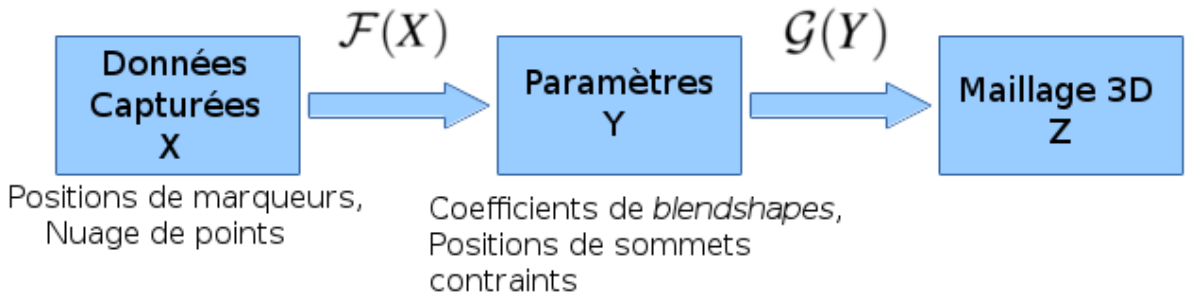


FIGURE 2.4 – Des données sources au maillage final.

Le moyen le plus commun de représenter le visage comme la plupart des objets $3D$ est sous la forme d'un maillage $3D$ de polygones (souvent triangulaires). L'ensemble des sommets et

leurs liens de connexité forment la structure géométrique du maillage. En règle générale, cette structure reste constante, en revanche les coordonnées de chaque sommet peuvent être modifiées afin de déformer le maillage à des fins d'animation.

Le but de l'animation basée données est de produire une séquence $Z = (Z_1, \cdots, Z_n, \cdots, Z_N)$ de N "poses", où chaque pose $Z_n = (Z_n^1, \cdots, Z_n^v, \cdots, Z_n^V)$ est le vecteur de dimension $3 \times V$ des positions $3D$ des sommets du maillage de la $n^{ième}$ pose.

Ces positions peuvent être déterminées à partir de paramètres représentés par le vecteur $Y = (Y_1, \cdots, Y_n, \cdots, Y_N)$ où chaque vecteur $Y_n = (Y_n^1, \cdots, Y_n^p, \cdots, Y_n^P)$ de dimension P est le vecteur des paramètres du modèle à la $n^{ième}$ pose. La relation entre Y et Z est donnée par le modèle représenté par la fonction :

$$\mathcal{G}(Y) = Z : \mathbb{R}^P \mapsto \mathbb{R}^V \tag{2.1}$$

Dans le cadre de l'animation basée données, on cherche à déterminer automatiquement les paramètres Y à partir des données $X = (X_1, \cdots, X_n, \cdots, X_N)$ avec $X_n = (X_n^1, \cdots, X_n^d, \cdots, X_n^D)$ le vecteur de dimension D des données capturées à la $n^{ième}$ pose. La fonction recherchée est :

$$\mathcal{F}(X) = Y : \mathbb{R}^D \mapsto \mathbb{R}^P \tag{2.2}$$

Donc par composition on peut relier les données "sources" X aux données "cible" Z (figure 2.4) :

$$(\mathcal{G} \circ \mathcal{F})(X) = Z : \mathbb{R}^D \mapsto \mathbb{R}^V \tag{2.3}$$

2.3.2 Création d'un maillage $3D$

Il est possible, si l'on dispose d'un nuage suffisamment dense de points représentant la surface de l'objet de générer un maillage $3D$.

Ce maillage peut être construit directement. Bradley et al. en 2008 [BBH08] ont utilisé une triangulation de Delaunay. Néanmoins, cette méthode n'étant efficace que sur de petits ensembles de points, une classification non supervisée (*clustering*) a dû au préalable être effectuée suivant un octree dirigé par la densité ; la triangulation a ensuite été effectuée localement sur chaque cluster. [BBB$^+$10] ont utilisé une méthode (*Poisson Surface Reconstruction*) proposée en 2006 par Kashdan et al. [KBH06].

Une autre solution consiste à déformer un maillage générique existant afin de coller au nuage de points. Classiquement, le problème est posé comme une optimisation visant à minimiser

un terme d'ajustement des sommets du maillage par rapport au nuage de points et un terme de régularisation visant à éviter des déformations incohérentes. C'est le choix qui a été fait notamment dans [ZSCS04] mais aussi dans [WLVGP09] .

L'usage de scanner laser est assez fréquent. En demandant à un acteur de conserver une expression fixe le temps d'un scan, il est possible d'obtenir un modèle fidèle de son visage. Si cette méthode ne permet d'obtenir qu'une modélisation statique du visage de l'acteur, le maillage obtenu peut ensuite être déformé à partir de données capturées.

Le *Morphable Model* proposé dans [BV99] est fréquemment utilisé afin de générer des maillages 3D. Il s'agit d'un système basé Analyse en Composante Principale (ACP) avec une segmentation éventuelle des différentes parties du visage. Une ACP est calculée sur une base de données de maillages $3D$ correspondant aux visages de personnes acquis via un scan $3D$. Les paramètres du modèle sont les coordonnées du maillage $3D$ dans le sous-espace défini par l'ACP. De nouveaux maillages peuvent être générés par optimisation de ces coefficients en minimisant l'écart entre la projection du maillage généré dans l'espace des données (X) et les données effectivement capturées.

2.3.3 Animation à base de formes clés (*blendshapes*)

Dans le cadre de l'animation basée *blendshapes*, la fonction $\mathcal{G}$ est une combinaison linéaire de P formes de bases, chacune pondérée par un coefficient w_p :

$$\mathcal{G}(Y) \equiv b_0 + \sum_{p=1}^{P} w_p b_p \tag{2.4}$$

avec b_0 représentant l'expression neutre (c.à.d. les coordonnées initiales des sommets) et b_p la $p^{ième}$ base du modèle. Avec une notation matricielle, on aura donc :

$$\mathcal{G}(Y) \equiv WB \tag{2.5}$$

avec W le vecteur de dimension P des coefficients w_p (avec $w_0 = 1$) et B la matrice de dimension $(P \times V)$ des sommets des bases b_p.

Les paramètres (Y) du modèle sont donc d'une part l'ensemble des bases b_p et d'autre part les coefficients w_p associés. Néanmoins, les bases sont en général déterminées une fois pour toute et restent inchangées par la suite. Les seuls paramètres évoluant au cours du temps sont donc les coefficients w_p.

Ce type de modèle présente deux principaux avantages. D'une part il offre un niveau d'abs-

traction supplémentaire facilitant le transfert d'animations d'un modèle à l'autre. En effet, pour deux maillages cibles différents, même topologiquement, si les bases associées à chacun d'entre eux représentent des expressions identiques, alors le transfert des animations peut se faire directement en prenant le même ensemble de coefficients. D'autre part, il s'agit d'un modèle linéaire relativement simple, ce qui permet des temps de calcul compatibles avec des applications en temps réel [WBLP11a].

Les deux principales problématiques liées aux *blendshapes* seront traitées dans les sous-sections suivantes.

Choix et génération des bases

Le choix des bases b_p doit répondre à plusieurs problématiques. En premier lieu, elles doivent être aussi indépendantes que possible les unes des autres de façon à faciliter l'éventuelle édition [LD08] ultérieure par des animateurs et ainsi éviter que lors du processus d'optimisation des coefficients w_p certains prennent des valeurs inférieures à 0 ou supérieurs à 1 pour compenser des déformations liées à d'autres bases (en général, les bases sont construites de telle sorte que le coefficient qui lui est associé doit être compris entre 0 et 1). En second lieu, l'espace de représentation défini par ces bases doit couvrir autant que possible celui de l'expressivité humaine. Enfin, il est préférable que les déformations liées à chacune de ces bases aient une signification particulière (e.g., sourire, fermeture d'un oeil, ouverture de la bouche, etc) afin d'être plus facilement manipulables par un animateur.

Le Facial Action Coding System (FACS) est un choix de bases assez fréquent. Originellement proposé par Ekman en 1978 [EF78] afin de décrire les expressions faciales humaines, il propose une cinquantaine d'Action Unitaires (AU) correspondant chacune à l'activation d'un muscle ou d'un groupe de muscles. Créer une base correspondante à chacune de ces AU [WBLP11a, CWZ+14] est un choix qui présente l'avantage d'être assez réaliste tout en facilitant l'édition.

Dans [HCTW11][XCLT14], les bases sont sélectionnées parmi les *frames* d'une séquence capturée via *mocap* ou d'une animation existante, l'idée étant de sélectionner un nombre minimal de *frames* à partir desquelles il est possible de minimiser sur l'intégralité des *frames* de la séquence l'erreur de reconstruction par rapport aux données capturées.

L'Analyse en Composantes Principales (ACP) est une méthode parfois utilisée pour créer les bases du modèle basé *blendshapes* [CB02, WLVGP09, LD08]. Elle vise à déterminer un ensemble de P vecteurs orthogonaux (appelés composantes principales) m_p, tels que la variance des projections des observations dans le sous-espace défini par les vecteurs m_p est maximisé.

Ainsi toute observation m peut être représentée sous la forme :

$$m = m_0 + \sum_{p=1}^{P} \alpha_p \times m_p \tag{2.6}$$

avec α_p les coordonnées dans l'espace de l'ACP et m_0 le vecteur moyen des observations. Cette représentation est assez proche du modèle de *blendshapes* puisque chaque observation est représentée par une combinaison linéaire de vecteurs. Dans [CB02] les auteurs ont choisi pour bases les observations les mieux représentées sur chacun des axes de l'ACP. Dans [LD08], une méthode de création et manipulation de *blendshapes* basée ACP permettant de faciliter l'édition est proposée.

Lorsqu'elles ne sont pas obtenues directement par ACP, les bases peuvent être créées manuellement ou générées à partir d'expressions humaines réelles capturées via un dispositif de capture $3D$ [CBK+06, HCTW11, WBLP11a, CWZ+14]. On ne dispose pas toujours d'une capture par base, par exemple, dans le cas du FACS la plupart des humains ne parviennent pas à contrôler chacune des AU indépendamment. La méthode proposée dans [LWP10a] permet d'inférer les bases à partir d'une série d'exemples d'expressions faciales quelconques pour lesquelles les degrés d'activation w_p de chacune des bases sont connus. Cette méthode a notamment été exploitée dans [WBLP11a] et [CWZ+14]. La méthode de transfert de déformations de [SP04] (section 2.3.6) a été utilisée dans [XCLT14] afin de générer un jeu de bases associées à un maillage cible à partir d'un jeu de bases associées à un premier maillage source. Dans [WLVGP09], les bases sont déterminées linéairement par résolution au sens des moindres carrés à partir d'un ensemble d'expressions pour lesquelles les déformations du maillage et les coefficients de *blendshapes* étaient connus.

Détermination des coefficients

Étant donné un maillage et un ensemble de bases du modèle *blendshapes* associé, l'obtention des coefficients w_p à partir de données sources X revient à un problème d'optimisation. Il s'agit de minimiser une énergie E_{match} telle que :

$$\mathcal{F}(X) = Y^* = arg \min_{Y} E_{match} \tag{2.7}$$

Le plus souvent le problème est formulé au sens des moindres carrés :

$$E_{match} = \sum_{d=0}^{D} \|WB(v_d) - x_d\|^2 \tag{2.8}$$

avec v_d, le sommet associé à la donnée capturée x_d et $B(v_d)$ le vecteur de dimension P des déformations de chaque base appliquée au vertex v_d correspondant à la donnée capturée $x_d \in X$. Il s'agit d'un problème d'optimisation linéaire pouvant être résolu de façon analytique.

Formaliser le problème au sens des moindres carrés n'est pas la seule option. Par exemple le problème peut être abordé avec une approche bayésienne [WBLP11a]. Dans ce cas le problème est formulé comme une MAP (maximisation de l'estimation *a posteriori* [Her04]) :

$$p(Y|X) = \frac{p(X|Y) \times p(Y)}{p(X)} \tag{2.9}$$

avec $p(Y|X)$ la probabilité a posteriori, $p(X|Y)$ la vraisemblance des données X connaissant les paramètres Y et $p(Y)$ la fonction de distribution *a priori* des paramètres Y. La probabilité $p(X)$ étant indépendante de Y, on peut se contenter de chercher :

$$Y^* = arg\max_{Y} p(X|Y) \times p(Y) \tag{2.10}$$

Cela peut se ramener à un problème de minimisation d'énergie tout en présentant l'avantage de permettre une interprétation bayésienne :

$$Y^* = arg\min_{Y} -ln(p(X|Y)) - ln(p(Y)) \tag{2.11}$$

D'une manière générale, le problème n'est pas toujours posé de façon linéaire et doit être résolu par des méthodes d'optimisation non-linéaires itératives (par exemple : descente de gradient, Gauss-Newton, gradient conjugué, etc).

Un problème qui peut se poser vient du fait que l'optimisation posée en l'état ne tient compte ni de la cohérence temporelle des coefficients trouvés (les coefficients doivent être cohérents les uns par rapport aux autres au cours des *frames* successives) ni de la cohérence spatiale (les coefficients doivent rester autant que possible entre 0 et 1). En effet, il n'est pas rare d'obtenir des coefficients très grands compensés par d'autres coefficients inférieurs à 0 [CB02]. Une première méthode permettant d'éviter cet effet indésirable est d'ajouter une contrainte de non-négativité dans le processus d'optimisation. Ainsi [XCLT14, HCTW11, CB02] ont utilisé le *Non-Negative Least Square Solver* (NNLS) proposé par [LH74]. On peut aussi adjoindre une

seconde énergie E_{reg} de régularisation à minimiser, l'équation (2.7) devient alors :

$$\mathcal{F}(X) = Y^* = arg\min_{Y}(E_{match} + \alpha E_{reg}) \tag{2.12}$$

avec α un paramètre permettant de pondérer le degré de régularisation et E_{reg} une énergie pénalisant les valeurs de Y improbables. Dans [SLS+12b, CWLZ13, WBLP11a], cette énergie de pénalisation est assimilée à une fonction de densité de probabilité a priori $p(Y)$ dont les paramètres ont déjà été calculés sur des données d'entraînement.

Deng et al. ont proposé en 2006 [DCFN06b] une approche consistant à entraîner à partir de L couples d'entraînement (X_l, W_l) un régresseur RBF (*Radial Basis Function network*) inférant les coefficients w_i de données sources quelconques X. Il s'agit alors de construire la fonction $\mathcal{F}(X)$ sous cette forme :

$$Y* = \mathcal{F}(X) = \sum_{l=0}^{L} c_l \phi_l(X) \tag{2.13}$$

avec c_l les paramètres de $\mathcal{F}(X)$ que l'on va chercher à estimer et ϕ_l une fonction noyau (par exemple gaussien $\phi_l(X) = exp(-\|X - X_l\|^2/2\sigma_l^2)$), les paramètres $C = (c_0, c_1, ..., c_L)$ seront optimisés selon le principe des moindres carrés à partir des exemples d'entraînement :

$$C^* = arg\min_{C} \sum_{l=0}^{L}(w_l - \mathcal{F}(X_l))^2 + \alpha \sum_{l=0}^{L} c_l^2 \tag{2.14}$$

$\alpha \sum_{l=0}^{L} c_l^2$ étant un terme de régularisation pénalisant les paramètres trop importants. Néanmoins pour appliquer cette méthode, il est nécessaire de se procurer au préalable les couples (X_l, W_l).

2.3.4 Modèles à couches fines

Le but de ce type de méthode consiste à trouver à partir d'un sous-ensemble de sommets d'un maillage les coordonnées $3D$ de chacun de ses autres sommets en minimisant la déformation de la surface initiale. La déformation est décrite par deux termes, l'énergie d'étirement et l'énergie de torsion. Le problème est détaillé dans [BS08], brièvement il est montré que minimiser cette énergie revient à résoudre l'équation aux dérivées partielles d'Euler-Lagrange suivante :

$$-k_s \Delta d + k_b \Delta^2 d = 0 \tag{2.15}$$

avec k_s, k_b des paramètres permettant de contrôler la raideur de la surface, d la déformation de la surface par rapport à la position au repos et Δ et Δ^2 respectivement l'opérateur laplacien

et bilaplacien ($\Delta^2 d = \Delta(\Delta d)$). D'un point de vue numérique, l'opérateur laplacien doit être discrétisé afin de s'appliquer à un maillage triangulaire. Cette discrétisation revient à calculer le laplacien $\mathcal{L}$ du maillage : $\mathcal{L} = \mathcal{M}^{-1}\mathcal{L}^*$ avec $\mathcal{M}$ diagonale et $\mathcal{L}^*$ symétrique les matrices de normalisation associées respectivement à chaque sommet et à chaque arête (en général calculés selon la discrétisation par cotangeantes [MDSB03]). Si on reprend l'équation (2.15), en prémultipliant par $\mathcal{M}$ on obtient :

$$(-k_s\mathcal{L}^* + k_b\mathcal{L}^*\mathcal{M}^{-1}\mathcal{L}^*)d = 0 \tag{2.16}$$

En passant les colonnes correspondantes aux sommets dont les positions sont contraintes à droite de l'équation puis en supprimant les lignes correspondantes, on obtient :

$$ad^* = b \tag{2.17}$$

où d^* est le vecteur des déplacements que l'on souhaite calculer, a, b ne dépendent que du laplacien du maillage et des positions connues de sommets et d^* est le vecteur des positions de sommets inconnues.

Si l'on se rapporte au formalisme proposé précédemment, les paramètres du modèle $\mathcal{G}(Y) \equiv d^* = (a^T a)^{-1} b$ (a^T étant la transposée de a) sont d'une part les matrices $\mathcal{M}$ et $\mathcal{L}^*$ calculées en une seule fois sur le maillage initial, et d'autre part, les positions des sommets contraints évoluant au cours du temps. Si les sommets contraints demeurent toujours les mêmes, alors il suffit de calculer $(a^T a)^{-1}$ une seule fois, le reste du calcul pouvant quant à lui s'effectuer en des temps compatibles avec du temps réel. Si la liste des sommets contraints change, alors la structure de a est changée et il est nécessaire de recalculer l'inversion de $(a^T a)$ via une décomposition de Cholesky. Lorsque les données sont capturées via *mocap* et que le maillage est similaire au visage de l'acteur, on peut associer à chaque marqueur le sommet qui lui correspond sur le maillage, ces sommets correspondront aux sommets contraints dont les positions sont déterminées par les positions des marqueurs.

Le modèle à couches fines a notamment été employé dans [BBA+07b] et [LZD13] afin d'animer directement les maillages $3D$ à partir des positions des marqueurs *mocap*. [WLVGP09] s'est également servi de ce type de modèle comme énergie de régularisation dans son processus d'optimisation. [LYYB13] utilise également un modèle appartenant à la même famille (lapalcien discret).

2.3.5 Flux optiques

Lorsque l'on travaille avec des dispositifs de type *mocap* sans marqueurs (stéréovision, lumière structurée), il est possible d'utiliser des méthodes de flux optique afin de propager les déformations d'un maillage de référence au cours du temps.

Dans [BHPS10a] , un maillage initial G_t est d'abord calculé pour chaque *frame*. Ces différents maillages ne partagant pas une structure commune, le maillage de référence G'_t, calculé initialement comme étant G_0, est déformé à chaque *frame* t afin de s'approcher au plus près du maillage G_t puis repris afin de calculer G'_{t+1}. À chaque instant t et pour chaque sommet v du maillage G'_t, le flux du pixel $pi_{t,v,c}$ correspondant à la projection de ce sommet sur l'image de chaque caméra c est calculé afin d'obtenir une nouvelle position de ce pixel $pi_{t+1,v,c}$ à l'instant suivant. Cette nouvelle position est ensuite reprojetée sur la mesh G_{t+1} afin d'obtenir la position du sommet v du maillage G'_{t+1} à l'instant suivant.

Dans [ZPS04], à chaque *frame* un maillage générique est déformé afin de correspondre au visage de l'acteur. Cette déformation est calculée en minimisant une énergie $E = E_s + \alpha E_r + \beta E_m$, où E_s est une énergie de fitting liée à la distance entre chaque sommet du maillage générique et la carte de profondeur ; E_r est une énergie de régularisation visant à limiter les grands déplacements entre sommets voisins (les sommets voisins doivent présenter des déformations similaires), enfin, E_m est une énergie limitant la différence entre le déplacement du pixel $pi_{c,v,t}$ au pixel $pi_{c,v,t+1}$ (projections respectives du sommet v à l'instant t et $t+1$ sur la caméra c) et le déplacement du pixel $pi_{c,v,t}$ au pixel $pi'_{c,t+1}$ calculé par flux optique.

L'un des problèmes principaux lié à l'emploi de flux optique concerne la dérive numérique qui est liée à l'accumulation successive des erreurs au cours du temps. [BHB+11a] propose de résoudre ce problème en "ancrant" les déformations temporelles successives sur un ensemble de poses de référence. Cette méthode est reprise dans [FJA+15].

2.3.6 Transfert d'expressions et de détails fins

Le but du transfert d'expressions consiste à transférer les déformations appliquées à un maillage $3D$ source vers un autre maillage cible dont les proportions et la structure peuvent être différentes. Le premier papier à introduire ce terme est [NN01], qui propose une mise en correspondance des deux maillages par interpolation RBF. Une autre méthode fréquemment utilisée est celle proposée par [SP04] visant à transférer les déformations affines appliquées aux triangles du maillage source vers ceux du maillage cible tout en conservant la cohérence de ce dernier [WLVGP09, XCLT14, CWLZ13].

Les interpolations RBF sont fréquemment utilisées pour trouver les positions correspondantes sur une surface donnée de points situés sur une autre surface aux proportions différentes [BBA+07b, SLS+12b, NN01, ZLN+17], comme par exemple, pour déterminer les positions correspondantes de marqueurs *mocap* situés sur le visage d'un acteur sur un maillage $3D$ aux proportions différentes de celles de l'acteur.

Par ailleurs, Zell et al. [ZLN+17] appuient l'idée que les différentes bases d'une représentation par *blendshapes* représentent des expressions extrêmes et qu'elles ne sont pas forcément les mêmes que celles de l'acteur (ou d'un autre avatar). Ils proposent une méthode permettant d'adapter les bases du modèle *blendshapes* afin d'aligner ces expressions extrêmes avec celles de l'acteur.

Avec un objectif différent, un outil proposé par Sorkine et al. [SCOL+04], intitulé le *Laplacian Coating*, permet de transférer les détails fins d'un maillage vers un autre maillage. Il ne s'agit pas de transfert d'expression à proprement parler, mais de transfert de détails qui améliore sensiblement la crédibilité des animations produites lorsque le maillage animé n'est pas une représentation du visage de l'acteur [XCLT14].

On retrouve également l'idée de travailler sur différents niveaux de détails dans plusieurs travaux. Parmi ces derniers, Bickel et al. [BBA+07b] proposent de combiner deux méthodes indépendantes afin de suivre à la fois les déformations larges liées aux expressions via la méthode *thin-shell*, et les déformations plus fines liées aux rides via un modèle paramétrique. Les paramètres de ce dernier sont déterminés en analysant les ombres de la texture à partir de données capturées via de la *mocap* optique *RGB* multi-vues basée marqueurs. Dans ce cas, les marqueurs sont directement peints sur le visage. Similairement, à partir de vidéos *RGB* mono-vue, Shi et al. [SWTC14] utilisent des modèles faciaux multi-linéaires [CWLZ13, VBPP05] pour suivre les déformations larges, et également les ombres pour les détails fins, en calculant une carte de normales qui permet d'en déduire la profondeur. Enfin, Bermano et al. [BBB+14] utilisent la méthode de [BHB+11b] pour suivre les déformations larges et celle de [SP04] pour les détails fins.

2.3.7 Méthodes guidées par des formes 2D

Cette famille de modèles a été initiée notamment par les travaux de Cootes et al. [CTCG95, CC06, CET01] en 1995. Diverses méthodes sont décrites dans l'état de l'art de Salam et al. [SS18]. Elles regroupent les modèles *ASM* (*Active Shape Model*), *CLM* (*Constrained Local Model*) et *AAM* (*Active Appearence Model*). Une revue des méthodes *AAM* peut être trouvée dans [GSLT10]. Il s'agit de techniques principalement utilisées pour l'analyse et la synthèse d'images 2D mais

il est intéressant de les évoquer ici pour plusieurs raisons. Tout d'abord, leur champ d'application fait écho aux thèmes abordés dans cette thèse. [TMM+09, TMCB07] et [CDB02] ont utilisé ces modèles respectivement pour faire du transfert d'expressions faciales entre visages (entre images 2D) et pour de l'extraction de style (en l'occurrence pour des expressions et des déformations labiales). Ensuite, certains travaux [XBMK04] basés *AAM* permettent d'obtenir des formes 3D à partir d'images 2D. Enfin, ces modèles sont représentés par une combinaison linéaire de formes donnée par :

$$s = \bar{s} + \sum_i c_i \phi_i \tag{2.18}$$

avec s une forme définie par un ensemble de points en 2D, $\bar{s}$ la forme moyenne d'un ensemble de formes s_i sur lesquelles a été effectuée une analyse en composante principale (*ACP*), c_i le $i^{\text{ème}}$ paramètre et ϕ_i le $i^{\text{ème}}$ vecteur propre. Cette représentation n'est pas sans rappeler la représentation par *blendshapes*, elle aussi représentée comme une combinaison linéaire de formes.

2.3.8 Animation faciale à partir d'images 2D

Au cours des dernières années, l'animation faciale à partir de vidéos 2D a gagné en efficacité et en robustesse. Les auteurs travaillent fréquemment à partir de points caractéristiques (*landmarks* ou *feature points*) localisés sur des positions précises du visage [CHZ14, LXC+15, SWTC14]. Des méthodes permettant de localiser ces *landmarks* sur des images 2D sont présentées dans [RCWS14, CWWS12]. Pour obtenir des animations 3D à partir de ces *landmarks*, [CWLZ13, CWZ+14] utilisent un modèle bilinéaire afin de générer automatiquement un maillage 3D et des bases de *blendshapes* adaptées à l'utilisateur à partir de poses spécifiques prises par l'utilisateur lors d'une première phase d'enregistrement. Ensuite les coefficients du modèle et les transformations rigides (rotations + translations) peuvent être suivis en temps réel. C'est la méthode également utilisée par [LXC+15] et [SWTC14].

2.3.9 Mouvement labiaux

Les mouvements labiaux sont parmi les plus compliqués à reproduire automatiquement avec une précision véritablement convaincante. L'implication de la communauté issue de la synthèse de la parole dans le domaine a apporté un certain nombre d'idées intéressantes. La notion de *visème*, équivalent visuel des phonèmes, a notamment été introduite et est pertinente dans la mesure où il existe une corrélation entre le son émis et les mouvements labiaux. Dans certains travaux [LXC+15] [KAL+17] les auteurs n'ont pas hésité à utiliser des réseaux de neurones

artificiels profonds, encore peu utilisés en animation faciale basée données.

2.4 Bilan et discussion

En ce qui concerne la capture de données faciales, les méthodes reposant sur des dispositifs *RGB-D* (lumière structurée, objectifs multiples) ont gagné de l'ampleur au cours de la dernière décennie, notamment grâce aux travaux de [ZSCS04], [BHPS10a] et [BHB$^+$11a]. Ce nouveau type de technologie présente l'avantage d'offrir un degré de résolution spatiale très élevé si bien qu'il devient possible de reconstituer et suivre un maillage $3D$ avec un niveau de détails très fin [BHB$^+$11a] tout au long de la capture. Néanmoins ce type de dispositif demeure sensible aux mouvements brusques, ainsi qu'aux modifications de luminosité et le visage de l'acteur doit se trouver à une distance relativement courte de la ou des caméra(s). De plus les méthodes d'inférence de la profondeur des techniques *RGB-D* reposent souvent sur des méthodes de flux optiques ce qui les rend sensibles aux dérives numériques, aussi ne sont-elles pas recommandées pour la capture de longues séquences. Par ailleurs en terme de volume de données, là où la *mocap* ne capture les positions que d'un nombre limité de marqueurs, les techniques de types *RGB-D* sont nettement plus coûteuses.

La capture de type *mocap* permet de passer outre ces limitations. De plus, ce dispositif de capture est aussi bien adapté à la capture faciale que corporelle, cela simplifie le travail lorsque l'on souhaite une capture simultanée de ces deux types de mouvements, notamment en ce qui concerne la synchronisation des systèmes. Enfin, la fréquence d'échantillonnage élevée (nombre de *frames* capturées par seconde) des dispositifs de type *mocap* peut se révéler particulièrement intéressante lorsque l'on cherche à étudier des mouvements rapides tels que les micro-expressions. En définitive, si les méthodes de type *RGB-D* permettent de capturer efficacement et précisément les expressions faciales et que ces techniques continuent d'évoluer, il existe toujours un certain nombre de raisons justifiant l'emploi de la *mocap*.

Néanmoins au cours des dernières années les méthodes d'animation faciale basées données ont également gagné en efficacité et en popularité. Les travaux de [CWLZ13, CWZ$^+$14] reposent sur une base de données d'expressions faciale combinant vidéos *2D+z* et avatars *3D*. Cette approche a permis d'employer une méthode d'apprentissage automatique (un modèle bilinéaire) grâce à cette base de données d'entraînement et d'automatiser la production d'animations *3D* en temps réel à partir de vidéos *2D* capturées via une webcam. On peut envisager que si davantage de *BDD* sont créées et rendues accessibles, les réseaux de neurones profonds qui requièrent une quantité de données d'apprentissage importante soient à l'avenir employés afin

d'améliorer les résultats.

Les méthodes d'animation faciale à partir de données sont généralement formalisées à partir de techniques d'optimisation. Deux types de modèles émergent dans la communauté d'informatique graphique, qui sont représentatifs de deux philosophies différentes.

Le modèle à couches fines s'intéresse à la détermination optimale des positions de l'intégralité des sommets du maillage $3D$ à partir des positions d'un sous ensemble de sommets contraints. Ce type de modèle offre une grande liberté de déformation tout en préservant la cohérence du maillage. Par ailleurs, le problème est ramené à un système linéaire ce qui permet des temps de calculs compatibles avec du temps réel même si certaines précautions doivent être prises (à savoir, les sommets contraints doivent toujours rester les mêmes). Cependant, si ce type de méthode est adapté pour animer un maillage similaire au visage de l'acteur (pour de la *mocap* par exemple, il suffit d'établir une correspondance entre chaque marqueur *mocap* et un sommet du maillage de destination, puis de contraindre ces sommets afin qu'ils suivent les déplacements des marqueurs), transférer une expression faciale vers un maillage différent n'est pas aussi simple. Il est alors nécessaire d'établir une correspondance entre la structure du visage de l'acteur et celle du maillage cible (par exemple via une interpolation RBF). Une autre méthode consiste à réaliser préalablement l'animation d'un maillage similaire à celui de l'acteur, avant de transférer les déformations qui lui sont appliquées (par exemple via la méthode de [SP04]), ce qui complexifie le problème.

Le modèle basé *blendshapes* s'appuie quant à lui sur une représentation compacte du visage. Le calcul des paramètres et du maillage final à partir de ces derniers est rapide. Mais surtout le transfert d'une expression d'un maillage à l'autre peut se faire directement à condition que les bases du premier modèle représentent les mêmes expressions que les bases du second modèle. Néanmoins, il repose sur l'hypothèse selon laquelle les expressions faciales sont toutes représentables par combinaison linéaire d'un nombre limité d'expressions de base (ce qui n'est pas exact). Ainsi l'espace de représentation défini par ce type de modèle est plus restreint que l'espace défini par l'expressivité faciale réelle, il en résulte que les animations produites par *blendshapes* peuvent paraître légèrement moins naturelles (plus mécaniques) que celles produites via une méthode présentant un plus grand nombre de degrés de liberté.

CHAPITRE 3

CAPTURE DES DONNÉES

Préalablement à la capture de données basée *mocap* qui nous a conduit à construire plusieurs *dataset*, à la fois pour la synthèse et l'annotation automatique, nous avons effectué une étude dans le but de déterminer un jeu de marqueurs optimal, à la fois au niveau du nombre de marqueurs et de leur disposition. Ce chapitre fait l'objet d'une publication dans la conférence MIG 2015 [RGL15b].

We seek to determine an optimal set of markers for marker-based facial motion capture and animation control. The problem is addressed in two different ways : on the one hand, different sets of empirical markers classically used in computer animation are evaluated ; on the other hand, a clustering method that automatically determines optimal marker sets is proposed and compared with the empirical marker sets. To evaluate the quality of a set of markers, we use a blendshape-based synthesis technique that learns the mapping between marker positions and blendshape weights, and we calculate the reconstruction error of various animated sequences created from the considered set of markers in comparison to ground truth data. Our results show that the clustering method outperforms the heuristic approach.

3.1 Introduction

The animation of virtual characters has numerous applications, from entertainment such as video games or movies, to other serious game applications involving interaction with avatars for communication or education purposes. To make these avatars appear more attractive and realistic, special care must be taken at several levels of animation, e. g., behavior, body and hand movement, and facial animation. Using captured motion on real actors provides the ability to animate virtual characters with credible behavior and thus reinforce their comprehension and acceptability. Furthermore, data-driven approaches go beyond the production of realistic animations : they allow for a detailed analysis that might help to extract and understand some relevant key postures and dynamical features, and make possible the use of pre-recorded mo-

tion, through editing and synthesis operations. However, this approach is still challenging the computer animation community, in particular for multi-channel animation involving the simultaneous control of facial expression, hand and body movements in expressive or linguistic tasks. In this paper we focus on expressive facial animation with the aim to further linguistically edit our data within a concatenative synthesis framework dedicated to virtual signers [GCDLN11].

There are two possibilities to capture facial motion : either marker-based or markerless techniques. Different reasons may be argued to prefer one over the other solution. As our future aim is to animate signing avatars through motion capture, our application requires to capture full-body movements, including body and hand motion, facial expression and gaze direction. In this particular case, marker-based motion capture (MoCap) is necessary because the different body channels need to be captured simultaneously, each channel conveying a specific meaning.

One challenge in facial marker-based motion capture is the choice of the marker layout. Numerous empirical facial MoCap layouts have already been proposed to capture facial expressions and control facial animation systems. However little work has been done to determine and evaluate the best marker set for motion acquisition and animation.

In this paper, we propose a method to compute an optimized marker set for facial motion acquisition and synthesis control. Our aim is not so much to produce animations with high accuracy and realism as this is done in [LZD13], but to be able to easily capture new sequences of facial and full-body motions, avoiding as much as possible the intervention of skilled animators. That is why we seek the best trade-off between complexity (we try to minimize the number of markers required) and the quality of the produced animation (which should best render the facial expressiveness). In addition we want to remain as independent as possible of the synthesis method to facilitate the reutilization of the corpus in different contexts. We propose a dual heuristic / automated approach : after analyzing the capability of different empirical maker layouts used in previous studies to produce credible facial animations, we automatically determine marker layouts by using a clustering technique, and evaluate these marker layouts by using the same synthesis technique.

In the remainder of the paper, we first review the related work. Then section 3.3 presents the methodology we have used to create our animations and to evaluate the efficiency of each marker set we have tested. Section 3.4 presents the data set on which the algorithms are applied. Section 3.5 presents the results obtained for empirical marker sets, while section 3.6 defines a novel approach to automatically compute optimized marker layouts and shows the corresponding results. Finally section 3.7 concludes and proposes directions for future work.

3.2 Related Work

Performance-Based Motion Capture. Marker-based facial motion capture usually consists in following the $3D$ positions of markers disposed on an actor's face with a network of cameras. This method presents the advantage of allowing a capture with a very high frame rate (≥ 120 fps) and a good precision (≤ 1 millimeter). However, since the number of markers that can be put on the actor's face is limited, this method can only output a sparse representation of the face. So if this technique efficiently captures large scale deformations, it is not sufficient for fine scale details like wrinkles.

In order to get a dense and direct mesh representation of the actor's face, several markerless methods relying on structured light [ZSCS08] or stereo cameras [BHPS10b] [BHB$^+$11a] have been developed. These methods allow for the acquisition of a series of high resolution triangle meshes at 30 to 42 fps. However these methods are more sensitive to light conditions than the traditional marker-based motion capture techniques. Moreover, applications that necessitate simultaneous capture of body and facial motion are incompatible with the capture conditions required by these methods.

Marker Set Optimization. Amazingly, whereas the marker placement is a decisive choice, only few studies have been dedicated to this issue. Private companies usually exploit empirical marker layouts which are often homemade. In [LZD13] a method is proposed to automatically determine an optimal marker set, the optimization process relying on the minimization of the reconstruction error from the ground truth with respect to the chosen animation synthesis method - so that the marker set found can be optimized for this synthesis method in particular.

The problem of marker optimization is also similar to some of the issues related to mesh compression where the shape of the whole mesh may be determined from a subset of its vertices [SCO04], [MA07], [SZ11]. Our approach is similar to [SZ11] and [SSK05] who have experimented K-means-based methods for mesh compression.

Animation Synthesis from Marker-Based Motion Capture. Facial animation by blendshapes from motion capture is a well-known technique. Following this method, several facial key shapes, designed most of the time by an animator, are blended to produce appropriate facial animations. Blendshapes present the advantage of providing both a level of abstraction and a compact representation which allows easiness of editing and retargeting to other blendshape-based facial models. However, it is still necessary to find a mapping between marker positions and blendshape coefficients. The quality of the animation is also dependent on the choice of key shapes.

In [WBLP11b], the authors propose a method to provide a mesh representation of the user's face and its corresponding blendshapes [WLVG08b, LAGP09, LWP10b] from data captured via a depth sensor camera. Then, blendshape weights are computed by minimizing a cost function taking into account both the geometry and the texture of the model.

Other methods such as the least squares mesh technique [SCO04] or the thin-shell model [LZD13, BS08] may be used. Both techniques are based on the same principle : the positions of all vertices of a mesh are directly estimated from the positions of a subset of these vertices.

In this paper, we do not focus on fine scale details but on easiness of data acquisition and genericity. For this reason we have chosen to rely on a blendshape animation system to compute facial deformations. Moreover, we argue that the system described in [WBLP11b] is able to provide data of sufficient quality to both serve as training data for this algorithm but also to serve as ground truth for the evaluation of our results.

Cross-Mapping of Facial Data and Blendshape Parameters. The process of cross-mapping MoCap data and blendshape parameters is not trivial as it is a one-to-many mapping due to the fact that multiple blendshape weights combinations may lead to the same facial configuration. Traditional approaches identify pairs of MoCap data and blendshape parameters that are carefully selected and designed by the animator [DCFN06b]. These pairs are then used in a learning process that determines the selection of corresponding blendshape parameters from new MoCap data input values. Other current methods largely rely on radial basis functions and kernel regression to achieve these steps [CTFP05, DCFN06b, LMX$^+$08]. However, such methods have several drawbacks : a number of localized basis functions have to be chosen prior to the learning process, and the result is conditioned by the quality and density of input data. Thus, noisy input often yields bad estimates, this being known as the classical over-fitting problem.

In our work, we need to simultaneously record body, manual and facial data at high frequency rates. Such technical difficulty may result in noisy positions of the facial markers. Therefore, the mapping between motion capture data and blendshape parameters is done via a machine learning algorithm. We consider the problem as a Bayesian inference problem. Instead of incorporating explicit basis functions (such as radial basis functions), we use a Gaussian Process regression technique to describe a distribution over functions that map the MoCap data and the blendshape parameters [RW05].

3.3 Methodology

We use two different approaches to find the best possible marker set : an empirical approach detailed in section 3.5 and an automatic clustering approach detailed in section 3.6. The empirical approach evaluates different facial MoCap marker layouts that have been used in previous work. The automatic approach takes as input the vertex positions of the facial mesh for all of the frames of a training sequence and computes through a clustering technique the optimized marker layout for a given number of clusters. Both approaches are evaluated through the same synthesis technique that achieves the mapping from motion capture data to blendshape weights.

3.3.1 Animation Synthesis

For each marker layout empirically or automatically determined, the synthesis can be decomposed into three steps : (i) for a long sequence of facial animation referred as range-of-motion sequence, we first take as training examples markers-weights pairs (X_i, Y_i) that describe the marker positions and the corresponding blendshape weights at each frame i ; (ii) for this training data, a Gaussian Process regression technique (GP) learns the mapping between the MoCap data and the weights ; (iii) for new MoCap test data, the new blendshape weights are estimated from the same GP regression technique (see figure 3.1).

Facial *mocap* Representation. In traditional marker-based facial motion capture, captured data can be represented by a sequence of N facial poses over time $X = (X_1, ..., X_i, ..., X_N)^T$. Each facial pose X_i at frame i is encoded by a $3 \times K$ dimensional vector, K being the number of markers, and x_i^k the 3D position of the k^{th} marker : $X_i = (x_i^1, x_i^2, ..., x_i^K)$.

Blendshape Representation. In blendshape animation, a mesh is represented by a neutral shape and a set of basic deformations of this shape, where each basic deformation applied to the neutral shape represents a particular pose (e.g., one of the Action Units of the Facial Animation Coding System [EF78]). Hence the shape B_i of the mesh at frame i is defined as a linear combination of these basic deformations :

$$B_i = B_0 + \sum_{l=1}^{L} w_i^l B^l \tag{3.1}$$

where B_0 is the neutral shape, L is the number of basic deformations, B_l is the l^{th} basic deformation and w_i^l its associated weight. Let $Y_i = (w_i^1, w_i^2, ..., w_i^L)$ be the vector of the L weights at frame i. We can express the blendshape animation sequence as the vector $Y = (Y_1, ..., Y_i, ..., Y_N)^T$.

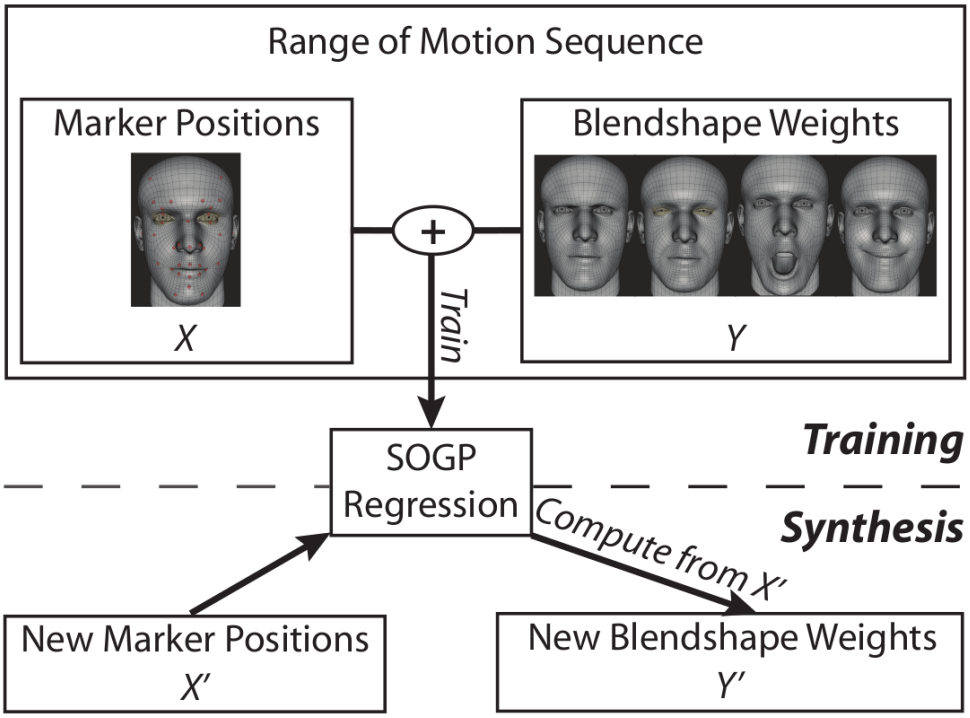


FIGURE 3.1 – Animation synthesis overview.

Animation by Gaussian Process Regression. Formally, we consider a set of N observations $\{(X_i, Y_i), i = 1...N\}$, where X_i denotes the input vector (facial marker positions at frame i), and Y_i denotes the output vector (blendshape weights at frame i). In a GP model, we make the assumption of a double stochastic process on the distribution f and the noise ϵ_i :

$$Y_i = f(X_i) + \epsilon_i \tag{3.2}$$

and for all the observations :

$$Y = f(X) + \epsilon \tag{3.3}$$

The Gaussian Process is defined as :

$$f(X) = GP(\mu(X), K(X, X')) \tag{3.4}$$

where $\mu(X)$ is the mean function and $K(X, X')$ the covariance function. Based on the set of input-output observations, the Bayesian approach computes the posterior distribution of the real process f using the prior and the likelihood [RW05].

One major drawback prevents GP from being applied to large datasets : the computation

of the covariance matrix which is highly costly. To overcome this limitation, we used a derived method called Sparse Online Gaussian Process (SOGP) which combines a sparse representation (using a smaller subset of input data) and an online algorithm of the posterior process [CO02].

3.3.2 Quality Measurement

To evaluate the different marker sets, we compare the animations produced via our animation synthesis method using virtual markers with the animations produced by a reference animation system presented in section 3.4 and considered as the ground truth data. The quantitative measure of quality is based on the root mean squared error (RMSE) calculated from the positions of the vertices of the mesh for the ground truth data and the synthesized ones :

$$RMSE = \sqrt{\frac{1}{N}. \sum_{n=1}^{N} \sum_{p=1}^{P} \|v_n^p - \hat{v}_n^p\|^2} \tag{3.5}$$

where N is the number of frames, P the number of vertices, v_n^p the position of the p^{th} vertex at the n^{th} frame, and $\hat{v}_n^p$ the position of the same vertex at the same frame in the ground truth mesh.

RMSE results presented in this study are computed over all frames of all test sequences performed by the concerned actor except the range-of-motion sequence (see section 3.4).

3.3.3 Notations

In section 3.5, we consider two categories of empirical marker sets : the first category, named STAR, denotes state-of-the-art marker layouts that have proven to be satisfactory for target applications ; the second category, named MAN, denotes manual marker layouts manually defined from a given existing marker set. In section 3.6, each clustering experiment uses a marker set named K-means. Moreover, each K-means experiment is based on a range-of-motion sequence performed by an actor Φ. Therefore, a given marker set will be labeled STAR, MAN, or K-means_Φ, Φ representing the actor, $\Phi \in \{A, B, C, D\}$. It may be optionally followed by the number of markers. Given one marker set, the corresponding synthesis can be defined by its training sequence, applied on the range-of-motion of actor Ψ, and its test sequence applied on the test sequences of actor Ω. The synthesis will be labeled train_Ψ, synth_Ω.

3.4 Data Set

Since our synthesis process is based on learning algorithms, we need to get data on which these algorithms can be first trained and on which to rely as ground truth data to evaluate our resulting animations.

3.4.1 Ground Truth Data

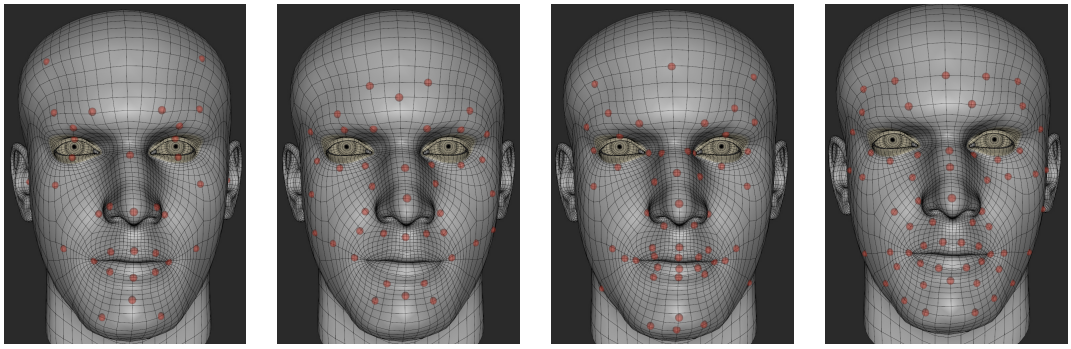
We decided to use the Faceshift commercial software [fac12] as a reference system to easily collect the data required by the learning algorithms. We chose this system for several reasons. Firstly, it offers a flexible mesh and a set of blendshapes that match the face of the actor. The basic blendshapes are based on key shapes from the FACS coding system, thus producing dynamical facial animations that are quite similar to those of the human face. Moreover, this system provides a good approximation of the large scale facial deformations. Secondly, the technique employed by Faceshift produces pairs of selected virtual markers on the actor's face and blendshape weights. This allows us to learn the mapping of the markers-weights pairs, using our GP regression model, thus avoiding the tedious work of manually tuning the blendshape weights [DCFN06b]. Using this system is a good way to collect a large amount of training data with different subjects.

To simulate each marker set with Faceshift (virtual markers), each marker has been positioned on a vertex of the actor's mesh. The training sequences consist of time series (about $1min30$ to $2min$ at 30 fps) of facial expressions performed by each actor.

For each actor, the mesh and the corresponding set of blendshapes optimized for this actor have been computed and exported. Regardless of the actor, the produced facial mesh has the same number of vertices (12021) and the same topology (connectivity matrix). The performances are captured via a depth sensor camera and the animations produced by Faceshift are output as the sequences Y of blendshape weights. These sequences are considered as a fair enough approximation of the ground truth for the purpose of this study.

3.4.2 Corpus

Our corpus contains the facial expressions of four non-deaf actors named A, B, C and D. There is one range-of-motion sequence per actor, designed for training and normalizations goals. Each actor/tress was instructed to freely explore the deformation capability of his/her face, making grimaces during $1min30$ to $2min$. The purpose of this sequence is to capture the



facerobot
35 markers

signcom
41 markers

mpeg4
53 markers

sign3d
60 markers

FIGURE 3.2 – STAR marker sets.

facial deformation space specific to each actor. Apart from the training sequence, each actor has performed between 25 and 30 sequences lasting from $2s$ to $6s$ and recorded at a frequency of 30 fps. Each sequence has been performed once. For each sequence, the actor was instructed to watch a video and then mimic the facial expression he has seen. The videos are small sentences in French Sign Language with emotional content performed by deaf people.

3.4.3 Preprocessing

Each sequence X (input MoCap data) of a given marker set performed by a given actor is centered and scaled by respectively the mean vector and the standard error vector computed on the training range-of-motion sequence of this actor with the same marker set. The positions of the virtual markers for each sequence (both training and testing) have been centered and divided by the standard deviation of the training sequence of the corresponding actor.

3.5 Empirical Marker Sets

In this section, we consider the following two categories of empirical marker sets : STAR and MAN. The results are presented with respect to the notation introduced in section 3.3.3.

3.5.1 State of The Art (STAR) Marker Sets

In our first approach we have tested different known marker sets of different sizes (see figure 3.2) :

— MPEG4 (53 markers) : the set of Facial Feature Points used in the MPEG4 standard ;
— Face Robot (35 markers) : the set of markers used in the commercial software Autodesk Face Robot (Softimage) ;
— SignCom (41 markers) : the set of markers used in [GCDLN11] designed for French Sign Language (FSL) capture ;
— Sign3D (60 markers) : the set of markers used in [LAGT+13] designed for FSL capture.

We have then quantified the RMSE error between the ground truth data and the synthesized data. As shown in figure 3.3, this first experiment does not yield any significant quantitative results

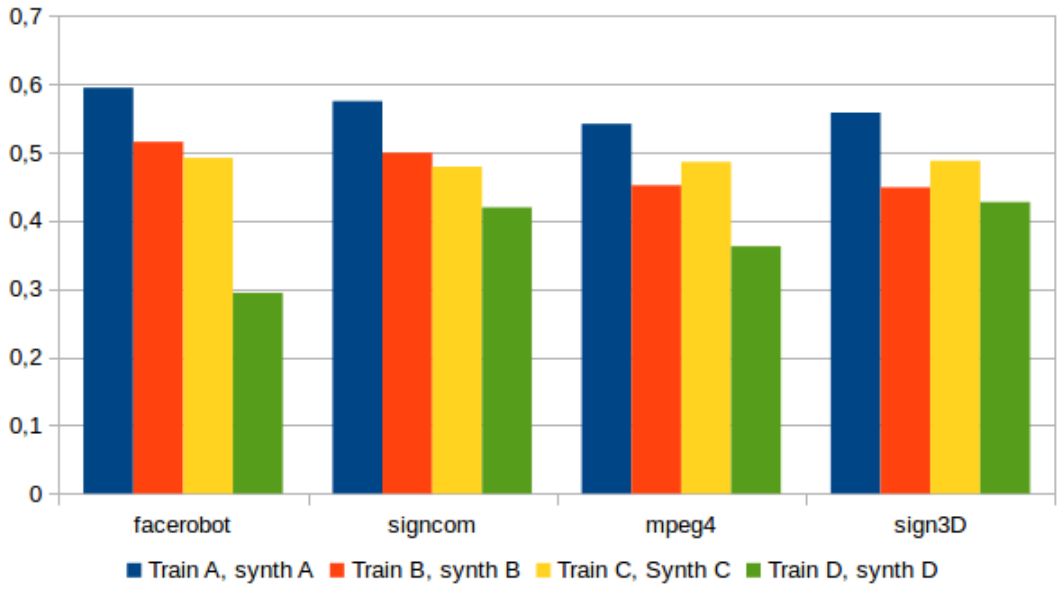


FIGURE 3.3 – RMSE of synthesis for the STAR marker sets and the same actors in training as in testing sequences.

in terms of errors. Nevertheless, the spatial placement of markers (figure 3.2) is different from one marker set to another. Accordingly, errors on the synthesized data are distributed differently along time and space (face regions) for the various marker sets. For example (see figure 3.4), data synthesized with the sign3D marker set has a higher error rate on the eye region than the Face Robot marker set.

3.5.2 Manual (MAN) Marker Sets

In order to analyze the influence of the number of markers on the synthesis quality, we manually established a set of manual marker layouts. We initially took a mix of 62 markers

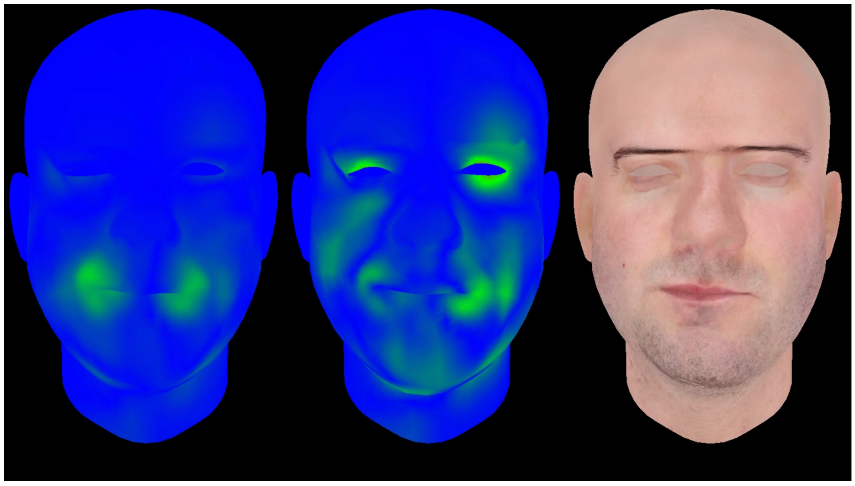


FIGURE 3.4 – Left : RMSE with Face Robot marker set ; center : RMSE with Sign3D marker set ; right : original animation sequence.

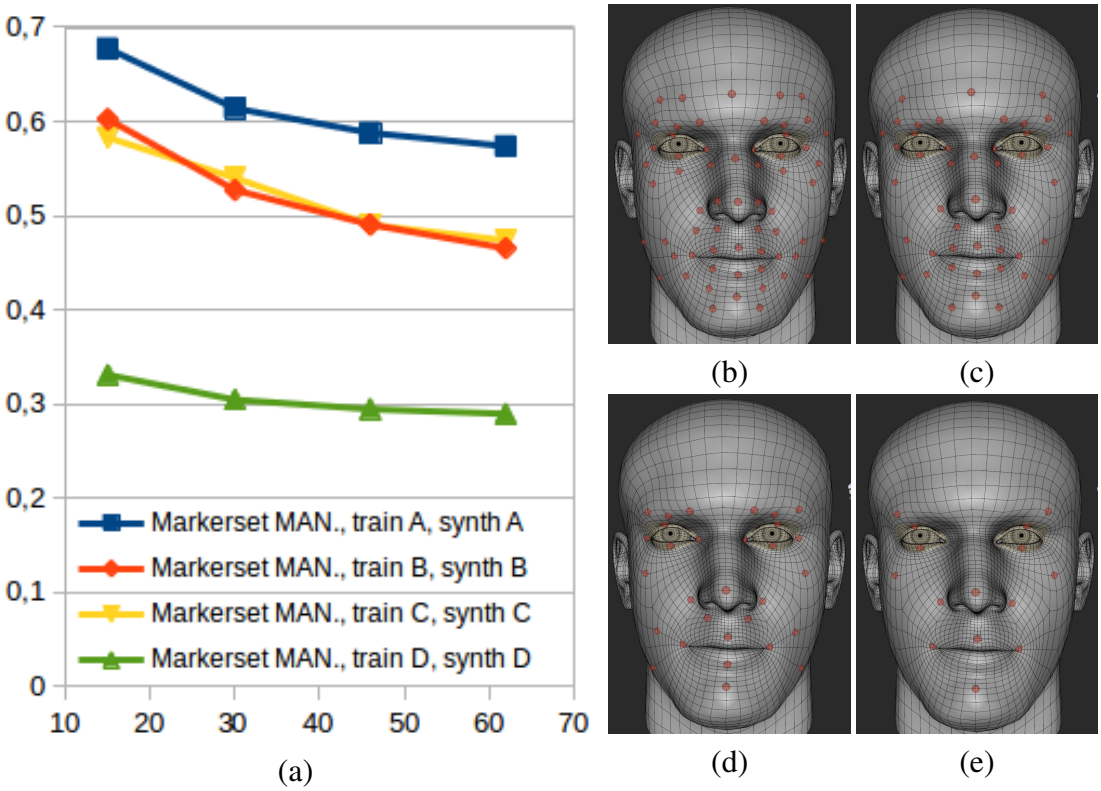


FIGURE 3.5 – (a) : RMSE on manual marker sets, synthesis achieved with the same actor in training and test ; (b, c, d, e) : the MAN marker sets with respectively 62, 46, 30 and 15 markers.

belonging to the STAR marker layouts, and then successively took off some markers from it.

We thus formed four marker sets of sizes 62, 46, 30 and 15 (see figure 3.5 (b,c,d,e)). We removed in priority the markers that visually appeared the less useful.

As expected, figure 3.5 (a) shows a constant decrease of the RMSE according to the number of markers. However, beyond 35 or 40 markers, we can observe on the basis of the RMSE error computed along the whole sequence, that the gain is not significant.

3.6 Automatic Determination of Marker Sets via Unsupervised Clustering

3.6.1 K-means Clustering Method

The key idea is to apply a clustering technique on the vertices of a training data set that mostly covers a large part of the human facial expression space. Let V be the matrix that represents the sequence of the mesh deformation where $\bar{v}_n^p = \frac{v_n^p - \mu_p}{\sigma_p}$ is the normalized position of the p^{th} vertex at the n^{th} frame (with μ_p and σ_p respectively the mean vector and standard deviation vector of the p^{th} vertex over time) :

$$V = \begin{bmatrix} \bar{v}_1^1 & \cdots & \bar{v}_n^1 & \cdots & \bar{v}_N^1 \\ \cdots & \cdots & \cdots & \cdots & \\ \bar{v}_1^p & \cdots & \bar{v}_n^p & \cdots & \bar{v}_N^p \\ \cdots & \cdots & \cdots & \cdots & \\ \bar{v}_1^P & \cdots & \bar{v}_n^P & \cdots & \bar{v}_N^P \end{bmatrix} \tag{3.6}$$

In our approach, the p^{th} line V_p of this matrix (i.e. the whole trajectory of the p^{th} vertex along the training sequence) is considered as an observation. The K-means algorithm aims at partitioning these observations into K clusters C^k by iteratively assigning each observation to the nearest cluster C^k represented by its centroid m^k such as :

$$C^k = \{V^p / \|V^p - m^k\| \leq \|V^p - m^{k^*}\|, \forall k^* \in \{1, ..., K\}, k^* \neq k\} \tag{3.7}$$

We then recompute each cluster's centroid from its assigned vertices and repeat iteratively both operations until convergence. The centroids have been randomly initialized and only vertices that are affected by at least one blendshape are considered (4859 on 12021).

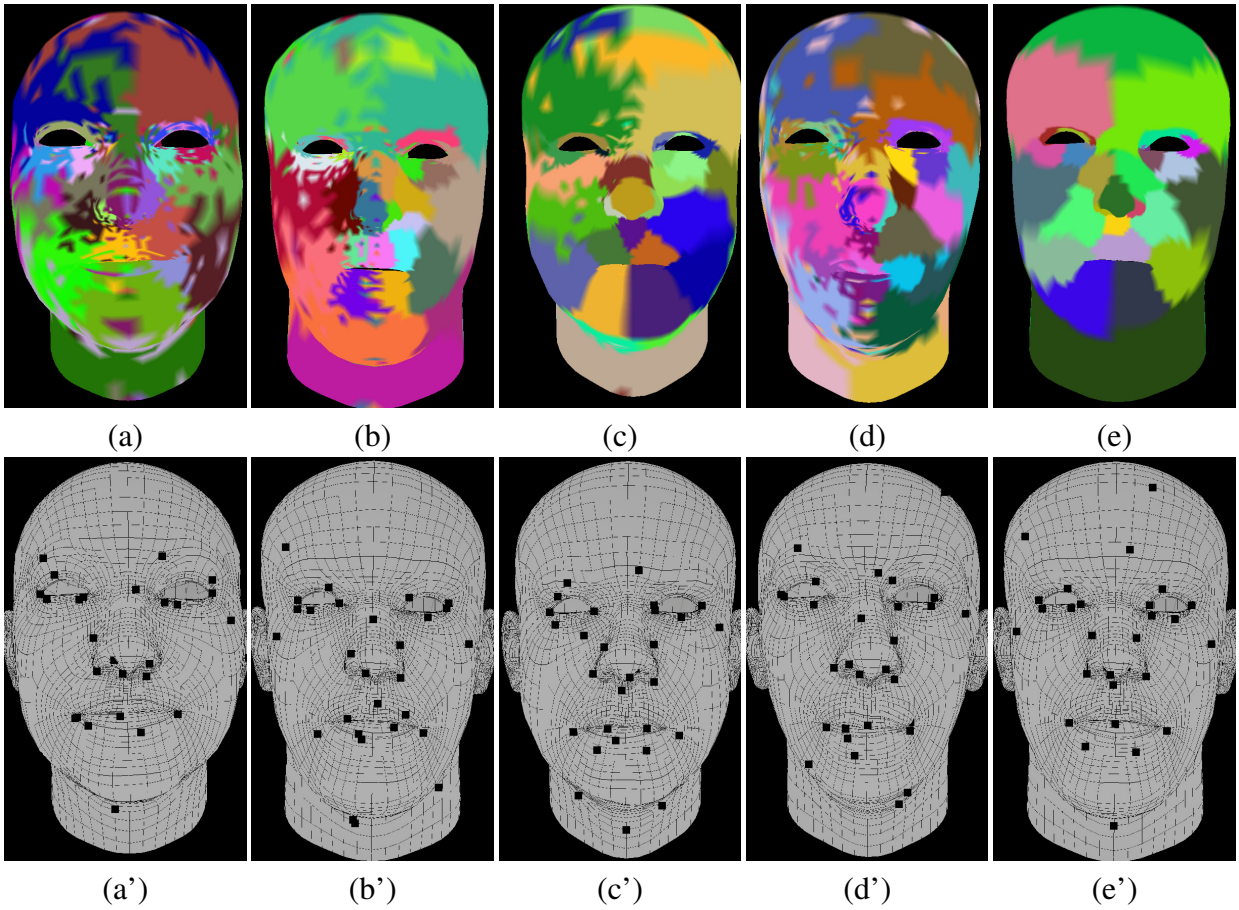
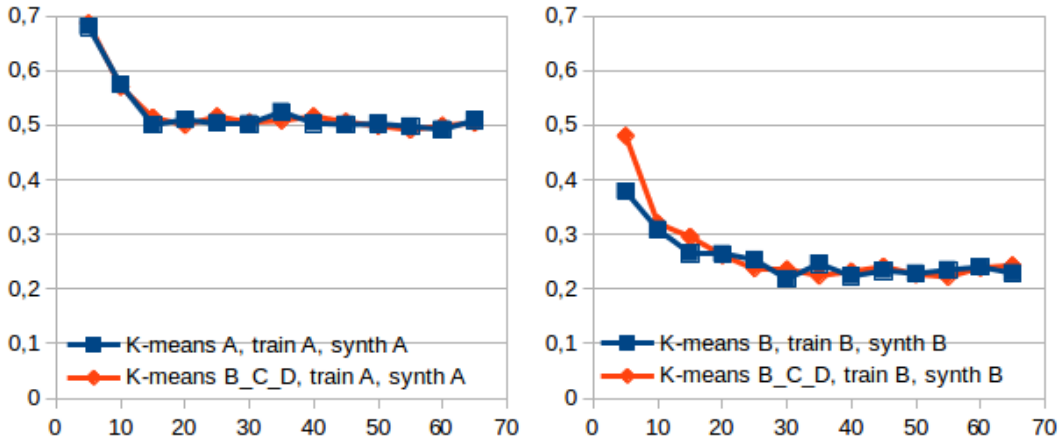


FIGURE 3.6 – Clusters and corresponding marker sets automatically determined by applying our K-means clustering algorithm (with $K = 30$ clusters) on the range-of-motion sequences of actors A (a,a'), B (b,b'), C (c,c'), D(d,d') and combined sequences of actors B+C+D (e,e').

3.6.2 Results

We first analyze the influence of the training matrix on the quality of the markers found by the clustering K-means technique. As illustrated in figure 3.7, the evolution of RMSE over the number of markers doesn't change, whether the K-means is applied from a training achieved on one actor - actor A (left) or B (right) - or on several actors - actors B, C and D - for which the training sequences are concatenated. Moreover, we can see that the error rapidly decreases with the increase of the number of markers and reaches a minimum at about 30 markers. This preliminary analysis therefore highlights that taking a large amount of clusters will not lead to better results.

Nevertheless, as shown in figure 3.6, the regions that are determined by the K-means algorithm are less fragmented and the marker placement appears more symmetric when the cluste-



Actor A Actor B

FIGURE 3.7 – Influence of the training dataset on the quality of markers found by clustering. Red : results with K-means trained on 3 actors ; blue : results for K-means trained on only one actor.

ring is performed on training data from multiple actors. For these reasons, it seems best to keep the marker sets automatically determined from the combined training data of the actors B, C and D.

We also compared the performances of the marker sets obtained by the K-means clustering technique with the empirical marker sets STAR and MAN applied to the four actors. figure 3.8 shows that the marker sets automatically determined via K-means clustering always leads to better results than the empirical marker sets.

3.7 Conclusion

We presented two approaches to help in the pose of markers for facial motion capture. In a first approach we tested empirical marker sets that have previously proven to give satisfactory results in blendshape animation from MoCap data. In a second approach, we used the K-means clustering method to partition facial meshes into geometrical regions that are significant for given actors. The results, quantified though the error between ground truth and synthesis animations, seem coherent. However, the automatic clustering method gives better results in terms of RMSE error. After a training on each actor, we showed in particular that it was possible, for a given number of clusters, to determine a good partitioning of vertices that show similar dynamics. Moreover, with a training performed over several actors, we showed that it was pos-

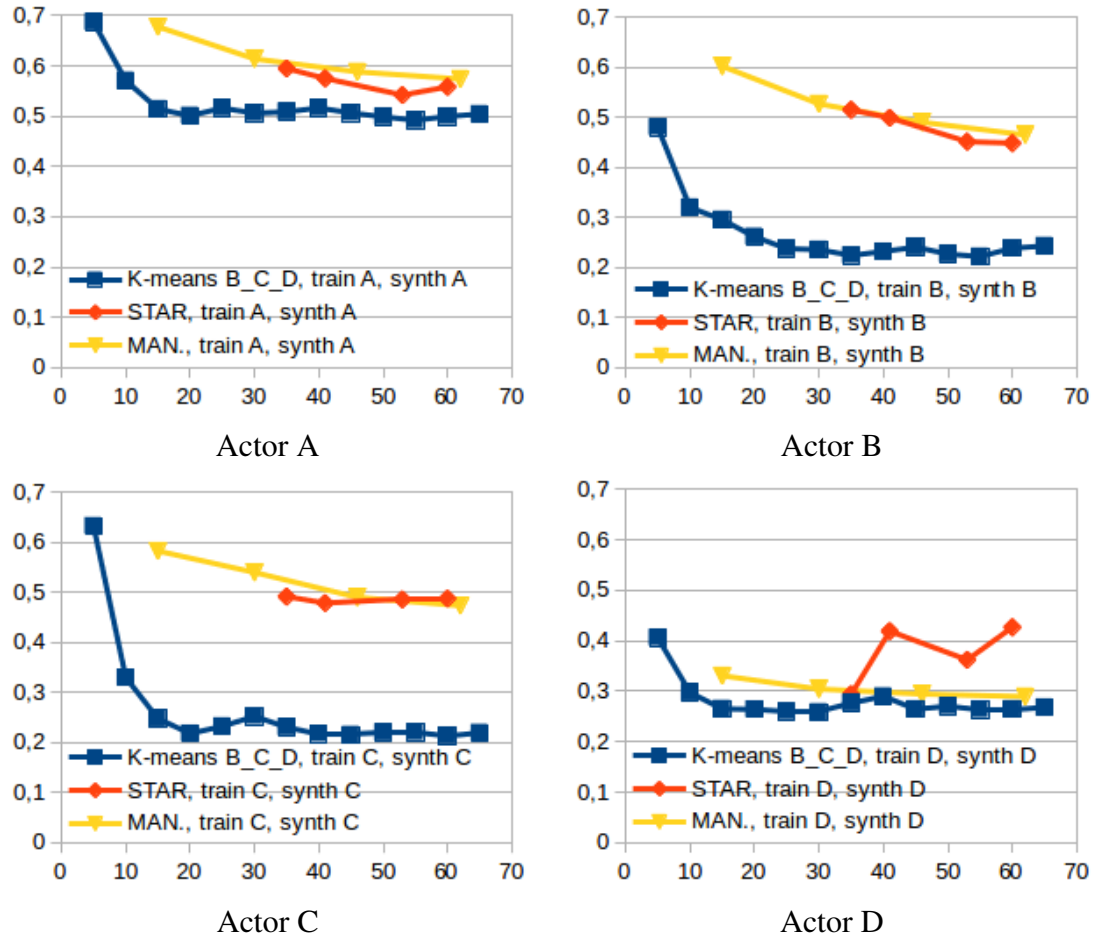


FIGURE 3.8 – Comparison of RMSE between STAR, MAN and K-means marker sets. The K-means is performed on combined range-of-motion sequences of the actors B, C and D.

sible to identify more stable regions, and therefore to deduce reliable candidates for the pose of markers.

Nevertheless, some further investigations are required to complete this statement. In real conditions, the installation of markers is subject to inaccuracies, especially when there are several actors with different facial morphologies. Thus, the robustness to noise of the marker sets determined by this method is still to be verified. Future work that includes the use of these marker sets in real conditions will answer this question.

Furthermore, the facial representation used for facial animation, i.e. the blendshapes, is extremely compact. Hence, on the one hand, the mapping of marker positions to the space of blendshape coefficients is fairly trivial if we consider the abundance of training data available

to us. On the other hand, the simplicity of this representation does not allow the expression of all the subtleties of human facial expression. For these reasons, it would be interesting to test this approach with other animation methods which would allow us to check the genericity of the proposed method.

The clustering approach has several limitations. Although it is independent of the synthesis method in itself, this approach remains dependent on the training data. Furthermore, being independent of the synthesis method will not give a marker set optimized for that synthesis method in particular. Finally, the K-means algorithm seeks to minimize the intra-class variance and to maximize the inter-class variance which leads to the creation of vertex groups that have a similar behavior. This means that the algorithm captures in priority the major sources of deformation leaving out the more subtle sources of deformation such as wrinkles.

CHAPITRE 4

ANNOTATION AUTOMATIQUE DES DONNÉES

L'annotation manuelle est une tâche longue et fastidieuse, en partie à cause du nombre de canaux linguistiques qui structurent les données en langues des signes. Dans ce chapitre, extrait d'une publication dans la conférence LREC 2018 [NRLG18], nous proposons d'automatiser l'annotation des expressions faciales sur le canal affectif. Après une description du corpus de données capturé à cet effet, nous présentons notre méthode pour annoter et étiqueter automatiquement ces données, en passant par les différentes étapes : segmentation, reconnaissance, puis annotation.

Manual annotation is an expensive and time consuming task partly due to the high number of linguistic channels that usually compose sign language data. In this chapter, extracted from our publication at the LREC 2018 conference [NRLG18], we propose to automatize the annotation of sign language motion capture data by processing the facial expression channel. Motion features, such as facial descriptors, that take advantage of the $3D$ nature of motion capture data and the specificity of the channel are computed in order to (i) segment and (ii) label the sign language data. A method of automatic annotation of French Sign Language utterances using similar processes is developed. It describes the annotation of facial expressions using a closed vocabulary of seven expressions. Results are then presented and discussed.

4.1 Introduction

Whether we want to linguistically study sign languages, use digital data to identify salient linguistic components, recognize or synthesize continuous signing, an annotation of the data is needed. The annotation of sign language is a two-step process. The first step, called *segmentation*, consists in dividing the stream of data in segments of interest. Those segments are then identified in a second step called *labeling*. This annotation might be done manually but is a

fastidious, time consuming and expensive task. Not only does it require the skills of language experts, but it is also subject to inaccuracies and mistakes as the experts may not have exactly the same segmentation criteria. In the particular case of sign language, the annotation process needs to be done by experts in sign language and gesture annotation. In addition, sign languages are expressed simultaneously on multiple channels (manual configuration, wrist orientation, body posture, facial expression, etc.), thus complicating the task of the annotators. When comparing the duration of the annotation process to the duration of the data to annotate, [DN08b] introduces a real-time factor of 100 (i.e. all the manual and non manual features of a 1 minute video of sign language will be annotated in 100 minutes). We propose to automatically annotate each channel separately following the scheme of Figure 4.1.

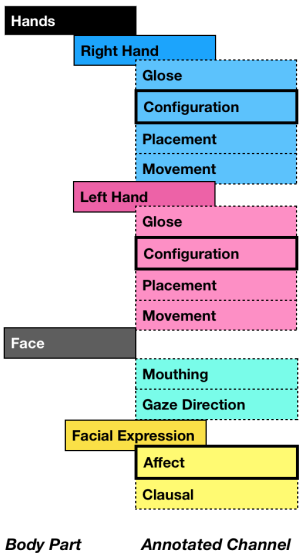


FIGURE 4.1 – Our intended annotation scheme : in this chapter, we propose a method to automatically annotate the affect channel (yellow box with bold border).

Automatic annotation of sign languages could reduce the annotation costs but is still a challenging and yet to be solved task. One way to tackle this problem is the *machine learning* approach (e.g. use of Bayesian/statistical models or artificial neural networks) which aims at automatically learning the parameters of a model from a sample of manually annotated data. This model is then used to automatically annotate new data. The corpus intended to train the machine learning model must thus be designed carefully before recording sign language data using either video or motion capture technologies. Despite being easy to use and relatively cheap, video does not preserve the third spatial dimension of motion. Furthermore, the resolution of classic video recordings is rarely high enough to obtain a precise segmentation. Motion

capture (MoCap) technologies offer a better precision, both spatially and temporally, and spatial information that is not available with 2D cameras. MoCap data can be used for sign language analysis – using motion descriptors computed from the 3D data, such as distances between joints, velocity or curvature of selected joints – as well as for synthesis by using the captured motions to animate a signing avatar. In this paper, we will address the problem of the automatic annotation of sign language MoCap data, focusing more specifically on two main channels : facial expression, and hand configuration.

Previous work on automatic annotation on video and MoCap data are described in section 4.2. The specification, capture and manual annotation of the input sign language data used in this study is introduced in section 4.3. In section 4.4, the annotation methodology is presented and illustrated with the example of the automatic annotation of the facial expressions in French Sign Language. Section 4.5 describes the results of the automatic annotation. Finally, section 4.6 discusses the perspectives and challenges of our method.

4.2 Related Work

In this section, previous studies on automatic annotation of sign language are presented. For gesture segmentation in general, a very complete framework is developed in [LKK16]. It provides a general overview and a comparison of several works in human motion segmentation using different data sources (camera, MoCap, sensors, etc.) but does not address the problem of "per channel segmentation" or the particular application of sign language processing.

Continuous signing segmentation and recognition can be opposed to isolated sign language recognition. Coarticulation effects present in continuous signing and absent from isolated signs make the segmentation of the former harder. Most of the existing work on the automatic segmentation of continuous signing relies on video footage to segment at a gloss-level. [KPBB02] take advantage of Hidden Markov Model (HMM) to segment a continuous stream in Korean Sign Language into signs segments. Similarly, [RS06] perform sign segmentation of continuous American Sign Language using Conditional Random Fields (CRF). In their article they demonstrate the superiority of the CRF approach (85% accuracy) compared to HMM (60%). A different approach was developed by [LADG08]. It presents a computer-aided segmentation of sign language sequences based on the detection of motion cues such as symmetry, repetition and hand trajectories templates. The algorithm is helped by the punctual intervention of an operator who has to specify one frame belonging to each sign.

However, those segmentation approaches do not take into account the multichannel aspect

of sign languages and lead to segmentation schemes highly dependent on the context of the utterance, i.e. the segments implicitly contain the coarticulation effects of the sequential signs. The resulting segmentation is thus hardly reusable in a different context, for example to synthesize new utterances. A lower-level segmentation, based in particular on phonological elements would facilitate sign composition in various contexts in order to produce new utterances. However, although several studies address the issue of the annotation of sign language video data at a gloss level, little attention has been given to the automatic annotation of the different linguistic channels of sign languages. The work of [Sto60] gives a phonological structure to signs by specifying three linguistic parameters to describe all signs : hand motion, hand configuration and hand placement. Each feature can take a discrete value in a limited vocabulary. Two complementary features were later added, hand orientation [Bat78] and non manual features such as shoulder tilt or facial expressions. Many phonological structures use those five features to define signs which can be used as a basis for video annotation : the segments are of a finer level than the gloss segmentation and retain a linguistic value. In an early work, [VM01] break signs into "phonemes" (close to the previous five features) and use HMM on the combination of the phonemes to recognize signs. The channels are processed separately but the ultimate purpose is gloss recognition and not channel annotation. In addition, this work is based on the Movement-Hold model which has been later replaced by a more precise phonetic model [JL11]. Furthermore, due to the difficulty of capturing the finger movements, the hand configuration channel was not processed and the authors of the article also chose not to deal with non manual features. More recently, [DYW$^+$14] propose to add some linguistic knowledge about the composition of lexical signs to considerably improve the performances of the recognition system.

Work on sign language MoCap data is scarcer than on video. For gloss-level segmentation, [NLG17] use some kinematic properties of the two wrists that can be extracted from MoCap data. At a finer level, [HGMC05] focused on the segmentation of hand configurations using Principal Composant Analysis (PCA) but the work is restricted to fingerspelling segmentation.

The recognition of facial expressions has received increasing attention in recent years, mainly from the computer vision community. Regarding the existing datasets, the availabiliy of 2D recording devices made possible the creation of large data collections, such as the Cohn-Kanade Dataset (posed facial expressions) and its extension [LCK$^+$10] (posed + non-posed facial expressions) or the MUG database [APD10b] (posed + non posed) with many (up to hundreds) actors. High resolution 3D facial databases with expressions have also been created using binocular/structured light cameras [ZYC$^+$14], [YCS$^+$08]. The frame rate of such optical device is often limited to 60 or less frames per second (usually 20/30 fps) which may be insufficient

for those who are interested in dynamic expressive variations. MoCap techniques can capture movements up to 200 fps and more, which makes them much more powerful for analyzing fine expressive variations. Nevertheless, the publicly available facial MoCap databases are still scarce. The multimodal database described in [BBL+08], which includes facial MoCap with speech and elicited emotional expressions is one of the few existing ones.

This remainder of the chapter presents the automatic annotation of continuous signing in French Sign Language using MoCap data on the facial expressions channel.

4.3 Input Data

This section describes the definition and recording of sign language data as well as the specification of the corpus that has been used for the studies.

4.3.1 General Considerations

To develop models for automatic annotation, the first solution that might come to mind is to record all the existing signs in all the possible contexts in order to cover exhaustively all the possible cases of sign production. While this can be attempted (with varying degrees of success) for oral languages by, for example, retrieving huge databases from the Internet (e.g. Wikipedia pages, Twitter posts, etc.), this is impossible for sign languages. Indeed, (i) sign language databases are scarce, especially MoCap databases, (ii) sign languages use the 3D space and the temporal dimension which leads to the production of an infinite number of combinations of the different physical channels, and (iii) sign language cannot be limited to their standard, reference signs : many sign language mechanisms such as classifier-predicates are as important as standard signs and depend strongly on the context of the sentence.

Instead of collecting a large set of random data, it might be more relevant to design a corpus specifically suited to the studied problem. One way to reduce the complexity of the capture is to consider each chosen channel separately. Indeed, each channel can display a limited number of different behaviors. For example, we can enumerate a limited number of different facial expressions in sign languages. As a consequence, a corpus designed to study and automatically annotate the affect would focus only on a small number of emotions to cover all the possible expressions.

To sum up, for the application of automatic annotation, a corpus containing many repetitions of a limited number of different occurrences of the studied element will be preferred. The

variability induced by a different context in the element production (for example, by capturing the same facial expression in different sentences) or by a different signer, must be recorded in order to improve the generalization capacity of the resulting model.

4.3.2 Corpus

A novel corpus, *FEeL*, has been specifically developed to study and synthesize the facial expressions.

Characteristics

The *FEeL* corpus has been captured using two signers (learner level). Three kinds of sequences - corresponding to different sets of instructions given to the signers - were recorded. We worked exclusively on the affect channel of the face and chose to analyze this channel within the Ekman framework with six categorical classes of basic emotions (i.e. *anger* - *A*, *disgust* - *D*, *fear* - *F*, *joy* - *J*, *sadness* - *Sa*, *surprise* - *Su* and *neutral* - *N*), which was easier to annotate and more understandable by humans than continuous models (e.g. the *Pleasure - Arousal - Dominance* framework). Three kinds of sequences were recorded :

i) *Isolated Expressions - IE :* the signers were asked to perform a given expression five times, each expression had to be maintained several seconds before returning to neutral (e.g. for joy we have : $N - J - N - J - N - J - N - J - N - J - N$). Six sequences were recorded per signer, one for each class of affect.

ii) *Sequences of Expressions - SE :* the signers were asked to alternate a given expression with each of the five other expressions (e.g. for joy : $N - J - Su - J - A - J - F - J - Sa - J - D - J - N$). Five sequences were recorded per signer.

iii) *Expressive Utterances - EU :* Sign language sentences with emotional content were prepared. The signers were asked to repeat three times each sentence with a given affect (e.g. it was asked to the signer to sign the following sentence with disgust : "There is a spider on my pizza ! Yuk !"). 18 sequences were recorded per signer, one for each sentence.

The corpus has been recorded via a *Qualysis* MoCap system. A total of 40 facial markers were tracked at 200 fps.

Manual Annotation

Manual annotations are used as reference and training data for our automatic annotation. It is thus necessary to have a thorough annotation. The *FEeL* corpus has been annotated using

the ELAN software [Max17]. We focused our efforts on the affect channel and, so far, a single annotator has been involved in the process. This annotator has been instructed to "subjectively annotate what he saw" with respect to the two following rules : (i) we distinguish two kinds of segments : the *transition* segments where the class vary from a starting expression to an ending expression, and the stable segments where the class doesn't vary along time ; (ii) the name of a *transition* segment is the concatenation of the name of the starting class and of the name of the ending class (e.g. *NA* means that the transition come from the *neutral* class to the *anger* class). A stable segment is named according to the maintained expression displayed (e.g. *Sa* stands for *sadness*).

4.4 Automatic Annotation

This section describes the principal steps to automatically annotate the affect channel. Motion descriptors are first computed in order to segment and then label the sign language data.

4.4.1 Facial Descriptors

The raw data collected from motion capture is the vector of the $3D$ positions of the body markers along time and might not be the best representation to study the facial expressions. Indeed, it is often required to transform the initial data in order to get a descriptor that depends only on the phenomenon that we intend to analyze. For instance, if the system is supposed to recognize facial expressions, it should not be sensible to morphological differences between the signers.

A common approach to animate facial expressions of a virtual character is the blendshapes method. An expression $Expr$ can be expressed as the sum of the mesh B_0 representing the neutral expression and a weighted linear combination of n basic deformations b_i expressed differentially from the neutral expression (see also fig. 4.2) :

$$Expr = B_0 + \sum_{i=1}^{n} w_i \cdot b_i \tag{4.1}$$

This method has the advantage of providing a light representation (in our case only 51 basic deformations) which leads to faster computations and facilitates storing in our database. In order to automatically obtain the appropriate set of parameters $\{w_{1..n}\}$ at each frame we have to face two problems : i) the targeted avatar and the signer don't have exactly the same morphology

(the *retargeting* problem), ii) for one given expression E there might be multiple existing linear combinations $\sum_{i=1}^{n} w_i \cdot b_i$ that minimize the distances between the markers and the corresponding vertices of the mesh (the *non unicity* problem).

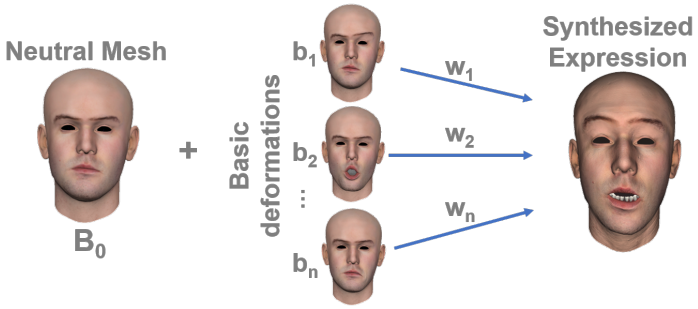


FIGURE 4.2 – Synthesizing expressions from a linear combination of blendshapes.

We dealt with the *retargeting* problem as in [**?**] or [**?**]. Given one frame where the signer shows a neutral expression, a RBF regression is trained in order to make the link between the position of any point of the signer's face and the position it would have on the avatar's face :

$$\hat{M} = F_{RBF}(M) = \sum_{k=1}^{K} u_k f_k(M) \tag{4.2}$$

where $\hat{M}$ is the estimated position of the signer's corresponding marker M retargeted on the avatar's mesh and $\{u_{1..K}\}$ are the optimized weights associated to the radial basis functions $\{f_{1..K}\}$.

The *non unicity* problem is formulated as a minimization problem in which the parameters $\{w_{1..n}\}$ are optimized so that the distances between the retargeted marker positions $\hat{M}_{1..40}$ and the corresponding vertices of the mesh $V_{1..40}$ are reduced. To ensure that the optimal weights found with this method do not generate visual artifacts, some constraints (e.g. non-negativity constraint) and/or some regularization energy that penalizes weights outside the $[0, 1]$ range can be incorporated. In our case, we introduced the *Thin-shell* model [BS08] as a regularization energy that doesn't directly ensure that the weights stay between 0 and 1 but penalizes the bending and stretching deformations of the initial mesh :

$$\hat{W} = \underset{\{w_{1..K}\}}{\operatorname{argmin}}(dist_{eucl}(\hat{M}_{1..40}, V_{1..40})^2 + E_{TS})c \tag{4.3}$$

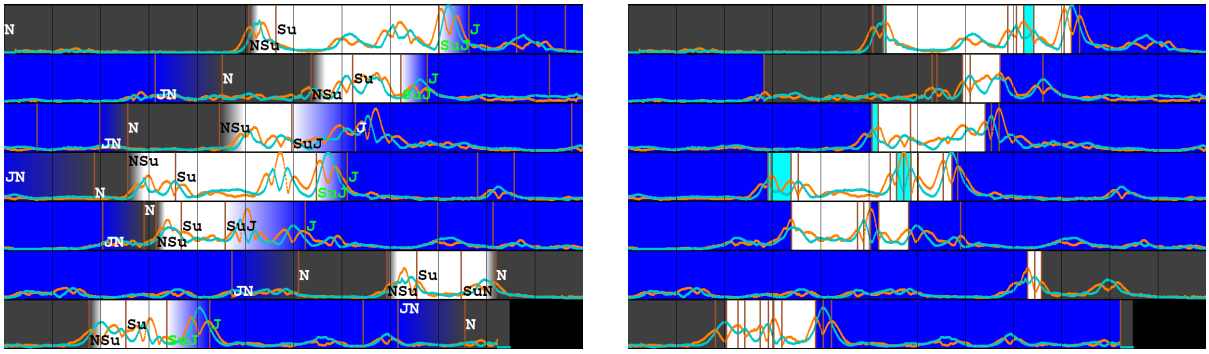
with $\hat{W}$ the optimal vector of weights $\{w_{1..K}\}$ for the given expression and E_{TS} the *Thin-shell* energy. The vector of blendshape weights $\hat{W}$ is the chosen descriptor for the analysis of the

affect channel.

4.4.2 Automatic Segmentation

The localization of segments of interest in a stream of sign language data is called the *segmentation*. It is done by detecting manually or automatically the temporal points corresponding to the beginning and the end of a behaviour (hand configuration or facial expression in our case). The coarseness of the behaviour to detect depends on the chosen annotation scheme. For example, sign language data can be segmented at a gloss level by detecting the beginning and end of a sign, or at a finer level such as facial expression, by detecting the beginning and end of an affect.

Segmentation of the Affect Channel



Manually segmented and labeled sequence Automatically segmented and labeled sequence

FIGURE 4.3 – An example of automatic annotation : "What ? I won 1000 € !" repeated 3 times (white : surprise, blue : joy, cyan : fear ; the orange curve stands for the sum of accelerations, the turquoise one for the sum of velocities ; the vertical brown lines represent the limits of each segment ; each vertical black line stands for 0.5 second).

The affect channel is segmented in a similar way, before the labeling step . We aim at detecting the frames that are located at the border between two segments. Since the border frames are related to the transitions from one expression to another we consider the energy curves of the velocity and acceleration of the n blendshape coefficients :

$$E_{VelBS} = \sum_{i=1}^{n} |\frac{dw_i}{dt}| \tag{4.4}$$

$$E_{AccBS} = \sum_{i=1}^{n} |\frac{d^2 w_i}{dt^2}| \tag{4.5}$$

In order to detect the local peaks, we consider the local optima of the curve $E^2_{Vel_{BS}} + E^2_{Acc_{BS}}$, and only keep those for which the local variation is important. This detection procedure is achieved by computing the derivative values on a window surrounding the detected optima, and applying a threshold. The orange and turquoise curves in Figure 4.3 show an example of segmentation using this method.

4.4.3 Automatic Labeling of the Affect Channel

The identification of the previously defined segments of interest is called the *labeling*. This task is highly dependent on the chosen annotation scheme. Typically, it will consists in selecting the right label from a closed vocabulary to identify a segment.

The facial channel labeling is a supervised learning task aiming at identifying the correct class among the 7 defined in section 4.3.2. Different methods were tested : kNN (*1NN* and *3NN*), SVM (*linear* and *RBF* kernels) and Random Forests (RF). The sequences recorded on each signer were processed separately. For each signer, the sequences IE and SE which represent roughly 50% of the data were used as the training set while the EU sequences were used as the test set. Whereas during the training phase, each frame of the training set with its corresponding manually annotated label was considered as a training sample, the test examples were constituted of the average along time of the frames composing each segment. These segments have been previously obtained according to the method presented in section 4.4.2. Each of these segments was classified as a whole represented by its average vector of blendshape weights :

$$\bar{\hat{W}} = \frac{\sum_{f=1}^{F} \hat{W}_f}{F} \tag{4.6}$$

with F the number of *frames* of the considered segment. Figure 4.3 shows an example of classification using this method ; results are detailed in next section.

4.5 Results

The results of the automatic annotation of the facial expression channel are presented in this section.

The sequences are first segmented according to the methods described in section 4.4.2. Each segment is then labeled independently of the others, according to the following procedure. For each segment, we compute the average vector descriptor over time : $\bar{\hat{W}}$. This vector is used as input of the classification model (kNN or SVM or Random Forest) previously trained on the basis of the learning set. The classifier then returns the label associated with this segment. In order to evaluate the error due to the segmentation, the automatic annotation is performed on both the automatically detected segments and the manually defined ones. For both segmentations, Figure 4.4 gives the accuracy of the classifier for each tested algorithm. It shows that the best results are obtained with the Random Forest algorithm.

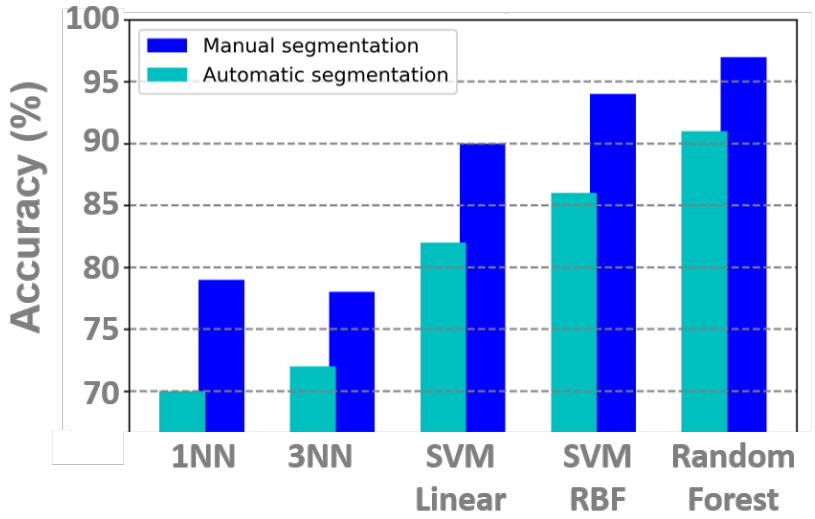


FIGURE 4.4 – Accuracy on the test set for the affect channel depending on the machine learning algorithms and the segmentation.

4.6 Conclusion

We designed an approach to automatically annotate the facial expression channel of sign language MoCap data.

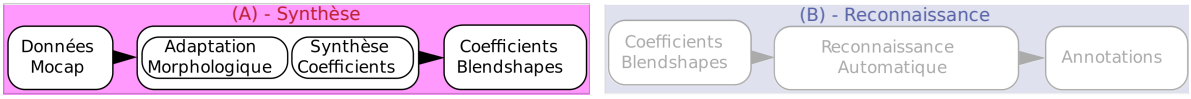
There are still many challenges to overcome. Using machine learning models, the automatic annotation could be significantly improved by increasing the size of the dataset, so that the training phase would be more efficient. In addition, following the approaches developed in language processing, we could also use models that learn the dynamics of the sequences, such as Hidden Markov Models, Conditional Random Fields, or Recurrent Neural Networks. However, these methods require large databases.

Another challenge concerns the evaluation of the annotation results. Indeed, for the manual annotation, we rely on a ground truth which may be subject to errors or imprecision. This problem occurs for most recognition or annotation tasks. One solution could be to define a ground truth from a set of previously trained annotators, following strict instructions. In the near future, we plan to validate our annotations by defining quantitative metrics. As a complement to assess the quality of the annotation, we also plan to perceptually evaluate the results.

CHAPITRE 5

ANIMATION FACIALE À PARTIR DE DONNÉES CAPTURÉES

Ce chapitre propose une solution au problème de la synthèse d'animations faciales à partir de données issues de la *motion capture* basée marqueurs. La solution proposée utilise la représentation par blendshapes commune à la représentation utilisée pour l'annotation automatique (voir figure 5). Les résultats obtenus sont validés par des études perceptuelles.

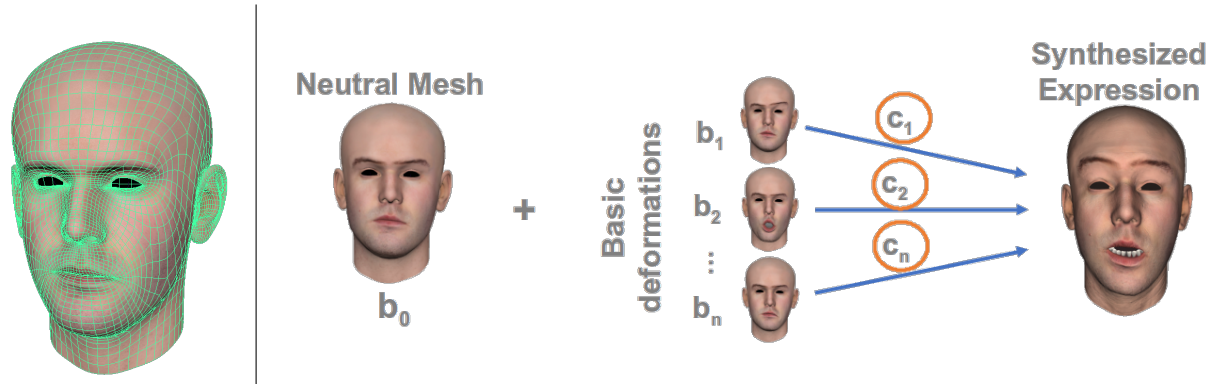


5.1 Généralités

5.1.1 Représentation et animation par blendshapes

En modélisation et animation $3D$, un objet est en général représenté par un maillage de sommets ou graphe G (illustration en figure 5.1) reliés entre eux par des arêtes formant des polygones (le plus souvent des quadrilatères et triangles), ce maillage peut-être numériquement représenté à un instant (ou *frame*) f donné par le couple (v_f, Adj) où v_f de dimension $(3 \times S)$ est le vecteur des positions de sommets, S est le nombre de sommets du maillage et Adj est la matrice d'adjacence des sommets ($Adj[i,i] = 0$). Animer un maillage consiste à modifier les positions des sommets du maillage au cours d'une séquence de *frames*, nous notons V la matrice de dimensions $(F, 3 \times S)$ décrivant les positions des S sommets au cours des F frames d'une séquence donnée, la matrice d'adjacence Adj de dimensions (S, S) ne change pas au cours du temps :

$$Adj[i,j] = \begin{cases} 1, \text{ si le } i^{\text{ème}} \text{ et le } j^{\text{ème}} \text{ sommets sont adjacents} \\ 0, \text{ sinon} \end{cases} \tag{5.1}$$



(a) (b)

FIGURE 5.1 – Maillage (a) et animation basée blendshapes (b).

Une méthode communément employée pour représenter et animer les visages de personnages virtuels est celle dite des *blendshapes* (illustration en figure 5.1). Ce type de représentation est composé d'un maillage représentant le visage lorsqu'il affiche une expression neutre et un ensemble d'expressions de base représentant des déformations unitaires de ce maillage (exprimées différentiellement par rapport au maillage neutre). Le nombre de bases peut varier d'un avatar à l'autre (allant d'une quinzaine de bases jusqu'à un peu plus d'une centaine suivant les cas) en fonction du degré de liberté souhaité. Nos avatars ont chacun $N = 51$ bases inspirées des *Action Units* du système *FACS* d'Ekman [EF78]. Numériquement, les positions des sommets du maillage (affichant une expression neutre) sont notées b_0 de dimension $(3 \times S)$ et les positions de ces sommets notées différentiellement par rapport à b_0 pour chacune des N bases choisies sont notées $b_{i \in \{1..N\}}$. Une expression faciale v_f quelconque est représentée comme étant des positions des sommets pour l'expression neutre b_0 et une combinaison linéaire des bases qui lui sont associées :

$$v_f = b_0 + \sum_{i=1}^{N} c_f[i] b_i \tag{5.2}$$

où les coefficients $c_f[i]$ sont les paramêtres permettant d'animer le maillage. Nous notons B la matrice de dimensions $(N + 1, 3 \times S)$ contenant le vecteur $B[0, :]$ décrivant les positions de sommets de l'expression neutre b_0 et les vecteurs $B[i, :]$ correspondant à chacune des N bases b_i ainsi que C la matrice de dimensions $(F, N + 1)$ avec $C[f, 0] = 1$ et $C[f, i] = c_f[i]$ (pour $i \in [1..N]$). Pour une séquence de *frames* données V, l'équation précédente peut donc se réécrire sous la forme du produit matriciel :

$$V = C \cdot B \tag{5.3}$$

En résumé nous notons :

— F : le nombre de *frames* d'une d'une séquence donnée.
— S : le nombre de sommets d'un maillage donné
— N : le nombre de bases d'un modèle blendshapes
— V : la matrice de dimension $(F, 3\times S)$ décrivant les positions des sommets d'un maillage au cours du temps
— ${}^1v_f = V[f, :]$: le vecteur de dimension $(3 \times S)$ des positions des sommets pour la $f^{\text{ème}}$ *frame*
— ${}^1v_i = V[k, 3i : 3i + 3]$: le vecteur de dimension (3) donnant la position du $i^{\text{ème}}$ sommet à un instant quelconque et fixe k
— Adj : la matrice de dimensions (S, S) d'adjacence d'un maillage
— G : un maillage 3D ou couple (v, Adj) composé d'un ensemble de position de sommets (dimension $(3 \times S)$) et d'une matrice d'adjacence de ces sommets (dimensions (S, S))
— B : la matrice de dimensions $(N, 3 \times S)$ décrivant les différences de position de chaque sommet du maillage par rapport à l'expression neutre pour chacune des bases
— $b_i = B[i, :]$: le vecteur de dimension $(3 \times S)$ décrivant les différences de position de chaque sommet pour la $i^{\text{ème}}$ base
— C : la matrice de dimensions (F, N) décrivant les coefficients associés à chaque base blendshape au cours du temps
— $c_f = C[f, :]$: le vecteur de coefficients pour la $f^{\text{ème}}$ *frame*

5.1.2 De l'opérateur Laplacien à l'énergie de déformation *Thin-Shell*

L'opérateur Laplacien se retrouve dans une myriade de domaines différents, en particulier en physique, mathématiques et informatique. L'application de cet outil mathématique et son adaptation aux surfaces discrètes tridimensionnelles ont fait l'objet de plusieurs études dont celles de Sorkine et al. [SCOL$^+$04, Sor05]. Le Naour et al. [LN13, NCG19] utilisent cet opérateur pour reconstruire des trajectoires tridimensionnelles ou contrôler la déformation de maillages guidée par des marqueurs mocap.

1. En cas de possible confusion concernant la signification de l'indice associé à v_i ou v_f la notation précise $V[f, 3i : 3i + 3]$ sera employée.

Définition de l'opérateur Laplacien continu

L'opérateur Laplacien Δ est un opérateur mathématique défini comme étant la divergence du gradient.

Le gradient ∇ est défini comme l'opérateur qui, à une fonction scalaire $f : \mathbb{R}^n \to \mathbb{R}$, associe le vecteur des dérivées partielles de cette dernière :

$$\nabla f = (\frac{\partial f}{\partial x_0}, \cdots, \frac{\partial f}{\partial x_{n-1}}) \tag{5.4}$$

La divergence div est quant à elle définie comme étant l'opérateur qui, à une fonction scalaire $f : \mathbb{R}^n \to \mathbb{R}$, associe la somme de ses dérivées partielles :

$$div(f) = \sum_{i=0}^{n} \frac{\partial f}{\partial x_i} \tag{5.5}$$

L'opérateur Laplacien est alors défini comme l'opérateur qui, à une fonction scalaire deux fois dérivable $f : \mathbb{R}^n \to \mathbb{R}$, associe :

$$\Delta f = div(\nabla f) = \sum_{i=0}^{n} \frac{\partial^2 f}{\partial x_i^2} \tag{5.6}$$

Par l'utilisation de la méthode des différences finies on remarque que pour h tendant vers 0 :

$$\Delta f = \sum_{i=0}^{n} \frac{\partial^2 f}{\partial x_i^2} \approx \sum_{i=0}^{n} \frac{2}{h^2}(\frac{f(x_i + h) - f(x_i - h)}{2} - f(x_i)) \tag{5.7}$$

Pour chaque dimension de x, $(\frac{f(x_i+h)-f(x_i-h)}{2} - f(x_i))$ représente l'écart entre la moyenne de f au voisinage de x et la valeur de f en x. Cette observation permet d'intuiter la généralisation de l'opérateur laplacien aux maillages tridimensionnels.

Adaptation du Laplacien aux surfaces discrètes tridimensionnelles

Ainsi, l'opérateur Laplacien peut-être étendu aux graphes et en particulier aux maillages tridimensionnels tels que décrits précédemment. Prenons un maillage quelconque $G = (v, Adj)$, le Laplacien d'une fonction $f : \mathbb{R}^3 \to \mathbb{R}$ prenant en entrée un sommet quelconque peut-être défini comme la somme pondérée des écarts entre les voisins de ce sommet et sa position réelle :

$$\Delta f(v_i) = w_{\text{vert}}[i] \sum_{j=0}^{S} Adj[i,j] W_{\text{edges}}[i,j](f(v_i) - f(v_i)) \tag{5.8}$$

Où $w_{\text{vert}}[i]$ est le vecteur de dimension (S) des poids de normalisation associés à chaque sommet et W_{edges} est la matrice de dimension (S, S) des poids de normalisation associés à chaque arête du maillage ($W_{\text{edges}}[i, i] = 0$). Différents types de normalisation sont possibles, comme par exemple :

— les poids uniformes $W_{\text{edges}}[i, j] = 1$, $w_{\text{vert}}[i] = 1$

— les poids co-tangents $W_{\text{edges}}[i, j] = \frac{1}{2}(cot(\alpha_{i,j}) + cot(\beta_{i,j}))$, $w_{\text{vert}}[i] = \frac{1}{vor_i}$ où vor_i est l'aire de Voronoï autour du $i^{\text{ème}}$ sommet (voir figure 5.2)

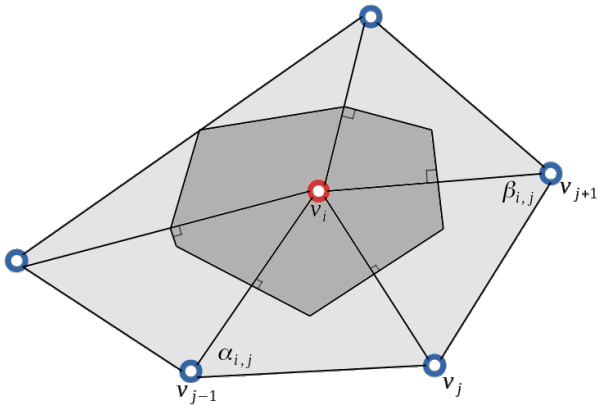


FIGURE 5.2 – L'aire de voronoï (gris foncé) autour du $i^{\text{ème}}$ sommet est construite en connectant pour chaque triangle autour de ce sommet les points médians des deux côtés adjacents et, pour les angles aigus, l'intersection des bissectrices de ces deux côtés ; pour les angles obtus, le point médian du côté opposé. Les angles $\alpha_{i,j}$ et $\beta_{i,j}$ sont les angles opposés au segment reliant les $i^{\text{ème}}$ et $j^{\text{ème}}$ sommets.

Les Laplaciens d'ordre supérieurs Δ^k sont définis récursivement [BS08] :

$$\Delta^k f(v_i) = w_{\text{vert}}[i] \sum_{j=0}^{S} Adj[i, j] w_{\text{edges}}[i, j](\Delta^{k-1} f(v_j) - \Delta^{k-1} f(v_i)) \tag{5.9}$$

$$\Delta^0 f(v_i) = f(v_i) \tag{5.10}$$

L'opérateur Laplacien Δ pour un maillage V et une fonction $f : \mathbb{R}^n \to \mathbb{R}$ donnée peut s'écrire sous forme matricielle [BS08] :

$$\begin{pmatrix} \Delta f(v_1) \\ \vdots \\ \Delta f(v_S) \end{pmatrix} = W_{vert} \cdot L_{\mathcal{L}} \cdot \begin{pmatrix} f(v_1) \\ \vdots \\ f(v_S) \end{pmatrix} \tag{5.11}$$

Avec W_{vert} : la matrice diagonale de dimensions (S, S) contenant les poids de normalisation $W_{vert}[i, i] = w_{vert}[i]$, et L : la matrice symétrique de dimension (S, S) :

$$L[i,j] = \begin{cases} -\sum_{k=0}^{S} Adj[i,k] W_{\text{edges}}[i,k] & \text{, si i=j} \\ Adj[i,j] W_{\text{edges}}[i,j] & \text{, sinon} \end{cases} \tag{5.12}$$

Modèle *thin-shell*

Prenons deux surfaces $\mathcal{S}_0$ et $\mathcal{S}_1$ définies par deux fonctions $p_0 : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{S}_0 \subset \mathbb{R}^3$ et $p_1 : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{S}_0 \subset \mathbb{R}^3$. Nous cherchons des propriétés de ces surfaces dont les différences d'une surface à l'autre (par exemple de $\mathcal{S}_0$ à $\mathcal{S}_1$) permettent de rendre compte de la "quantité de déformation" à appliquer à la première pour obtenir la seconde. La géométrie différentielle définit la première forme fondamentale $I : \mathbb{R}^2 \rightarrow \mathbb{R}^{2\times2}$ permettant d'obtenir des informations sur la longueur et l'aire d'une surface, et la seconde forme fondamentale $II : \mathbb{R}^2 \rightarrow \mathbb{R}^{2\times2}$ [GUE15][doC76] permettant d'obtenir des informations sur la courbure d'une surface. Ces deux formes fondamentales sont donc de bonnes candidates pour mesurer les dissimilarités entre deux surfaces, pour nos deux surfaces $\mathcal{S}_0$ et $\mathcal{S}_1$ dont les formes fondamentales sont respectivement notées I_0, II_0 et I_1, II_1. Cette mesure est alors définie ainsi :

$$E_{ts}(\mathcal{S}_0, \mathcal{S}_1) = \int_\Omega k_s ||I_1(u,v) - I_0(u,v)||_F^2 + k_b ||II_1(u,v) - II_0(u,v)||_F^2 \, \mathrm{d}u\mathrm{d}v \tag{5.13}$$

où k_s et k_b représentent respectivement les paramètres de résistance à l'étirement, et de résistance à la courbure de la surface. $||\cdot||_F$ représente la norme de Frobenius. Soit $d_{01} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ la fonction de déplacement ayant le même espace de définition que $\mathcal{S}_0$ et $\mathcal{S}_1$ et qui, à chaque point de cet espace, associe la différence entre $\mathcal{S}_0$ et $\mathcal{S}_1$:

$$d_{01}(u,v) = p_1(u,v) - p_0(u,v) \tag{5.14}$$

Dans [BS08], Botsch et Sorkine expliquent que pour l'équation 5.13, la première et la seconde formes fondamentales peuvent être approchées par les dérivées partielles de premier et de second ordre de la fonction de déplacement d_{01} :

$$E_{ts}(d_{01}) = \int_\Omega k_s \left(||\frac{\partial d_{01}}{\partial u}||^2 + ||\frac{\partial d_{01}}{\partial v}||^2 \right) + k_b \left(||\frac{\partial^2 d_{01}}{\partial u \partial u}||^2 + 2||\frac{\partial^2 d_{01}}{\partial u \partial v}||^2 + ||\frac{\partial^2 d_{01}}{\partial v \partial v}||^2 \right) du dv \tag{5.15}$$

Dans la section 5.2.3, nous chercherons à calculer efficacement cette énergie E_{ts} et à l'adapter à la représentation par blendshapes afin de l'intégrer comme énergie de régularisation pour

le processus d'optimisation. L'équation des dérivées partielles d'Euler-Lagrange permet de caractériser le minimiseur de cette équation [BS08] :

$$-k_s \Delta d_{01} + k_b \Delta^2 d_{01} = 0 \tag{5.16}$$

En considérant pour chacune des composantes x, y et z de d_{01} et en reprenant l'équation 5.11 avec $f = d_{01}^{x/y/z}$, nous en arrivons à l'équation suivante pour chacune de ces composantes :

$$(-k_s \mathcal{L} + k_b \mathcal{L}) d_{01}^{x/y/z} = 0 \tag{5.17}$$

5.2 Animation basée données

La première question qui s'est posée est celle de la disposition des marqueurs sur le visage de l'acteur. En particulier, deux questions se posent : i) combien de marqueurs sont nécessaires pour reconstituer efficacement une expression faciale ? ii) Où placer ces marqueurs ? Nous émettons l'hypothèse que tous les points de la surface du visage ne portent pas la même quantité d'information et que certains points (notamment ceux situés dans une même région) portent une information redondante les uns par rapport aux autres.

Il convient également de garder en vue l'aspect pratique de la pose de marqueurs. En effet, la pose de marqueurs est intrusive pour l'acteur et la pose de marqueurs sur des régions telles que les paupières ou le contour des yeux peut s'avérer génante. De plus, certains points de la surface du visage peuvent être plus compliqués à retrouver que d'autres d'une séance à l'autre et d'un acteur à l'autre. Enfin, plus le nombre de marqueurs à placer est important plus le risque de mauvais placement et la gêne pour l'acteur seront importants.

Au chapitre 3, nous avons établi qu'une quarantaine de marqueurs serait amplement suffisant et qu'en utiliser davantage n'apporterait pas de gain significatif en terme d'erreur de reconstitution. Au moins quatre marqueurs supplémentaires (six préférérablement) sont à prévoir afin de déterminer les transformations de rotations et translations du visage et appliquer leur inverse afin de ne garder que les transformations liées aux expressions faciales. Le jeu de marqueurs utilisé est présenté en figure 5.3. La section 5.2.1 décrit l'étape de post-traitement des données capturées qui doit être effectuée afin de rendre les données utilisables.

L'objectif ensuite est de déterminer les coefficients de blendshapes c_i de l'animation. Deux problèmes sont alors à considérer :

— i) l'avatar ne présente pas une morphologie exactement similaire à celle de l'acteur. Il

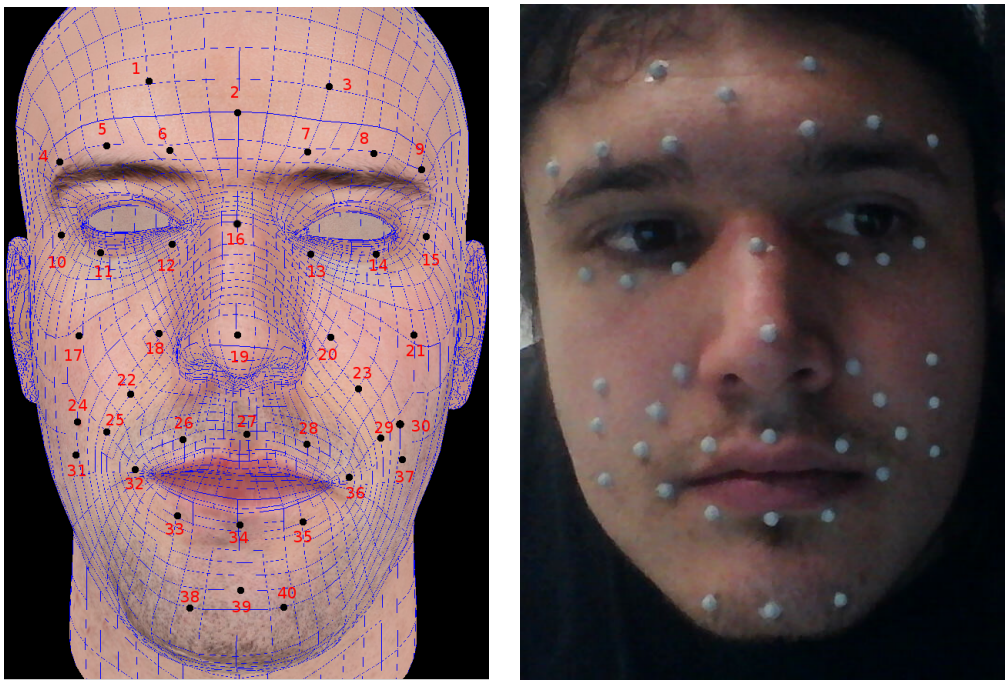


FIGURE 5.3 – Le jeu de marqueurs utilisé comporte 40 marqueurs.

faut donc faire une adaptation morphologique des données de l'acteur à la forme du visage de l'avatar. Cette étape est décrite dans la section 5.2.2 ;

— ii) comment déterminer automatiquement les paramètres de l'animation à partir des positions des marqueurs. La solution proposée est détaillée en section 5.2.3.

5.2.1 Post-traitement des données

Les données brutes issues de la capture d'une séquence consistent en une matrice de dimensions $(F, 3 \times (P + Q))$ représentant les trajectoires en $3D$ de $P + Q$ marqueurs au cours des F *frames* que dure la séquence. Q représente un jeu de marqueurs additionnels dont le rôle est de calculer les transformation rigides (translations et rotations) via la méthode de [SHR17] afin d'appliquer les transformations inverses et ne conserver que les déformations non rigides liées aux expressions faciales. $Q = 4$ est suffisant, pour plus de sureté il est intéressant d'en avoir $Q = 6$, les positions de ces marqueurs les uns par rapport aux autres doivent être fixe (les positionner sur le crane ou les tempes). Une fois ces transformations calculées, ces Q marqueurs ne sont plus utilisées. Par ailleurs, un ensemble de P sommets du maillage situés de façon similaire aux positions des marqueurs sur le visage de l'acteur est déterminé manuellement. Le principal post-traitement consiste à supprimer les déplacements de marqueurs liés aux déformations ri-

gides (e.g., rotation et translation de la tête). Pour se faire, la méthode proposée dans [SHR17] a été employée en deux temps.

Alignement sur une *frame*

Dans un premier temps, on considère les nuages de P_m points, correspondant aux positions des marqueurs, obtenus sur une *frame* où l'acteur effectue une expression neutre, et les positions des P_v sommets correspondants sur le maillage. La méthode proposée dans [SHR17] permet de déterminer le couple rotation/translation alignant au mieux les deux nuages de points P_m et P_v et de l'appliquer à chaque marqueur de chaque *frame* afin d'obtenir un nuage de points P_m aligné par rapport au maillage de l'avatar. Ce premier alignement ne peut et n'a pas à être rigoureusement précis. Avoir ces deux nuages de points relativement alignés ne sert qu'à rendre l'adaptation morphologique plus robuste (voir section 5.2.2). Une fois cette opération effectuée, le maillage et les marqueurs sont alignés pour une *frame* donnée. La seconde étape consistera donc à aligner les nuages de points sur l'ensemble de la séquence avec celle sur laquelle le premier alignement a été effectué.

Alignement sur la séquence

Cette fois-ci seuls les Q marqueurs sont utilisés afin de déterminer les couples (rotation / translation) pour chaque *frame*. On considère que les positions de chacun de ces marqueurs sont fixes les unes par rapport aux autres contrairement aux autres marqueurs dont les positions relatives les uns par rapport aux autres bougent en fonction des expressions effectuées par l'acteur. Les couples (rotation/translation) sont ensuite appliqués à l'ensemble des positions de marqueur de chaque *frame*.

Par commodité, on se référera par la suite à M comme étant la matrice de dimension $(F, 3 \times P)$ des trajectoires des P marqueurs faciaux au cours de F *frames* d'une séquence donnée après avoir effectué ces traitements.

5.2.2 Adaptation morphologique

Sauf dans le cas où l'avatar a été crée à partir d'une capture du visage de l'acteur [fac12], l'acteur et l'avatar ne partagent pas la même morphologie et il est nécessaire d'adapter les trajectoires enregistrées sur l'acteur afin qu'elles correspondent à la morphologie de l'avatar ciblé.

La matrice de dimension $(F, 3 \times P)$ contenant les positions $3D$ de P marqueurs au cours d'une séquence de F *frames* est notée M. Les trajectoires adaptées à la morphologie de l'avatar sont notées $\hat{M}$.

Pour résoudre ce problème d'adaptation, il a été choisi d'effectuer une régression ***RBF*** (Radial Basis Functions) [NN01, BBA+07a, SLS+12a]. On définit trois fonctions - définies sur $\mathbb{R}^3 \mapsto \mathbb{R}$ F^x, F^y et F^z - prenant chacune en entrée les coordonnées d'un point quelconque m situé sur la surface du visage de l'acteur et renvoyant l'une des trois coordonnées x, y et z de $\hat{m}$ la position qu'aurait ce marqueur sur le visage de l'avatar ciblé :

$$F(m) = \sum_{k=0}^{P} (u_k ||m - M_0[k]||^3) + q(m) \tag{5.18}$$

avec q un polynôme trivarié de la forme $q(x, y, z) = ax + bx + cx + d$ permettant de tenir compte des transformations affines et M_0 la position des marqueurs sur le visage de l'acteur lorsque celui-ci affiche une expression neutre. Le choix de la fonction radiale de type tri-harmoniques (de la forme $\phi(r) = r^3$) permet d'avoir une fonction C^2 continue [BBA+07a].

On cherche à déterminer les paramètres $u_{k=1..P}$ ainsi que le quadruplet a, b, c, d de cette fonction. On considère le vecteur $\hat{M}_0$ des positions des sommets correspondants aux marqueurs lorsque l'avatar affiche une expression neutre (i.e. quand tous les coefficients de *blendshapes* sont à 0) ce qui débouche sur l'équation suivante (ici pour F^x) :

$$\begin{pmatrix} \phi[0,0] & \phi[0,1] & \cdots & \phi[0,P-1] & M_0^x[0] & M_0^y[0] & M_0^z[0] & 1 \\ \phi[1,0] & \phi[1,1] & \cdots & \phi[1,P-1] & M_0^x[1] & M_0^y[1] & M_0^z[1] & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi[P-1,0] & \phi[P-1,1] & \cdots & \phi[P-1,P-1] & M_0^x[P-1] & M_0^y[P-1] & M_0^z[P-1] & 1 \\ M_0^x[0] & M_0^x[1] & \cdots & M_0^x[P-1] & 0 & 0 & 0 & 0 \\ M_0^y[0] & M_0^y[1] & \cdots & M_0^y[P-1] & 0 & 0 & 0 & 0 \\ M_0^z[0] & M_0^z[1] & \cdots & M_0^z[P-1] & 0 & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ \cdots \\ u_{P-1} \\ a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \hat{M}_0^x[0] \\ \cdots \\ \hat{M}_0^x[P-1] \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{5.19}$$

qui se trouve être de la forme :

$$A \cdot U = B \tag{5.20}$$

De plus, A est carrée et inversible si bien que les inconnues U peuvent être résolues comme un système linéaire classique. Une fois les paramètres optimaux déterminés pour F^x, F^y et F^z, ces fonctions peuvent être utilisées pour reconstituer les trajectoires des marqueurs sur l'avatar ciblé.

5.2.3 Détermination des coefficients par optimisation

Une fois l'adaptation morphologique effectuée il devient possible de déterminer automatiquement les paramètres de l'animation de l'avatar (c.à.d. le vecteur des coefficients de *blendshapes*) pour une séquence donnée.

Idée générale

Soit $\hat{M}$ la matrice de dimension $(F, 3 \times P)$ des positions des P marqueurs adaptés à l'avatar au cours d'une séquences de longueur F *frames*. On appelle Dm la matrice de dimension $(F, 3 \times P)$ des différences entre les positions de marqueurs adaptées $\hat{M}$ et la positions des sommets du maillage de l'avatar associés à chacun des marqueurs (voir section 5.2.2) ; $\hat{M} = \hat{M}_0$ lorsque l'avatar affiche une expression neutre (coefficients de *blendshapes* définis à 0) :

$$Dm[f, 3p : 3p + 3] = \hat{M}[f, 3p : 3p + 3] - \hat{M}_0[3p : 3p + 3] \tag{5.21}$$

pour une *frame* f et un marqueur p donnés. On appelle Da_c le vecteur de dimension $(3 \times P)$ représentant, pour un vecteur de coefficients c quelconque, les déformations des sommets associés aux marqueurs :

$$Da_c[p : p + 3] = c \cdot B[1 : n + 1, 3 \times id(p) : 3 \times id(p) + 3] \tag{5.22}$$

avec B la matrice de dimension (n, s) contenant les coordonnées des s sommets du maillage pour l'expression neutre ($B[0, :]$) ainsi que les vecteurs de déformation de chacune des n bases ($B[1 : n, :]$), et $id(p)$ la fonction qui à un indice de marqueur p donné renvoie l'identifiant du sommet qui lui est associé.

Pour chaque *frame* f, nous cherchons le vecteur de coefficients de *blendshapes* optimal $\hat{c}$ tel que :

$$\hat{c} = \underset{c}{\operatorname{argmin}}(||Dm[f, :] - Da_c||) \tag{5.23}$$

Néanmoins il existe plusieurs solutions à ce problème et une solution "optimale numériquement" peut ne pas l'être "visuellement". En effet, les bases de *blendshapes* sont en général conçues pour que les coefficients qui leur sont associés soient compris dans l'intervalle $[0; 1]$. Dans le cas où cela n'est pas respecté, cela peut conduire à des artefacts visuels imprédictibles. Il est donc nécessaire d'incorporer à cette équation des énergies de régularisation afin de s'assurer que cette contrainte soit raisonnablement respectée.

Énergies de régularisation

Nous définissons deux énergies, E_b et E_{ts} :

— la première, E_b, oppose une résistance lorsque les coefficients de *blendshapes* sortent de l'intervalle $[0; 1]$;

— la seconde, E_{ts} découle du modèle *thin-shell* [BS08] (voir section 5.1.2) et tend à conserver la forme (étirement et courbure) du maillage original dans son expression neutre de façon à ce que les solutions privilégiées parmi celles minimisant $||Dm[f, :] - Da_c||$, qui ne tient compte que des sommets associés aux marqueurs, soient celles préservant au mieux la structure de l'intégralité du maillage.

E_b est décrite par l'équation suivante :

$$E_b = \sum_{b=1}^{n+1}(exp(\frac{l_{inf} - c[i-1]}{s})) + \sum_{b=1}^{n+1}(exp(\frac{c[i-1] - l_{sup}}{s})) \tag{5.24}$$

avec s un paramètre de lissage (ici $s = 0.1$), l_{inf} et l_{sup} les bornes respectivement inférieures et supérieures (ici $l_{inf} = -0.1$ et $l_{sup} = 1.1$).

E_{ts} est quant-à elle décrite par l'équation suivante (voir equation 5.17) :

$$E_{ts} = \underbrace{(-k_s\mathcal{L} + k_b\mathcal{L}^2)}_{A} D \tag{5.25}$$

avec $\mathcal{L}$ et $\mathcal{L}^2$ le Laplacien et le bi-laplacien du maillage, k_s et k_b les coefficients pondérant respectivement l'importance de l'étirement et de la courbure du maillage (ici $k_s = 1$ et $k_b = 1$), D le vecteur des déformations appliquées au maillage. L'idée générale pour intégrer cette énergie au *framework* des blendshapes est de remarquer que $D = B^T[:, 1 : n] \cdot c$ ce qui permet la réécriture :

$$E_{ts} = A \cdot B^T[:, 1 : n] \cdot c \tag{5.26}$$

En regroupant la partie invariable qui peut être calculée une et une seule fois pour toutes, $AB = A \cdot B^T[:, 1 : n]$, on obtient :

$$E_{ts} = AB \cdot c \tag{5.27}$$

Soit $ABnc$ le sous ensemble des lignes de AB correspondant aux sommets auxquels aucun marqueur n'est associé, l'énergie de régulation retenue est donc $Enc_{ts} = ABnc \cdot c$. Le problème de détermination de coefficients optimaux $\hat{c}$ se ramène donc au problème d'optimisation

suivant :

$$\hat{c} = \underset{c}{\operatorname{argmin}}(||Dm[f,:] - Da_c|| + \alpha_{ts} Enc_{ts} + \alpha_b E_b) \tag{5.28}$$

où α_{ts} et α_b sont deux coefficients permettant de pondérer les influences respectives de Enc_{ts} et E_b. Nous obtenons des résultats raisonnables tant que $\alpha_{ts} + \alpha_b = 1$. AB est identique quelle que soit la *frame* et n'a à être calculée qu'une seule fois. Ce problème d'optimisation peut donc être résolu via des méthodes classiques telles que la descente de gradient.

Détails concernant l'implémentation *thin-shell*

La mise en oeuvre de cette méthode, en particulier en ce qui concerne l'incorporation de l'énergie *thin-shell* mérite d'être davantage détaillée.

La méthode permettant d'obtenir la matrice A de dimension (S, S) est décrite dans la section 5.1.2. Pour des raisons de simplicité, la matrice AB a été précédemment définie par $AB = A \cdot B$ ce qui n'est pas exactement le cas. Dans les faits, la matrice AB est de dimension $(3 \times S, n)$ et est calculée ainsi :

$$AB = \begin{pmatrix} \begin{pmatrix} \\ \left[A \cdot (B^T[:,1])^{(S,3)}\right]^{(3\times S,1)} \\ \\ \end{pmatrix} & \begin{pmatrix} \\ \left[A \cdot (B^T[:,2])^{(S,3)}\right]^{(3\times S,1)} \\ \\ \end{pmatrix} & \begin{pmatrix} \\ \cdots \\ \\ \end{pmatrix} & \begin{pmatrix} \\ \left[A \cdot (B^T[:,n])^{(S,3)}\right]^{(3\times S,1)} \\ \\ \end{pmatrix} \end{pmatrix} \tag{5.29}$$

La notation $X^{(q,p)}$ signifie, pour une matrice X, qu'elle a été re-dimensionnée aux dimensions (q, p). On remarquera également que sur l'équation 5.17, les 3 composantes sont résolues séparément tandis qu'ici les 3 composantes sont résolues simultanément, d'où une matrice de dimension $(3 \times S, N)$ et non (S, N).

Par ailleurs on remarquera que $AB \cdot x$ est un vecteur tandis que E_b est un scalaire. Nous considérons donc en réalité :

$$\hat{c} = \underset{c}{\operatorname{argmin}}(||Dm[f,:] - Da_c|| + \alpha_{ts} Enc'_{ts} + \alpha'_b E_b) \tag{5.30}$$

Avec $Enc'_{ts} = Enc_{ts} \cdot Enc_{ts}$ et $\alpha'_b = 3S \times \alpha_b$.

5.3 Résultats

La méthode proposée donne des résultats visuellement cohérents (voir figures 5.4, 5.5 et 5.6) sans pour autant devoir donner des paramètres spécifiques à chaque séquence autre qu'une frame de référence où l'acteur affiche une expression neutre. Néanmoins en faisant varier la valeur de α_{ts} (l'importance de l'énergie de régulation thinshell), nous remarquons en première impression qu'en diminuant cette dernière les animations tendent à afficher davantage d'artefacts tandis qu'en l'augmentant les expressions paraissent moins intenses. Il convient donc de trouver un bon compromis concernant ce paramètre. Dans cette optique, deux études perceptuelles ont été menées (voir chapitre 6) afin de i. déterminer quel serait la valeur d'α_{ts} la plus adaptée et ii. comparer les animations produites par notre méthode avec une autre méthode de l'état de l'art.

FIGURE 5.4 – Séquences (1fps) cr_IGE1_anger3 (gauche) et cr_IGE1_joy2 (droite) générées avec $\alpha_{ts} = 0.3$.

FIGURE 5.5 – Séquences (1fps) cr_IGE1_anger3 (gauche) et cr_IGE1_joy2 (droite) générées avec $\alpha_{ts} = 0.5$.

FIGURE 5.6 – Séquences (1fps) cr_IGE1_anger3 (gauche) et cr_IGE1_joy2 (droite) générées avec $\alpha_{ts} = 0.7$.

CHAPITRE 6

VALIDATION DE LA MÉTHODE D'ANIMATION PAR DES ÉTUDES PERCEPTUELLES

La mise au point de la méthode présentée en section 5.2 nous a amené à nous poser deux questions :

— i) Quel est le meilleur paramétrage (coefficients α) pour cette méthode ?

— ii) Comment positionner cette méthode par rapport à l'état de l'art ?

Pour y répondre, deux évaluations perceptuelles ont été mises en place. La première, détaillée dans la section 6.3, vise à répondre à la première question. La seconde (section 6.4), divisée en deux parties (2a et 2b), a pour objectif de répondre à la seconde question. Les cinq hypothèses testées afin de répondre à ces deux questions concernent la reconnaissance des émotions et de leur intensité, la difficulté à reconnaître ces émotions, et la crédibilité et la fidélité des animations produites. Ces hypothèses sont présentées dans les tables 6.2 et 6.7.

6.1 Description du jeu de données

Un nouveau jeu de données a été capturé sur un seul acteur pour réaliser cette étude perceptuelle. Pour chaque séquence, il a été demandé à l'acteur d'effectuer la même expression parmi les six de notre vocabulaire (*C : Colère, D : Dégoût, P : Peur, J : Joie, T : Tristesse, S : Surprise* + *N : Neutre*) et de maintenir cette expression quelques secondes trois fois d'affilée avec trois degrés d'intensité différents (1 : faiblement marqué, 2 : marqué, 3 : exagérément marqué) chacun entrecoupé d'une à deux secondes d'expression neutre. Par exemple pour la colère, nous avons : *N-C1-N-C2-N-C3-N-...-N-C1-N-C2-N-C3-N*.

Chaque séquence à été capturée simultanément via deux dispositifs différents : du matériel de *motion capture* basée marqueurs, et une caméra RGB-D (voir figure 6.1). Un même avatar a ensuite été animé à partir des données recueillies,

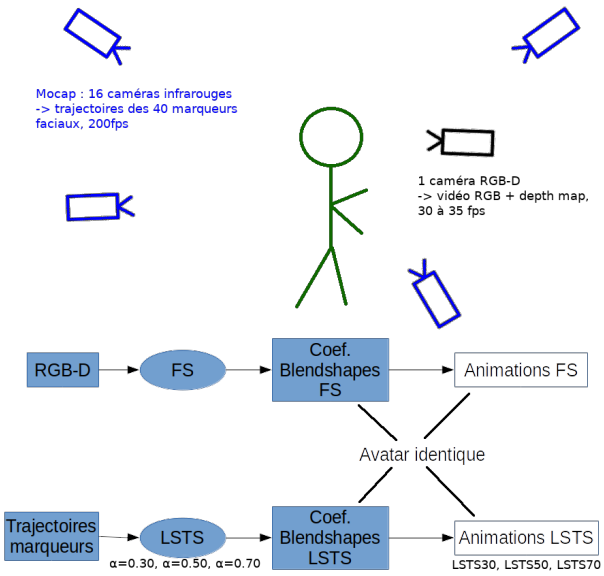


FIGURE 6.1 – La capture est faite simultanément par le système de *motion capture* et la caméra RGB-D afin de produire l'animation d'un même avatar à partir d'une scène identique via les méthodes LSTS (la notre) pour $\alpha = 0.30, 0.50, 0.70$ et FS (faceshift, logiciel commercial).

— via la méthode d'animation proposée dans le présent chapitre, que nous notons *LSTS* pour *Least-Square Thin-Shell*, en utilisant les données issues de la *motion capture* et en faisant varier le paramètre d'animation $\alpha = \alpha_{ts}$ [1] avec $\alpha_{ts} + \alpha_b = 1$;

— via le logiciel commercial *faceshift* [2][WBLP11a], que nous notons *FS*, reposant sur les données de type vidéo RGB-D.

Nous avons sélectionné un exemplaire pour chaque expression et chaque degré d'intensité pour chacune des évaluations perceptuelles présentées ci-après, soient 18 extraits au total (6 expressions $\times$ 3 intensités).

6.2 Description des tests statistiques utilisés

La plupart des conditions de validité des tests paramétriques telles que la distribution normale des variables étudiées ne sont pas vérifiables, aussi privilégions-nous dans cette étude les

1. Comme mentionné précédemment en section 5.2.3, une pondération raisonnable des deux énergies de régulation employées dans le processus d'optimisation consiste à appliquer la relation $\alpha_{ts} + \alpha_b = 1$ pour balancer ces deux énergies. Il n'est donc nécessaire de définir qu'une seule énergie, en l'occurrence α_{ts} que nous noterons simplement α dans le reste de ce chapitre.

2. `https://www.crunchbase.com/organization/faceshift#section-overview`

tests non-paramétriques.

La table 6.1 liste les différents tests utilisés pour analyser les résultats de nos études perceptuelles.

N^o-Nom	Statistique	Équivalent paramétrique	$H0$	conditions de validité
1-Test de corrélation de Spearman	ρ	Test de corrélation de Pearson	"Il n'existe pas de relation monotone entre le facteur (continu) et la variable (continuDe l'opérateur Laplacien à l'énergie de déformation Thin-Shelle) considérés."	La relation entre les deux variables à mettre en évidence doit être monotone
2-Test du χ^2 sur tableau de contingence	χ^2		"Il n'existe pas de relation entre le facteur (discret) et la variable (discrète) considérés."	Effectif théorique ≥ 5 pour chaque case du tableau de contingence
3-Test de Kruskal et Wallis	K	*Anova*	"La variable (continue) a la même distribution quelle que soit la valeur que peut prendre le facteur (discret)."	Effectif ≥ 5 pour chaque valeur que peut prendre le facteur
4-Test de Scheirer-Ray-Hare	SRH	*Anova* à deux facteurs	"La distribution de la variable (continue) ne dépend ni de la valeur d'un premier facteur (discret), ni d'un second facteur (discret) ni d'une combinaison de ces deux facteurs."	Effectif ≥ 5 pour chaque combinaison de valeurs que peuvent prendre les deux facteurs
5-Test des rangs signés de Wilcoxon	W	Test de Student pour des échantillons appariés	"Les deux populations (continues) provienne de la même distribution."	Effectifs ≥ 20

TABLE 6.1 – Les différents tests utilisés pour analyser les résultats de nos études perceptuelles.

6.3 Première évaluation : paramétrage de la méthode de synthèse

Le but de cette première évaluation perceptuelle est d'évaluer l'influence du paramétrage α de notre méthode de synthèse (LSTS) afin de choisir celui qui sera le plus approprié.

6.3.1 Hypothèses

Pour cette étude, nous nous sommes intéressés à quatre hypothèses ($H0_1$, $H0_2$, $H0_3$ et $H0_4$) afin de guider nos choix de paramétrage :

$H0_1$ Le paramétrage du modèle n'influe pas sur la capacité des interrogés à reconnaître le type d'émotion exprimée.

$H0_2$ L'intensité de l'émotion perçue est en moyenne la même quel que soit le paramétrage du modèle.

$H0_3$ La difficulté perçue à reconnaître l'émotion et son intensité est en moyenne la même quel que soit le paramétrage du modèle.

$H0_4$ La crédibilité de l'animation est en moyenne la même quel que soit le paramétrage du modèle.

6.3.2 Questionnaire

L'étude a été présentée sous la forme d'un questionnaire en ligne. Un panel de 27 personnes, dont 17 hommes et 10 femmes, composé majoritairement d'étudiants, âgés de 17 à 31 ans (moyenne $21,6$, médian 21) ont participé à l'étude. À chacun, il a été demandé de visionner 18 séquences de mouvement animées par notre méthode (voir section 6.1) x 3 paramétrages différents : $\alpha = 0.3$ (LSTS30), $\alpha = 0.5$ (LSTS50) et $\alpha = 0.7$ (LSTS70), soient 54 vidéos présentées dans un ordre aléatoire. Le choix de ces trois paramétrages possibles pour cette évaluation a été fait suite à une pré-évaluation informelle des animations produites par la méthode LSTS en fonction du paramétrage.

Avant de visionner les vidéos, les participants ont été préalablement informés qu'une collation leur serait offerte à l'issue de l'étude. Ils ont ensuite répondu à plusieurs questions d'ordre personnel concernant l'âge, le genre (H/F), leurs éventuels problèmes de vue [3] et l'habitude qu'ils ont des avatars virtuels (sur une échelle de Lickert allant de 1 à 7).

Après chaque vidéo, quatre questions étaient posées au participant (voir figure 6.2) :

1. Quelle expression reconnaissez-vous ? (parmi les choix suivants : Colère, Dégoût, Peur, Joie, Tristesse, Surprise)
2. Sur une échelle de 1 (très faible intensité) à 7 (très forte intensité), quelle est selon vous l'intensité de l'expression ?
3. Sur une échelle de 1 (très difficile) à 7 (très facile), répondre à ces deux questions vous a-t-il paru facile ?
4. Sur une échelle de 1 (très peu crédible) à 7 (très crédible), est-ce que l'expression faciale vous parait crédible ?

Pour la première question, nous avons opté pour une réponse à un choix forcé. Les questions 2 à 4 ont été évaluées sur une échelle de Lickert allant de 1 à 7.

6.3.3 Analyse des résultats

Chaque hypothèse a été testée par l'une des méthodes statistiques présentées précédemment. La table 6.2 récapitule les tests effectués.

Des quatre hypothèses testées, seule $H0_2$ a pu être invalidée (p-value < 0.01) (table 6.3).

3. tous les participants ayant mentionné un problème ont indiqué que ce dernier était corrigé (lunettes / lentilles) au moment de l'étude.

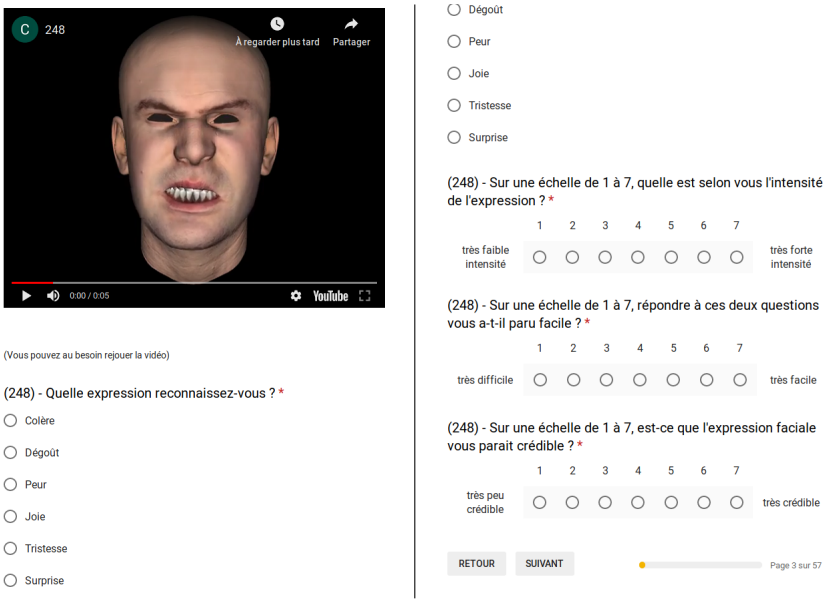


FIGURE 6.2 – [Étude 1] : le formulaire présenté aux 27 participants.

Hypothèse	Énoncé	test utilisé (voir table 6.1)
$H0_1$	*Le paramétrage du modèle n'influe pas sur la capacité des interrogés à reconnaître le type d'émotion exprimée*	2
$H0_2$	*L'intensité de l'émotion perçue est en moyenne la même quel que soit le paramétrage du modèle*	3
$H0_3$	*La difficulté perçue à reconnaître l'émotion et son intensité est en moyenne la même quel que soit le paramétrage du modèle*	3
$H0_4$	*La crédibilité de l'animation est en moyenne la même quel que soit le paramétrage du modèle*	3

TABLE 6.2 – [Étude 1] Hypothèses testées.

Afin d'approfondir l'analyse en ce qui concerne l'hypothèse $H0_2$, nous avons en outre effectué le test de Wilcoxon. Celui-ci nous permet d'affirmer que les résultats obtenus via le paramétrage $\alpha = 70$ est significativement différent de ceux obtenus via les paramétrages $\alpha = 30$ et $\alpha = 50$ et la table 6.4 nous montre que ce paramétrage a été perçu comme moins intense. Il ne nous permet pas en revanche d'affirmer que les résultats obtenus via le paramétrage $\alpha = 30$ sont significativement différents de ceux obtenus avec $\alpha = 50$.

Enfin, la figure 6.3 présente les matrices de confusion pour chacun des paramétrages utilisés (tout les degrés joués pour un même type d'expression confondu). Ces matrices sont semblables au sens où les principales sources d'erreur résultent de la confusion entre les expressions : *Colère* et *Dégout*, *Peur* vers *Surprise*, ainsi que *Tristesse* vers *Peur* et *Surprise*.

Hypothèse	$Z^{\text{test utilisé}}$ p-valeur	résultat
$H0_1$	Z^2 p-valeur	0.645 0.725
$H0_2$	Z^3 p-valeur	10.299 **0.006**
$H0_3$	Z^3 p-valeur	0.245 0.885
$H0_4$	Z^3 p-valeur	1.872 0.392

TABLE 6.3 – [Étude 1] Résultats des tests effectués pour les 4 hypothèses à invalider (χ^2 pour la première et Kruskal et Wallis pour les hypothèses 2, 3 et 4).

p-valeurs pour l'intensité perçue	LSTS50	LSTS70
LSTS30	0.185	**<0.001**
LSTS50		**0.006**

TABLE 6.4 – [Étude 1] Approfondissement en ce qui concerne l'hypothèse $H0_2$, test des rangs signés de Wilcoxon pour les intensités.

Nous avons également voulu vérifier s'il existe un effet d'interaction entre le paramétrage choisi et le type d'expression jouée ainsi que son intensité. La table 6.5 montre que l'on ne peut pas prouver l'existence d'un tel impact.

Par ailleurs, l'influence des caractéristiques personnelles des participants ainsi que celle de l'intensité jouée par l'acteur sur la perception des animations a été passée en revue. Les détails de cette analyse sont donnés en annexe A.1.

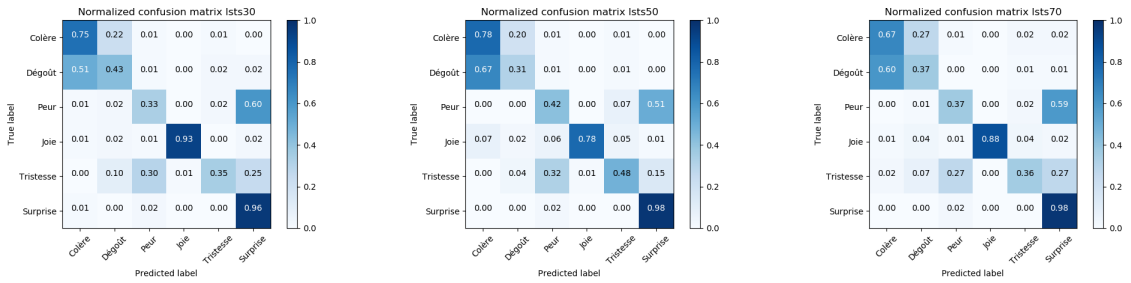


FIGURE 6.3 – Matrices de confusion pour chacun des paramétrages testés lors de la première étude : LSTS30 ($\alpha_{ts} = 0.3$), LSTS50 ($\alpha_{ts} = 0.5$), et LSTS70 ($\alpha_{ts} = 0.7$).

nom F1	nom F2	variable	[4]	F1	F2	F1*F2
Expression	méthode de	intensité	Z	110.063	10.299	5.106
jouée	synthèse	perçue	p-value	**<0.001**	**0.006**	0.884
Expression	méthode de	difficulté	Z	94.877	0.245	0.352
jouée	synthèse	perçue	p-value	**<0.001**	0.885	0.785
Expression	méthode de	crédibilité	Z	92.641	1.872	10.408
jouée	synthèse	perçue	p-value	**<0.001**	0.392	0.405
Intensité	méthode de	intensité	Z	385.94	10.30	0.520
jouée	synthèse	perçue	p-value	**<0.001**	**0.006**	0.971
Intensité	méthode de	difficulté	Z	15.238	0.245	0.157
jouée	synthèse	perçue	p-value	**<0.001**	0.885	0.997
Intensité	méthode de	crédibilité	Z	20.765	1.872	0.885
jouée	synthèse	perçue	p-value	**<0.001**	0.392	0.927

TABLE 6.5 – [Étude 1] Tests de Scheirer-Ray-Hare (N^o4).

6.3.4 Conclusions de la première étude

En définitive et au vu des résultats, la seule remarque qui puisse être faîte concernant cette première étude, à savoir définir un paramétrage optimal pour notre méthode de synthèse, est que les paramétrages $\alpha = 0.5$ et $\alpha = 0.3$ sont perçus comme plus intenses mais il n'est pas possible de les départager. Nous avons donc choisi le paramétrage $\alpha_{ts} = 0.3$ pour la suite des évaluations perceptuelles, car bien qu'il ne soit pas possible de le démontrer formellement, les animations produites avec le paramétrage LSTS30 semblent légèrement plus intenses que celles produites avec le paramétrage LSTS50 (voir table 6.6).

Question	Échelle	$\alpha_{ts} = 0.30$	$\alpha_{ts} = 0.50$	$\alpha_{ts} = 0.70$
Précision	(% expression correctement reconnues)	0.626	0.623	0.603
Intensité	1 (très peu intense) à 7 (très intense)	**5,086**	4,990	4,792
Difficulté	1 (très difficile) à 7 (très facile)	5,728	5,652	5,656
Crédibilité	1 (très peu crédible) à 7 (très crédible)	5,237	5,313	5,233

TABLE 6.6 – [Étude 1] Moyennes relevées pour chacune des questions posées selon le paramétrage du modèle.

6.4 Deuxième évaluation : évaluation de la méthode par rapport à l'état de l'art

Le but de cette deuxième évaluation perceptuelle est de comparer notre méthode d'animation avec une méthode du commerce. Pour ce faire, nous avons utilisé le logiciel commercial *faceshift*.

6.4.1 Hypothèses

Nous avons posé les quatre hypothèses de la première étude ($H0_1$, $H0_2$, $H0_3$, $H0_4$) auxquelles nous avons ajouté une cinquième ($H0_5$) :

— La fidélité de l'animation est en moyenne la même quelle que soit la méthode de synthèse.

6.4.2 Questionnaires

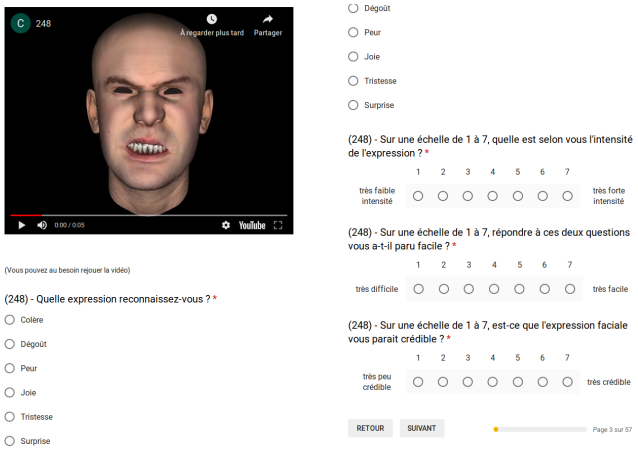
Cette étude a été présentée sous la forme de deux questionnaires qui ont été présentés à deux groupes de personnes différents. Nous nous référerons à ces deux évaluations sous les appellations d'étude $2a$ ou $2b$.

Groupe $2a$ Le groupe $2a$ est constitué de 21 participants, dont 14 hommes et 7 femmes, âgés de 18 à 52 ans (âge moyen 28.4, âge médian 24).

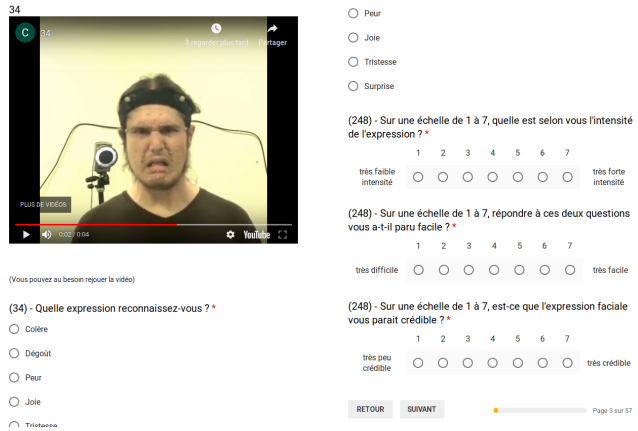
En suivant une méthodologie identique à la première étude, chaque participant du groupe $2a$ a dans un premier temps visualisé 36 vidéos dans un ordre aléatoire : les 18 (3 intensités $\times$ 6 expressions) séquences sélectionnées à la section6.1 animées par 2 méthodes différentes, soit par la méthode de synthèse présentée dans ce chapitre avec un paramétrage $\alpha = 0.3$ (LSTS30), soit par le logiciel commercial *faceshift* (FS).

Après chaque vidéo, les quatre questions concernant l'expression reconnue, son intensité, la difficulté et la crédibilité ont été posées au participant (voir figure 6.4). Puis, les participants du groupe $2a$ ont visionné dans un ordre aléatoire les vidéos réelles et se sont vus poser après chacune d'entre elles les quatre questions mentionnées précédemment (figure 6.4).

Groupe $2b$ Le groupe $2b$ est quant à lui constitué de 20 participants, 8 femmes et 12 hommes, âgés de 17 à 29 ans (âge moyen 20.45, âge médian 19).



(A)



(B)

FIGURE 6.4 – [Étude 2 groupe a] Le formulaire présenté aux participants. Les participants du groupe $2a$ ont d'abord visionné les animations LSTS30 et FS mélangée dans un ordre aléatoire en répondant aux questions au fur et à mesure, puis ont fait de même avec les vidéos dans un ordre aléatoire.

Les participants de ce groupe ont visionné 36 (6 expressions $\times$ 3 degrés d'intensité $\times$ 2 méthodes de synthèse) stimuli dans un ordre aléatoire. Cette fois-ci cependant, chaque stimulus a consisté en une animation et la vidéo correspondante jouées simultanément. Après chaque stimulus une unique question était posée (figure 6.5) : "Sur un échelle de 1 (très peu fidèle) à 7 (très fidèle), l'émotion exprimée par l'avatar est-elle fidèle à la vidéo ?".

6.4.3 Analyse des résultats

La table 6.7 récapitule les tests statistiques qui ont été effectués.

(C)

FIGURE 6.5 – [Étude 2 groupe *b*] Le formulaire présenté aux participants. Les participants du groupe *b* ont visionné l'animation (LSTS30 ou FS) et la vidéo correspondante simultanément dans un ordre aléatoire. À chaque fois, il leur a été posé une question unique portant sur la fidélité de l'animation par rapport à la vidéo.

Hypothèse	Énoncé	test utilisé (voir table 6.1)
$H0_1$ groupe 2a	*La méthode de synthèse n'influe pas sur la capacité des interrogés à reconnaître le type d'émotion exprimée*	2
$H0_2$ groupe 2a	*L'intensité de l'émotion perçue est en moyenne la même quel que soit la méthode de synthèse*	5
$H0_3$ groupe 2a	*La difficulté perçue à reconnaître l'émotion et son intensité est en moyenne la même quel que soit la méthode de synthèse*	5
$H0_4$ groupe 2a	*La crédibilité de l'animation est en moyenne la même quel que soit la méthode de synthèse*	5
$H0_5$ groupe 2b	*La fidélité de l'animation est en moyenne la même quel que soit la méthode de synthèse*	5

TABLE 6.7 – [Étude 2] Hypothèses testées.

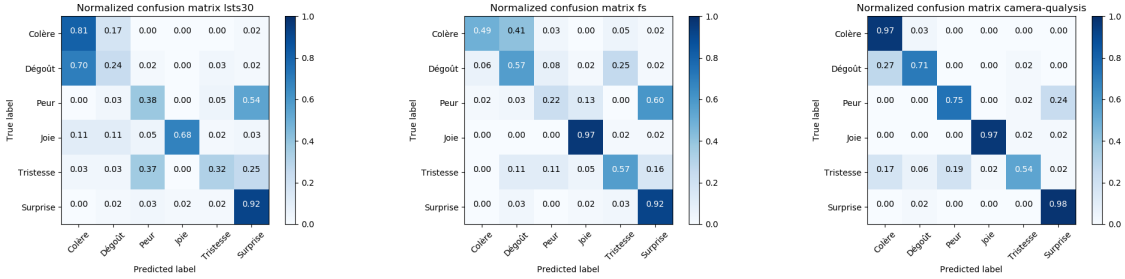


FIGURE 6.6 – Matrices de confusion pour la seconde étude : de gauche à droite : animation LSTS30 ($\alpha_{ts} = 0.3$), animation FS et vidéo réelle.

Les résultats sont donnés sur la table 6.8, les moyennes obtenues pour chaque question en table 6.9. Sans surprise, les expressions ont été globalement mieux reconnues sur les vidéos que sur les animations. De plus, la tâche a été jugée plus facile, l'intensité perçue et la crédibilité plus élevées sur les vidéos que sur les animations, quelle que soit la méthode de synthèse. Si l'on compare les deux méthodes, LSTS30 et FS uniquement, il n'est pas possible de prouver que l'une ou l'autre des méthodes est globalement mieux reconnue par les participants, en revanche il est possible d'invalider l'hypothèse selon laquelle les animations seraient perçues avec une intensité globalement égale ainsi que celle selon laquelle la difficulté de la tâche aurait été perçue comme globalement égale et celle selon laquelle la fidélité perçue est identique quelle que soit la méthode de synthèse. On peut ainsi affirmer que les expressions ont été perçues comme étant globalement plus intenses, que la tâche a été perçue comme plus facile et plus fidèle à la vidéo lorsque l'animation a été produite via la méthode LSTS30.

Enfin, la figure 6.6 présente les matrices de confusion pour chacune des deux méthodes de synthèse testées ainsi que pour la vidéo. En ce qui concerne la méthode LSTS30, les principales

$H0_1$ p-valeurs pour la précision	fs	vidéo
lsts30	0.076	**<0.001**
fs		**<0.001**

$H0_2$ p-valeurs pour l'intensité perçue	fs	vidéo
lsts30	**<0.001**	**<0.001**
fs		**<0.001**

$H0_3$ p-valeurs pour la difficulté perçue	fs	vidéo
lsts30	**<0.001**	**0.003**
fs		**<0.001**

$H0_4$ p-valeurs pour la crédibilité perçue	fs	vidéo
lsts30	0.192	**<0.001**
fs		**<0.001**

$H0_5$ p-valeurs pour la fidélite perçue	fs	vidéo
lsts30	**0.026**	NA
fs		NA

TABLE 6.8 – [Étude 2] P-valeurs pour chacune des hypothèses testées.

Question	Échelle	lsts30	faceshift	vidéo
Précision (groupe 2a)	(% expressions correctement reconnues)	0.558	0.624	0.82
Intensité (groupe 2a)	1 (très peu intense) à 7 (très intense)	4.862	4.476	5.241
Difficulté (groupe 2a)	1 (très difficile) à 7 (très facile)	5.442	5.164	6.119
Crédibilité (groupe 2a)	1 (très peu crédible) à 7 (très crédible)	5.090	4.966	5.923
Fidelité (groupe 2b)	1(très peu fidèle)) 7 (très fidèle)	5.042	4.820	NA

TABLE 6.9 – [Étude 2] Moyennes relevées pour chacune des questions posées.

sources d'erreur sont cohérentes avec celles de l'étude 1 réalisée avec un groupe de participants différent : *Colère* et *Dégout*, *Peur* vers *Surprise*, ainsi que *Tristesse* vers *Peur* et *Surprise*. En ce qui concerne la méthode FS nous retrouvons : *Colère* vers *Dégout* (mais pas *Dégout* vers *Colère*) et *Peur* vers *Surprise*, en revanche nous avons aussi *Dégoût* vers *Tristesse*. Pour ce qui est de la vidéo, les principales sources d'erreur sont globalement cohérente avec les animations : *Dégoût* vers *Colère*, *Peur* vers *Surprise*, *Tristesse* vers *Colère* et *Surprise*.

Enfin, l'influence des caractéristiques personnelles des participants du groupe 2a a été passée en revue. Les détails de cette analyse sont donnés en annexe A.1.

6.5 Discussion des résultats

En somme, ces deux évaluations perceptuelles avaient pour but de répondre à deux interrogations : i) Quel paramétrage choisir ? Il a été possible de démontrer que le paramétrage

LSTS70 produisait des animations perçues comme moins intenses que les deux autres paramétrage LSTS50 et LSTS30, en revanche il n'est pas possible de démontrer qu'il existe une différence significative entre ces deux derniers paramétrages, LSTS30 a donc été retenu. ii) Que vaut notre méthode de synthèse par rapport à une autre méthode de l'état de l'art ? Il a été possible de démontrer que la méthode LSTS30 produisait des animations jugées plus intenses, plus facile à reconnaître et plus fidèles vis-à-vis de l'expression jouée originale que les animations produites via FS, on estimera donc la méthode suffisamment bonne.

CHAPITRE 7

CONCLUSION

7.1 Contributions

Dans cette thèse nous nous sommes focalisés sur la mise en oeuvre d'une chaîne de traitement complète et cohérente allant de la capture de données jusqu'à la synthèse d'animations faciales sur la base de ces données et l'annotation automatique des expressions faciales pré-enregistrées.

Nous avons opté pour des modèles de synthèse basés données, en nous appuyant sur de la capture de mouvement (*mocap*). En effet, la *mocap* permet d'enregistrer le mouvement avec une grande finesse tant spatiale que temporelle. Pour ces mêmes raisons et pour faciliter la synchronisation entre les données faciales et corporelles, le choix a été fait d'adopter la même technique de capture. La *mocap* permet également d'effectuer des analyses du mouvement et d'améliorer sensiblement la segmentation et l'annotation des séquences capturées. Enfin, l'annotation automatique permet de réduire considérablement le temps consacré à la tâche fastidieuse de l'annotation manuelle.

De plus, la démarche que nous avons adoptée s'appuie sur une représentation des données homogène et cohérente à la fois pour l'animation et l'annotation automatique. La représentation choisie (vecteur de coefficients de blendshapes inspirés des actions unitaires *FACS*) est à la fois signifiante et précise pour la synthèse et suffisamment discriminante pour la reconnaissance.

Cette représentation par blendshapes présente d'autres avantages. Pour l'animation, il s'agit d'une représentation compacte, expressive (dans notre cas, chaque frame peut être représentée par un vecteur de dimensions 51) et linéaire, facilitant ainsi le stockage des données et un rendu 3D en temps réel. Par ailleurs, si les bases des blendshapes utilisées pour chacun des avatars de la base de données sont homogènes, c.à.d. si nous avons pour chaque avatar le même nombre de bases représentant chacune des déformations unitaires identifiées, on peut considérer que cela constitue un niveau d'abstraction intéressant pour la tâche de reconnaissance automatique, en particulier pour des données issues d'acteurs différents.

Outre la démarche globale en elle-même, la première contribution de cette thèse concerne

la méthode de synthèse que nous proposons (en chapitre 4) afin de générer les animations correspondant aux données capturées. Nous procédons en deux étapes, dans un premier temps (étape dite d'adaptation morphologique), nous calculons les positions qu'auraient chacun des marqueurs sur le maillage de l'avatar ciblé au moyen d'une régression RBF. Cette première étape permet une adaptation aux différences morphologiques entre l'acteur et l'avatar (et entre les acteurs). Dans un second temps nous calculons par optimisation le jeu de coefficients de blendshapes minimisant la distance entre les positions des marqueurs (adaptées) et le sommet du maillage qui lui correspond. Afin de pénaliser les résultats aberrants, nous avons utilisé de la manière originale le modèle *thin-shell* décrivant la déformation (étirement et courbure) d'une surface comme énergie de régularisation. Cette méthode a été validée par une étude perceptuelle.

La seconde contribution concerne l'annotation automatique des expressions faciales. Nous avions pour objectif de valider l'intérêt que peut représenter les blendshapes pour de la reconnaissance d'expressions faciales. Pour ce faire nous avons mis au point une méthode simple de segmentation et testé différents algorithmes basés machine learning. Les résultats présentés dans le chapitre 4 constituent la preuve de concept nécessaire avant de poursuivre plus en avant nos travaux dans cette direction.

Enfin la création de bases de données qui ont été successivement capturées afin de valider nos méthodes représente une contribution conséquente dans le cadre de ce travail. Ces bases de données constituent in fine un corpus complet regroupant les trajectoires des marqueurs au cours du temps, les animations correspondantes caractérisées par les coefficients de blendshapes, ainsi que les annotations réalisées manuellement.

7.2 Perspectives

Différentes perspectives viennent compléter ce travail. Certaines pistes abordées ici ont dans les faits déjà été explorées mais ne sont pas encore suffisamment abouties pour figurer dans le manuscrit de thèse.

Tout d'abord, en ce qui concerne la reconnaissance automatique, il est possible de passer outre l'étape de segmentation en effectuant la classification non pas sur des segments mais sur chaque frame. Les segments peuvent ensuite être déterminés en appliquant des méthodes de décision sur les séquences de frames reconnues. En particulier on peut agréger les suites contigües de frames classées de manière identique. Plusieurs travaux ont déjà été réalisés dans

ce sens en exploitant différents algorithmes : random forest, svm, knn, mais aussi des réseaux de neurones artificiels profonds (sans couche récurrente). L'une des difficultés qui apparaît est que lorsque les frames sont traitées indépendamment, on obtient potentiellement des résultats incohérents temporellement. En particulier, on obtient de nombreux segments contenant une ou deux frames seulement. C'est le cas lors des transitions d'une expression à l'autre ou lorsque l'expression faciale effectuée est ambigüe. Une solution qui a déjà été mise en place mais qui doit encore être validée de manière plus systématique est d'avoir recours à des méthodes tenant compte du contexte temporel pour classer chaque frame. Des solutions telles que les HMMs, les Conditionnal Random Fields (CRFs) et des réseaux de neurones comprenant des couches récurrentes (RNN de type LSTM ou GRU) ont été testées et donnent des résultats plus cohérents. Par ailleurs, ce type d'algorithme permet de répondre à une autre problématique qui est celle des transitions d'une expression faciale à l'autre. En effet, l'approche segmentale permet de détecter les limites temporelles entre lesquelles une expression faciale est stable et celles durant lesquelles elle ne le sont pas (transition) mais la question de ce qu'il convient de faire de ces intervalles de transition qui n'appartiennent à aucune classe n'est pas évidente. Le choix qui a été fait est, dans un premier temps, de ne pas tenir compte de ces transitions. Des solutions plus ou moins ad hoc peuvent être envisagées en restant dans une optique segmentale. Cependant les CRFs et RNNs apportent une solution nettement plus intuitive, les valeurs de sorties de ces algorithmes associées à chacune des classes étant très cohérentes temporellement. Cela devrait nous permettre d'analyser et de traiter en flux continu les données incluant zones stables et transitoires.

Jusqu'à présent, dans nos tâches de reconnaissance, nous avons utilisé des données de tests et d'apprentissage issues d'un même acteur. Il serait intéressant de vérifier si la représentation par blendshapes constitue une abstraction suffisante et permet d'appliquer à un acteur des modèles entraînés sur d'autres acteurs. Des travaux allant dans ce sens ont déjà été entrepris avec le corpus actuel mais il s'avère que le nombre d'acteurs (deux actuellement) et le volume de données n'est pas suffisant. Par conséquent, de nouvelles séances de captures sont à prévoir afin de réaliser cette étude.

Il est à noter que dans la constitution de nos corpus, des consignes d'intensité (peu marqué, marqué, exagérément marqué) associées à chaque expression ont été explicitement données aux acteurs. De plus les annotateurs ont eu pour consigne d'annoter de manière manuelle le niveau d'intensité observé. Ces différents niveaux d'intensité avaient, entre autre chose, pour objet d'ajouter une variabilité de style intéressante à notre corpus. Dans cette thèse nous avons présenté les résultats en ne tenant compte que des six expressions de base d'Eckman auxquelles

on a ajouté l'expression neutre. Des travaux sont en cours pour reconnaître automatiquement ces différents styles liés aux intensités et analyser leur influence. Cependant ils ne sont pas encore suffisamment aboutis pour figurer dans ce manuscrit.

Par ailleurs, nous n'avons abordé jusqu'ici que le vecteur de communication de l'expression faciale qualifié d'affectif (ou encore émotionnel). D'autres vecteurs de communication ont été identifiés dans des études sur les langues des signes (adjectival, clausal, direction du regard) et à l'avenir, il sera important de travailler sur ces différents canaux ainsi que sur la façon dont ils interagissent entre eux.

Enfin, il convient maintenant d'intégrer les travaux réalisés dans le cadre de cette thèse à la plateforme de synthèse concaténative de l'équipe, afin de générer des phrases expressives en langue des signes française.

Appendices

ANNEXE A

VALIDATION DE LA MÉTHODE D'ANIMATION PAR DES ÉTUDES PERCEPTUELLES

A.1 Première évaluation : paramétrage de la méthode de synthèse

La table A.2 restitue les principaux résultats obtenus concernant l'influence des caractéristiques personnelles des participants ainsi que l'intensité et l'expression jouées par l'acteur. En ce qui concerne le taux de bonne classification, il n'a pas été possible de montrer que ces informations ont un impact significatif. En revanche, ces éléments ont un impact significatif en ce qui concerne l'intensité, la difficulté et la crédibilité perçues. La table A.1 synthétise les principales remarques à ce sujet.

A.2 Deuxième évaluation : évaluation de la méthode par rapport à l'état de l'art

La table A.5 restitue les principaux résultats obtenus concernant l'influence des caractéristiques personnelles des participants ainsi que l'intensité et l'expression jouées par l'acteur. Certaines observations de cette deuxième étude diffèrent de celles de la première. La table A.2 résume les principales remarques à ce sujet.

Remarque	critiques	figures
Les expressions ont été perçues comme moins intenses par les plus âgés	i. La p-value (0.0498) est à la limite du rejet ii. Le coefficient $\rho = -0.051$ de corrélation de Spearman est relativement faible ce qui indique que quand bien même il existerait un lien, l'impact de ce lien est faible.	figure A.2
Les participants habitués aux avatars ont perçu les expressions comme plus intenses	i. Le coefficient $\rho = 0.068$ de corrélation de Spearman est relativement faible ce qui indique que quand bien même il existe un lien, l'impact de ce lien est limité. ii. 14 participants sur 27 ont choisi la valeur maximale pour décrire leur habitude des personnages virtuels, nous avons donc un risque que la relation entre les deux variables ne soit pas monotone ce qui pourrait expliquer la faiblesse du coefficient ρ.	figures A.2 et A.1(A)
Les participants ayant des problèmes de vue ont perçu les expressions comme moins intenses		figures A.2 et A.1(B)
La tâche a été perçue comme plus facile par les hommes que par les femmes	i. Cela ne signifie pas pour autant que les hommes ont été plus efficaces que les femmes pour classer les différentes expressions.	figures A.2 et A.1(C)
Les participants habitués aux avatars ont perçu la tâche comme étant plus difficile	i. Le coefficient $\rho = 0.057$ de corrélation de Spearman est relativement faible ce qui indique que quand bien même il existe un lien, l'impact de ce lien est limité. ii. 14 participants sur 27 ont choisi la valeur maximale pour décrire leur habitude des personnages virtuels, nous avons donc un risque que la relation entre les deux variables ne soit pas monotone ce qui pourrait expliquer la faiblesse du coefficient ρ.	figures A.2 et A.1(A)
La tâche a été perçue comme plus facile par les personnes n'ayant pas de problème de vue		figures A.2 et A.1(B)
Les personnes les plus âgées ont jugé les animations moins crédibles		figures A.2 et A.1(B)
Les expressions de forte intensité ont été jugées plus intense, plus facile à évaluer et plus crédibles	i. Cependant nous ne pouvons pas démontrer que les participants ont effectivement mieux reconnu les expressions jouées avec une forte intensité	figures A.2

TABLE A.1 – [Étude 1] Principales remarques concernant les caractéristiques personnelles.

facteur / variable		Age	genre	Habitué aux avatars	problèmes de vue	intensité jouée	expression jouée
Précision (Bonne classification)	Z	-0.290[1]*	0.017[2]	-0.026[1]*	<0.001[2]	3.36[2]	377.27[2]
	p-valeur	0.142	0.896	0.897	0.984	0.186	**<0.001**
Intensite perçue	Z	-0.051[1]	3.709[3]	0.068[1]	18.83[3]	0.50[1]	110.063[3]
	p-valeur	**0.0498**	0.054	**0.0096**	**<0.001**	**<0.001**	**<0.001**
difficulté perçue	Z	0.041[1]	49.16[3]	0.057[1]	31.81[3]	0.10[1]	94.877[3]
	p-valeur	0.114	**<0.001**	**0.030**	**0.001**	**<0.001**	**<0.001**
credibilite perçue	Z	-0.059[1]	0.022[3]	-0.014[1]	0.309[3]	0.12[1]	92.640[3]
	p-valeur	**0.024**	0.88	0.595	0.578	**<0.001**	**<0.001**

{[1], [2], [3], [4]} : référence des différents tests utilisés, cf. table 6.1.
* : Le test a été effectué sur la précision moyenne (nombre de bonne classification / nombre total de vidéos) de chaque individu afin de pouvoir appliquer le test de spearman (facteur et variable sont sensés être continus).

TABLE A.2 – [Étude 1] Tests de Kruskal-Wallis[3] sur différentes caractéristiques personnelles des pariticipants ainsi que sur les expressions et les intensités jouées par l'acteur.

Expression jouée	Colère	Dégoût	Peur	Joie	Tristesse	Surprise
Intensité	5.05	5.57	5.31	4.98	4.03	4.79
Difficulté	5.67	5.74	5.67	5.88	4.94	6.16
Crédibilité	5.47	5.49	5.40	4.75	4.76	5.70

TABLE A.3 – [Étude 1] Moyenne de l'intensité, la difficulté et de la crédibilité perçues suivant l'expression jouée.

nom F1	nom F2	variable	[4]	F1	F2	F1*F2
genre	méthode de synthèse	intensité perçue	Z	3.709	10.300	0.639
			p-value	0.054	**0.006**	0.727
genre	méthode de synthèse	difficulté perçue	Z	49.160	0.245	0.592
			p-value	**<0.001**	0.885	0.744
genre	méthode de synthèse	credibilité perçue	Z	0.022	1.872	0.0500
			p-value	0.881	0.392	0.975
problème de vue	méthode de synthèse	intensité perçue	Z	18.834	10.299	0.593
			p-value	**<0.001**	**0.006**	0.744
problème de vue	méthode de synthèse	difficulté perçue	Z	31.811	0.245	0.706
			p-value	**<0.001**	0.885	0.702
problème de vue	méthode de synthèse	credibilité perçue	Z	0.309	1.872	2.124
			p-value	0.578	0.392	0.346
Expression jouée	intensité jouée	intensité perçue	Z	110.06	385.94	35.69
			p-value	**<0.001**	**<0.001**	**<0.001**
Expression jouée	intensité jouée	difficulté perçue	Z	94.88	15.234	25.49
			p-value	**<0.001**	**<0.001**	**0.004**
Expression jouée	intensité jouée	crédibilité perçue	Z	92.64	20.76	27.98
			p-value	**<0.001**	**<0.001**	**0.002**

TABLE A.4 – [Étude 1] Tests de Scheirer-Ray-Hare[4], genre et problèmes de vue.

Habitude avatars (1 très peu habitué, 7 très habitué)	1	2	3	4	5	6	7
Effectifs	0	3	1	5	3	1	14
Intensité perçue moyenne (1 très peu intense, 7 très intense)	NA	4.73	5.30	4.92	4.54	5.41	5.05
Difficulté perçue moyenne (1 très difficile, 7 très facile)	NA	4.98	5.37	5.94	6.09	6.22	5.63
crédibilité perçue moyenne (1 très difficile, 7 très facile)	NA	5.40	5.43	5.46	4.70	5.85	5.22

(A)

Problème de vue	Oui	Non
Effectifs	8	19
Intensité perçue moyenne (1 très peu intense, 7 très intense) Toutes expressions confondues	4.69	5.07
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions faiblement marquées uniquement	3.47	3.89
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions moyennement marquées uniquement	5.04	5.38
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions exagérément marquées uniquement	5.55	5.94
Difficulté perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.42	5.79
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions faiblement marquées uniquement	5.27	5.54
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions moyennement marquées uniquement	5.48	5.88
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions exagérément marquées uniquement	5.51	5.94
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.30	5.24
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions faiblement marquées uniquement	5.18	4.98
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions moyennement marquées uniquement	5.28	5.32
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions exagérément marquées uniquement	5.44	5.43

(B)

genre	Homme	Femme
Effectifs	17	10
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.84	5.40
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.69	5.07
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.94	5.68
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.91	5.66
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.24	5.30
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	4.94	5.20
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.26	5.39
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.50	5.32

(C)

FIGURE A.1 – [Étude 1] (A) : Moyenne de l'intensité et de la difficulté perçues en fonction de l'habitude des participants vis-à-vis des personnages virtuels, (B) : Intensité moyenne perçue selon que le participant a des problèmes de vue ou non, (C) : Moyenne de la difficulté perçue selon le genre du participant.

facteur / variable		Age	genre	Habitué aux avatars	problèmes de vue	intensité jouée	expression jouée
Précision (Bonne classification)	Z	0.196[1]*	0.014[2]	0.232[1]*	1.599[2]	2.416[2]	200.047[2]
	p-valeur	0.395	0.705	0.312	0.206	0.299	**<0.001**
Intensite perçue	Z	-0.203[1]	2.015[3]	-0.040[1]	22.077[3]	0.464[1]	70.504[3]
	p-valeur	**<0.001**	0.156	0.175	**<0.001**	**<0.001**	**<0.001**
difficulté perçue	Z	0.062[1]	1.901[3]	0.054[1]	36.995[3]	0.136[1]	102.010[3]
	p-valeur	**0.038**	0.180	0.068	**<0.001**	**<0.001**	**<0.001**
credibilite perçue	Z	0.021[1]	15.973[3]	0.088[1]	14.988[3]	0.082[1]	55.106[3]
	p-valeur	0.487	**<0.001**	**0.003**	**<0.001**	**0.005**	**<0.001**

{[1], [2], [3], [4]} : référence des différents tests utilisés, cf. table 6.1.
* : Le test a été effectué sur la précision moyenne (nombre de bonne classification / nombre total de vidéos) de chaque individu afin de pouvoir appliquer le test de spearman (facteur et variable sont sensés être continus).

TABLE A.5 – [Étude 2a] Tests de Kruskal-Wallis[3] sur différentes caractéristiques personnelles des pariticipants ainsi que sur les expressions et les intensités jouées par l'acteur.

Expression jouée	Colère	Dégoût	Peur	Joie	Tristesse	Surprise
Intensité	5.06	5.24	5.34	4.74	3.97	4.81
Difficulté	5.93	5.30	5.62	5.92	4.65	6.03
Crédibilité	5.64	5.09	5.36	5.39	4.67	4.81
Fidélité (groupe 2b)	5.32	4.85	4.37	5.44	3.79	5.90

TABLE A.6 – [Étude 2a, 2b] Moyenne de l'intensité, la difficulté, de la crédibilité et de la fidélité perçues suivant l'expression jouée.

Remarque	critiques	figures
Les expressions ont été perçues comme moins intenses par les plus âgés		figure A.5
La tâche a été perçue comme plus facile par les plus âgés	i. Le coefficient $\rho = -0.062$ de corrélation de Spearman est relativement faible ce qui indique que quand bien même il existerait un lien, l'impact de ce lien est faible.	figure A.5
Les expressions ont été jugées plus crédibles par les femmes que par les hommes		figure A.5
Les expressions ont été jugées plus plus crédibles par les participants les plus habitués aux avatars	i. Le coefficient $\rho = -0.088$ de corrélation de Spearman est relativement faible ce qui indique que quand bien même il existerait un lien, l'impact de ce lien est faible.	figure A.5
Les expressions ayant des problèmes de vue ont perçu les expressions comme moins intenses		figure A.5
Les expressions ayant des problèmes de vue ont perçu la tâche comme plus difficile		figure A.5
Les expressions ayant des problèmes de vue ont perçu les expressions comme moins crédibles		figure A.5

TABLE A.7 – [Étude 2a] Principales remarques concernant les caractéristiques personnelles.

Habitude avatars (1 très peu habitué, 7 très habitué)	1	2	3	4	5	6	7
Effectifs	1	3	2	NA	1	6	8
Intensité perçue moyenne (1 très peu intense, 7 très intense)	5.33	5.55	4.40	NA	4.39	4.55	4.95
Difficulté perçue moyenne (1 très difficile, 7 très facile)	5.24	5.50	6.09	NA	5.15	5.40	5.70
crédibilité perçue moyenne (1 très difficile, 7 très facile)	5.15	5.38	6.06	NA	4.93	4.66	5.69

(A)

Problème de vue	Oui	Non
Effectifs	12	9
Intensité perçue moyenne (1 très peu intense, 7 très intense) Toutes expressions confondues	4.66	5.13
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions faiblement marquées uniquement	2.51	4.24
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions moyennement marquées uniquement	4.77	5.35
Intensité perçue moyenne (1 très peu intense, 7 très intense) Expressions exagérément marquées uniquement	5.70	5.80
Difficulté perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.38	5.83
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions faiblement marquées uniquement	5.12	5.67
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions moyennement marquées uniquement	5.34	5.93
Difficulté perçue moyenne (1 très difficile, 7 très facile) Expressions exagérément marquées uniquement	5.70	5.89
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.17	5.53
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions faiblement marquées uniquement	5.18	5.35
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions moyennement marquées uniquement	5.04	5.51
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Expressions exagérément marquées uniquement	5.30	5.72

(B)

genre	Homme	Femme
Effectifs	14	7
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.59	5.54
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.45	5.16
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.65	5.46
Difficulte perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.67	6.00
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.18	5.61
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.22	5.33
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.13	5.48
Crédibilité perçue moyenne (1 très difficile, 7 très facile) Toutes expressions confondues	5.20	6.04

(C)

TABLE A.8 – [Étude 2] (A) : Moyenne de l'intensité et de la difficulté moyennes perçues en fonction de l'habitude des participants vis-à-vis des personnages virtuels, (B) : Intensité moyenne perçue en selon que le participant a des problèmes de vue ou non, (C) : Moyenne de la difficulté perçue selon le genre du participant.

ANNEXE B

CORPUS

B.1 Temps d'enregistrement

Un corpus de données faciales a été capturé et annoté (manuellement) au cours de cette thèse. Il est en partie décrit en section 4. Les temps d'enregistrement associés à chaque expressions faciales sont présentés ici dans le tableau B.1.

d0 $\in N$	C	D	P	J	T	S	N	NA	transitions	Total
cr_IE	12.7	22.7	10.2	31.1	24.8	8.0	89.5	0.3	63.5	262.8
cr_SE	16.4	23.2	18.6	23.2	30.7	5.9	4.6	0.0	37.2	159.7
cr_IGE	14.6	14.6	23.1	18.7	19.3	11.4	93.5	0.0	59.6	254.8
cr_EU	62.4	20.4	55.3	53.2	49.2	72.0	251.9	0.5	118.0	682.9
cl2cr_IE	17.6	17.5	19.1	23.6	29.2	17.5	42.6	2.0	36.1	205.3
cl2cr_SE	5.1	5.1	7.8	5.6	8.6	2.8	8.0	0.0	20.3	63.3
cl2cr_IGE	5.3	5.3	3.8	16.5	14.1	8.5	80.2	4.0	35.5	173.3
cl2cr_EU	36.2	13.6	15.4	40.5	51.4	7.1	316.3	1.1	73.3	554.9
Toutes	170.2	122.5	153.3	212.6	227.2	133.2	886.7	7.9	443.5	2357.0

TABLE B.1 – Temps d'enregistrement pour chacune de 7 classes d'affect non-gradués en secondes selon l'annotation manuelle (les degrés d'intensité 0 sont considérés comme neutres).

BIBLIOGRAPHIE

[ACD+] Charly Awad, Nicolas Courty, Kyle Duarte, Thibaut Le Naour, and Sylvie Gibet. A Combined Semantic and Motion Capture Database for Real-Time Sign Language Synthesis. In 9th Int. Conf. on Intelligent Virtual Agent (IVA 2009), pages 432–438.

[APD10a] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. In 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, pages 1–4, April 2010.

[APD10b] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. In 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, pages 1–4, April 2010.

[Bat78] Robbin Battison. Lexical borrowing in American sign language. ERIC, 1978.

[BBA+07a] Bernd Bickel, Mario Botsch, Roland Angst, Wojciech Matusik, Miguel Otaduy, Hanspeter Pfister, and Markus Gross. Multi-scale capture of facial geometry and motion. ACM Transactions on Graphics, 2007.

[BBA+07b] Bernd Bickel, Mario Botsch, Roland Angst, Wojciech Matusik, Miguel Otaduy, Hanspeter Pfister, and Markus Gross. Multi-scale capture of facial geometry and motion. ACM Transactions on Graphics, 2007.

[BBB+10] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. In ACM SIGGRAPH 2010 Papers, 2010.

[BBB+14] Amit H Bermano, Derek Bradley, Thabo Beeler, Fabio Zund, Derek Nowrouzezahrai, Ilya Baran, Olga Sorkine-Hornung, Hanspeter Pfister, Robert W Sumner, Bernd Bickel, et al. Facial performance enhancement using dynamic shape space analysis. ACM Transactions on Graphics (TOG), 33(2) :13, 2014.

[BBH08] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008.

[BBL+08] Carlos Busso, Murtaza Bulut, Chi Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. Iemocap : Interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42(4) :335–359, 12 2008.

[BHB+11a] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. ACM Transactions on Graphics, 2011.

[BHB+11b] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. In ACM SIGGRAPH 2011 Papers, 2011.

[BHGS06] Tamy Boubekeur, Wolfgang Heidrich, Xavier Granier, and Christophe Schlick. Volume-Surface Trees. Computer Graphics Forum, 2006.

[BHPS10a] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. In ACM SIGGRAPH 2010 Papers, 2010.

[BHPS10b] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. ACM Transactions on Graphics, 2010.

[BS08] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. IEEE Transactions on Visualization and Computer Graphics, 14(1) :213–230, January 2008.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999.

[CB02] Erika Chuang and Chris Bregler. Performance driven facial animation using blendshape interpolation. Technical report, 2002.

[CBK+06] Cristóbal Curio, Martin Breidt, Mario Kleiner, Quoc C. Vuong, Martin A. Giese, and Heinrich H. Bülthoff. Semantic 3d motion retargeting for facial animation. In Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization, 2006.

[CC06] David Cristinacce and Timothy F Cootes. Feature detection and tracking with constrained local models. 2006.

[CDB02] E.S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on, 2002.

[CET01] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001.

[CHZ14] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. ACM Transactions on Graphics, 2014.

[CO02] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. Neural Computation, 2002.

[CTCG95] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. Computer vision and image understanding, 1995.

[CTFP05] Yong Cao, Wen C. Tien, Petros Faloutsos, and Frédéric Pighin. Expressive speech-driven facial animation. ACM Transactions on Graphics, 2005.

[CWLZ13] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. ACM Transactions on Graphics, 2013.

[CWWS12] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In CVPR 2012, 2012.

[CWZ+14] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse : A 3d facial expression database for visual computing. Visualization and Computer Graphics, IEEE Transactions on, 2014.

[DCFN06a] Zhigang Deng, Pei-Ying Chiang, Pamela Fox, and Ulrich Neumann. Animating blendshape faces by cross-mapping motion capture data. In Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, 2006.

[DCFN06b] Zhigang Deng, Pei-Ying Chiang, Pamela Fox, and Ulrich Neumann. Animating blendshape faces by cross-mapping motion capture data. In Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, 2006.

[DN08a] Zhigang Deng and Junyong Noh. Computer facial animation : A survey. In Data-Driven 3D Facial Animation. 2008.

[DN08b] Philippe Dreuw and Hermann Ney. Towards automatic sign language annotation for the elan tool. In 3rd Workshop on the Representation and Processing of Sign Languages, 2008.

[doC76] Manfredo P. doCarmo. Differential geometry of curves and surfaces, 1976.

[DYW+14] Mark Dilsizian, Polina Yanovich, Shu Wang, Carol Neidle, and Dimitris Metaxas. A new framework for sign language recognition based on 3d handshape identification and linguistic modeling. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[EF78] P. Ekman and W. Friesen. Facial Action Coding System : A Technique for the Measurement of Faci Consulting Psychologists Press, 1978.

[fac12] faceshift. http ://www.faceshift.com/product/, 2012.

[FJA+15] Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. Driving high-resolution facial scans with video performance capture. ACM Transactions on Graphics, 34, 2015.

[GCDLN11] Sylvie Gibet, Nicolas Courty, Kyle Duarte, and Thibaut Le Naour. The SignCom System for Data-Driven Animation of Interactive Virtual Signers : Methodology and Evaluation. ACM Transaction on Interactive Intelligent Systems, 2011.

[GSLT10] Xinbo Gao, Ya Su, Xuelong Li, and Dacheng Tao. A review of active appearance models. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(2) :145–158, 2010.

[GUE15] Vincent GUEDJ. Géométrie différentielle. 2015.

[HCTW11] Haoda Huang, Jinxiang Chai, Xin Tong, and Hsiang-Tao Wu. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. In ACM SIGGRAPH 2011 Papers, 2011.

[Her04] Aaron Hertzmann. Introduction to bayesian learning. In ACM SIGGRAPH 2004 Course Notes, 2004.

[HGMC05] A. Héloir, S. Gibet, F. Multon, and N. Courty. Captured motion data processing for real time synthesis of sign language. In Motion in Games, 2005.

[HLR11] Matt Huenerfauth, Pengfei Lu, and Andrew Rosenberg. Evaluating importance of facial expression in american sign language and pidgin signed english animations. In Proc. of the 13th International ACM SIGACCESS Conf. on Computers and Accessibility, ASSETS, pages 99–106, New York, NY, USA, 2011.

[HWHM15] AlicePuiLam Hung, Tim Wu, Peter Hunter, and Kumar Mithraratne. A framework for generating anatomically detailed subject-specific human facial models for biomechanical simulations. The Visual Computer, 2015.

[JL11] Robert E Johnson and Scott K Liddell. A segmental framework for representing signs phonetically. Sign Language Studies, 11(3) :408–463, 2011.

[KAL+17] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. ACM Transactions on Graphics (TOG), 36(4) :94, 2017.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing, volume 7, 2006.

[KCT00] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pages 46–53, 2000.

[KPBB02] J.-B. Kim, K.-H. Park, W.-C. Bang, and Z.Z. Bien. Continuous korean sign language recognition using gesture segmentation and hidden markov model. In Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, 2002.

[LADG08] François Lefebvre-Albaret, Patrice Dalle, and Frederick Gianni. Toward a computer-aided sign segmentation. In Language Resources and Evaluation Conference (LREC). European Language Resources Association, 2008.

[LAGP09] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. ACM Trans. Graph., 2009.

[LAGT+13] François Lefebvre-Albaret, Sylvie Gibet, Ahmed Turki, Ludovic Hamon, and Rémi Brun. Overview of the Sign3D Project High-fidelity 3D recording, indexing and editing of French Sign Language content. In Third International Symposium on Sign Language Translation and Avatar Technology (SLTA Chicago, United States, 2013.

[LAR+] J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. Practice and Theory of Blendshape Facial Models. In Eurographics 2014 - State of the Art Reports.

[LCK+10] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+) : A complete dataset for action unit and emotion-specified expression. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pages 94–101, June 2010.

[LD08] Qing Li and Zhigang Deng. Orthogonal-blendshape-based editing system for facial motion capture data. Computer Graphics and Applications, IEEE, 2008.

[LH74] Charles L Lawson and Richard J Hanson. Solving least squares problems. SIAM, 1974.

[LKK16] J. F. S. Lin, M. Karg, and D. Kulić. Movement primitive segmentation for human motion modeling : A framework for analysis. IEEE Transactions on Human-Machine Systems, 2016.

[LMX+08] Xuecheng Liu, Tianlu Mao, Shihong Xia, Yong Yu, and Zhaoqi Wang. Facial animation by optimized blendshapes from motion capture data. Computer Animation and Virtual Worlds, 2008.

[LN13] Thibaut Le Naour. Utilisation des relations spatiales pour l'analyse et l'édition de mouvement. PhD thesis, 2013.

[LTW95] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, 1995.

[LWP10a] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. In ACM SIGGRAPH 2010 Papers, SIGGRAPH '10, 2010.

[LWP10b] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. ACM Trans. Graph., 2010.

[LXC+15] Yilong Liu, Feng Xu, Jinxiang Chai, Xin Tong, Lijuan Wang, and Qiang Huo. Video-audio driven real-time facial animation. ACM Transactions on Graphics (TOG), 34(6) :182, 2015.

[LYYB13] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. ACM Transactions on Graphics, 2013.

[LZD13] B.H. Le, Mingyang Zhu, and Zhigang Deng. Marker optimization for facial motion acquisition and deformation. Visualization and Computer Graphics, IEEE Transactions on, 2013.

[MA07] Mark Meyer and John Anderson. Key point subspace acceleration and soft caching. ACM Transactions on Graphics, 2007.

[MAW+07] P. Merrell, A. Akbarzadeh, Liang Wang, P. Mordohai, J.-M. Frahm, Ruigang Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007.

[Max17] Max Planck Institute for Psycholinguistics. Elan v.4.9.4. http://tla.mpi.nl/tools/tla-tools/elan/, 2017.

[MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and AlanH. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In Visualization and Mathematics III. Springer Berlin Heidelberg, 2003.

[NCG19] Thibaut Le Naour, Nicolas Courty, and Sylvie Gibet. Skeletal mesh animation driven by few positional constraints. Journal of Visualization and Computer Animation, 30(3-4), 2019.

[NLG17] Lucie Naert, Caroline Larboulette, and Sylvie Gibet. Coarticulation analysis for sign language synthesis. In International Conference on Universal Access in Human-Computer Interaction, 2017.

[NN01] Jun-yong Noh and Ulrich Neumann. Expression cloning. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001.

[NRLG18] Lucie Naert, Clément Reverdy, Caroline Larboulette, and Sylvie Gibet. Per channel automatic annotation of sign language motion capture data. In Proceedings of the 8th Workshop on the Representation and Processing of Sign Languages : Involving the Language Community, LREC, pages 139–146, may 2018.

[RCWS14] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1685–1692, 2014.

[RGL15a] Clément Reverdy, Sylvie Gibet, and Caroline Larboulette. Animation faciale basée données : un état de l'art. In Proceedings of AFIG, nov 2015.

[RGL15b] Clément Reverdy, Sylvie Gibet, and Caroline Larboulette. Optimal marker set for motion capture of dynamical facial expressions. In Proceedings of Motion In Games, 2015.

[RHHL02] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. ACM Transactions on Graphics, 2002.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, 2001.

[RS06] Ruiduo Yang and S. Sarkar. Detecting coarticulation in sign language using conditional random fields. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), volume 2, pages 108–112, Aug 2006.

[RW05] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2005.

[SCO04] O. Sorkine and D. Cohen-Or. Least-squares meshes. In Proceedings of Shape Modeling Applications, 2004.

[SCOL+04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pages 175–184, 2004.

[SFPL10] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. A state of the art in structured light patterns for surface profilometry. Pattern Recognition, 2010.

[SHR17] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd, 2017.

[SLS+12a] Yeongho Seol, J.P. Lewis, Jaewoo Seo, Byungkuk Choi, Ken Anjyo, and Junyong Noh. Spacetime expression cloning for blendshapes. ACM Transactions on Graphics, 2012.

[SLS+12b] Yeongho Seol, J.P. Lewis, Jaewoo Seo, Byungkuk Choi, Ken Anjyo, and Junyong Noh. Spacetime expression cloning for blendshapes. ACM Transactions on Graphics, 2012.

[SNF05] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In ACM SIGGRAPH 2005 Papers, 2005.

[Sor05] Olga Sorkine. Laplacian mesh processing. In Eurographics (STARs), pages 53–70, 2005.

[SP04] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. In ACM SIGGRAPH 2004 Papers, 2004.

[SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 2002.

[SS18] Hanan Salam and Renaud Séguier. A survey on face modeling : building a bridge between face analysis and synthesis. The Visual Computer, 34(2) :289–319, 2018.

[SSK05] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Simple and efficient compression of animation sequences. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2005.

[Sto60] William C Stokoe. Sign language structure : An outline of the visual communication systems of the american deaf. Studies in Linguistics, Occasional Papers, 8, 1960.

[SWMT13] Jerry Schnepp, Rosalee Wolfe, John McDonald, and Jorge Toro. Generating co-occurring facial nonmanual signals in synthesized american sign language.(2013). 2013.

[SWTC14] Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. Automatic acquisition of high-fidelity facial performances using monocular videos. ACM Transactions on Graphics (TOG), 33(6) :222, 2014.

[SZ11] R. Southern and J.J. Zhang. Motion-sensitive anchor identification of least-squares meshes from examples. Visualization and Computer Graphics, IEEE Transactions on, 2011.

[TMCB07] Barry-John Theobald, Iain A. Matthews, Jeffrey F. Cohn, and Steven M. Boker. Real-time expression cloning using appearance models. In Proceedings of the 9th International Conference on Multimodal Interfaces, 2007.

[TMM+09] B.J. Theobald, I. Matthews, M. Mangini, J.R. Spies, T.R. Brick, J.F. Cohn, and S.M. Boker. Mapping and manipulating facial expression. Language and Speech, 2009.

[VBPP05] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In ACM transactions on graphics (TOG), volume 24, pages 426–433, 2005.

[VM01] Christian Vogler and Dimitris Metaxas. A framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding, 81(3) :358–384, 2001.

[WBLP11a] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In ACM SIGGRAPH 2011 Papers, 2011.

[WBLP11b] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. ACM Transactions on Graphics, 2011.

[Web17] Raphaël Weber. Construction non supervisée d'un modèle expressif spécifique à la personne thèse. PhD thesis, 2017.

[WLVG07] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007.

[WLVG08a] T. Weise, B. Leibe, and L. Van Gool. Accurate and robust registration for in-hand modeling. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008.

[WLVG08b] T. Weise, B. Leibe, and L. Van Gool. Accurate and robust registration for in-hand modeling. In Computer Vision and Pattern Recognition., 2008.

[WLVGP09] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. Face/off : Live facial puppetry. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, 2009.

[WPK+04] Tim Weyrich, Mark Pauly, Richard Keiser, Simon Heinzle, Sascha Scandella, and Markus Gross. Post-processing of scanned 3d surface data. In Eurographics symposium on point-based graphics, 2004.

[XBMK04] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

[XCLT14] Feng Xu, Jinxiang Chai, Yilong Liu, and Xin Tong. Controllable high-fidelity facial performance transfer. ACM Transactions on Graphics, 2014.

[YCS+08] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In 2008 8th IEEE International Conference on Automatic Face Gesture Recognition, pages 1–6, Sept 2008.

[ZH04] Song Zhang and Peisen Huang. High-resolution, real-time 3d shape acquisition. In Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on, 2004.

[Zha10] Song Zhang. Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. Optics and Lasers in Engineering, 2010.

[ZLN+17] Eduard Zell, JP Lewis, Junyong Noh, Mario Botsch, et al. Facial retargeting with automatic range of motion alignment. ACM Transactions on Graphics (TOG), 36(4) :154, 2017.

[ZPS04] Yu Zhang, E.C. Prakash, and E. Sung. A new physical model with multi-layer architecture for facial expression animation using dynamic adaptive mesh. Visualization and Computer Graphics, IEEE Transactions on, 2004.

[ZSCS04] Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces : High-resolution capture for modeling and animation. In ACM Annual Conference on Computer Graphics, 2004.

[ZSCS08] Li Zhang, Noah Snavely, Brian Curless, and StevenM. Seitz. Spacetime faces : High-resolution capture for modeling and animation. In Data-Driven 3D Facial Animation. 2008.

[ZYC+13] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy 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, 2013.

[ZYC+14] Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M. Girard. Bp4d-spontaneous : a high-resolution spontaneous 3d dynamic facial expression database. Image and Vision Computing, 32(10) :692 – 706, 2014. Best of Automatic Face and Gesture Recognition 2013.

Titre : Annotation et synthèse basée données des expressions faciales de la Langue des Signes Française.

Mot clés : expressions faciales, Langue des Signes Française (LSF), capture de mouvement, synthèse basée données, annotation par apprentissage automatique, évaluation perceptuelle

Resumé : La Langue des Signes Française (LSF) représente une part de l'identité et de la culture de la communauté des sourds en France. L'un des moyens permettant de promouvoir cette langue est la génération de contenu par le biais de personnages virtuels appelés avatars signeurs. Le système que nous proposons s'intègre dans un projet plus général de synthèse gestuelle de la LSF par concaténation qui permet de générer de nouvelles phrases à partir d'un corpus de données de mouvements annotées et capturées via un dispositif de capture de mouvement basé marqueurs (MoCap) en éditant les données existantes.

En LSF, l'expressivité faciale est le vecteur de nombreuses informations (e.g., affectives, clausales ou adjectivales), d'où son importance. Cette thèse a pour but d'intégrer l'aspect facial de la LSF au système de synthèse concaténative décrit précédemment. Ainsi, nous proposons une chaîne de traitement de l'information allant de la capture des données via un dispositif de MoCap jusqu'à l'animation faciale de l'avatar à partir de ces données et l'annotation automatique des corpus ainsi constitués.

La première contribution de cette thèse concerne la méthodologie employée et la représentation par blendshapes à la fois pour la synthèse d'animations faciales et pour l'annotation automatique. Elle permet de traiter le système d'analyse / synthèse à un certain niveau d'abstraction, avec des descripteurs homogènes et signifiants. La seconde contribution concerne le développement d'une approche d'annotation automatique qui s'appuie sur la reconnaissance d'expressions faciales émotionnelles par des techniques d'apprentissage automatique. La dernière contribution réside dans la méthode de synthèse qui s'exprime comme un problème d'optimisation assez classique mais au sein duquel nous avons inclus une énergie basée laplacien quantifiant les déformations d'une surface en tant qu'énergie de régularisation. Les résultats de la synthèse ont été évalués et validés au moyen de deux études perceptuelles.

Title : Data-driven annotation and synthesis of facial expressions in French sign language

Keywords : facial expressions, French Sign Language (LSF), motion capture (mocap), data-driven synthesis, annotation through machine learning, perceptual evaluation

Abstract :

French Sign Language (LSF) represents part of the identity and culture of the deaf community in France. One way to promote this language is to generate signed content through virtual characters called signing avatars. The system we propose is part of a more general project of gestural synthesis of LSF by concatenation that allows to generate new sentences from a corpus of annotated motion data captured via a marker-based motion capture device (MoCap) by editing existing data.

In LSF, facial expressivity is particularly important since it is the vector of numerous information (e.g., affective, clausal or adjectival). This thesis aims to integrate the facial aspect of LSF into the concatenative synthesis system described above. Thus, a processing pipeline is proposed, from data capture via a MoCap device to facial animation of the avatar from these data and to automatic annotation of the corpus thus constituted.

The first contribution of this thesis concerns the employed methodology and the representation by blendshapes both for the synthesis of facial animations and for automatic annotation. It enables the analysis/synthesis scheme to be processed at an abstract level, with homogeneous and meaningful descriptors. The second contribution concerns the development of an automatic annotation method based on the recognition of expressive facial expressions using machine learning techniques. The last contribution lies in the synthesis method, which is expressed as a rather classic optimization problem but in which we have included a Laplacian-based energy quantifying the deformations of a surface as regularization energy. The results of the synthesis system have been evaluated and validated through two perceptual studies.