Capture, annotation and synthesis of motions for the data-driven animation of sign language avatars

Lucie Naert

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

THÈSE DE DOCTORAT DE

UNIVERSITÉ BRETAGNE SUD
Comue Université Bretagne Loire

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par **Lucie NAERT**

Capture, Annotation and Synthesis of Motions for the Data-Driven Animation of Sign Language Avatars

Thèse présentée et soutenue à VANNES, le 03 juillet 2020
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires, UMR 6074
Thèse N° : 561

Rapporteurs avant soutenance :

Rosalee Wolfe	Professeur des universités, DePaul University, Chicago
Jean-Philippe Vandeborre	Professeur des universités, Institut Mines-Telecom (IMT Lille Douai), CRIStAL (UMR CNRS 9189)

Composition du Jury :

Président :	Pierre-François Marteau	Professeur des universités, Université Bretagne Sud
Examinateurs :	Marion Blondel	Chargée de recherche, CNRS, SFL (UMR CNRS 7023)
	Eleni Efthimiou	Directrice de recherche, Laboratoire ILSP, Athènes
Dir. de thèse :	Sylvie Gibet	Professeur des universités, Université Bretagne Sud
Co-enc. de thèse :	Caroline Larboulette	Maître de conférences, Université Bretagne Sud

ACKNOWLEDGEMENT

Ce travail a été le résultat d'efforts conjugués et je tiens à remercier toutes les personnes qui ont contribué de près ou de loin à sa finalisation.

Tout d'abord, merci à Sylvie et Caroline, mes deux directrices de thèse, pour leur disponibilité sans faille, leurs conseils et leur absence de réticence à travailler avec moi jusqu'à des heures indues. Mes travaux ont été meilleurs grâce à vous. Merci d'avoir été là, toutes les deux, quand je doutais de pouvoir continuer. Merci pour votre amitié.

I would also like to thank my reviewers, Rosalee Wolfe and Jean-Philippe Vandeborre, for agreeing to do a review of this (very) long thesis manuscript, for being available by e-mail and by video during this difficult time of global health crisis. Many thanks to my examiners, Marion Blondel, Eleni Efthimiou and Pierre-François Marteau for being part of this PhD defense jury.

Merci aussi (en vrac) :

À Michel Irdel ("Peau rouge") et à Christophe Gendreau Touchais ("Oreille blanche") d'avoir accepté de porter, pendant de longues heures, des marqueurs réfléchissants et une combinaison moulante dans une salle surchauffée. Merci de m'avoir donné les bases de la LSF et d'avoir toujours répondu à mes questions. Merci également à tous les participants à mes évaluations perceptuelles.

Aux membres de mon comité de suivi, Luce Morin et Christophe Baley, pour m'avoir encouragé à chaque réunion annuelle et pour vos si précieux conseils. Vous m'avez donné la motivation qui me manquait parfois pour mener ce travail à bien.

À l'équipe Expression dans son ensemble pour la bonne ambiance des séminaires au vert et les réunions mensuelles qui donnent l'impression d'appartenir à une équipe bien sympathique. Au laboratoire IRISA pour m'avoir accueillie dans les meilleures conditions possibles. À Mario et Sylviane pour leur aide technique et logistique incroyable, leur disponibilité et leur gentillesse. Les doctorants ont bien de la chance de les avoir.

Au groupe des non-permanents, qui se transforme au fil des départs des uns et arrivées des autres, et dont les membres passagers se serrent pourtant constamment les coudes. À Raounak, qui a commencé en même temps que moi et est passée par les mêmes étapes en même temps, ou presque, pour nos confidences, nos pleurs et nos fous rires, nos thés

d'après-midi, nos petits-déjeuners entre filles. À Nan, pour les cours de sport (et surtout les discussions après), pour ses raviolis, son honnêteté, ses encouragements et son pragmatisme. Tout devient possible quand on te parle, Nan. Surtout, merci à toutes les deux pour votre amitié ! À Clément, pour avoir été un co-bureau attentionné, pour avoir été le premier à repérer Biquette, la buse du lampadaire et, surtout, pour avoir apporté du chocolat (et des courges BIO !) à chaque fois qu'il y en avait besoin. À Jim pour ses plats chinois délicieusement pimentés. À Jamila, pour avoir eu le courage de terminer sa thèse et de montrer ainsi que c'était possible. À Jade, pour apporter l'amour des mots dans un monde de sciences "dures". À Jean-Christophe, Lionel, Tiago, Claire, Iris, Vanea, Pamela, Laureline, Izaskun, Elia, Jamal, Fadhlallah, Lei, Maël, Armel, Behzad, Edward, Romain, Minh Tan, Tra, and for all the others I forgot (shame on me!) for your good spirit and our talks. I am happy to know all of you. Merci !

À Nicolas, Kevin, Fabienne, Sophie et Florent pour avoir été une merveilleuse seconde famille et m'avoir accueilli à chaque fois que j'en avais besoin. Je suis contente de vous connaître.

À Carole, pour être mon amie depuis teeeeellllement de temps, pour sa fraîcheur et ses choix parfois surprenants mais toujours intéressants. À Nolwenn, pour son cynisme, son amour de la Bretagne qui me permet de la voir plus souvent, pour jouer à Skribbl et nous massacrer à chaque fois. C'est si agréable de pouvoir compter sur vous.

À l'ensemble de clarinettes pour les fous rires musicaux du vendredi soir.

À Maman, Papa, Paul, pour m'aimer et pour me supporter dans tous mes états même les moins flatteurs. Pour accepter mes décisions, quelles qu'elles soient, ou en tout cas, m'en donner l'impression. C'est ce qui compte, de toute façon. À mes mamies et papis, à mes oncles, tantes, cousins et cousines, pour leurs appels, leurs encouragements, leurs anecdotes, leurs recettes, pour les vacances au soleil et les Noëls au chaud.

Et enfin, à Mathieu, pour être là, avec moi, au moment où j'en ai besoin, pour partager avec moi les affres de la thèse de façon admirablement synchronisée, pour cuisiner comme un dieu et pour ses attentions permanentes.

Sans vous tous, mes "Frankensigns" n'auraient pas vu le jour.

Table of Contents

1 Présentation du travail de thèse **9**

1.1 Contexte . . . 11

1.1.1 Langues des signes . . . 11

1.1.2 Capture de mouvements . . . 11

1.1.3 Animation d'avatars signeurs . . . 12

1.2 Objectif et problématiques de recherche . . . 13

1.3 Contributions de la thèse . . . 14

1.4 Conclusion . . . 17

2 Introduction **19**

2.1 Context . . . 20

2.1.1 Sign Languages . . . 21

2.1.2 *Motion Capture* . . . 21

2.1.3 Animation of Signing Avatars . . . 22

2.2 Objective and Challenges . . . 23

2.3 Contributions of the Thesis . . . 24

2.4 Outline of the Thesis . . . 25

3 French Sign Language: Context and Challenges of the Thesis **27**

3.1 Sign Languages . . . 27

3.1.1 Status of Sign Languages . . . 28

3.1.2 Deaf's Culture and LSF History . . . 28

3.2 LSF Linguistic Theories . . . 29

3.2.1 Phonological Level: the Parametric Approach . . . 30

3.2.2 Lexical Level: Cuxac's Iconicity Theory . . . 33

3.2.3 Phonology and Iconicity Reconciled Thanks to Millet . . . 35

3.3 Synthesis Objectives . . . 38

3.3.1 Sign Construction from Phonological Recombination . . . 39

3.3.2 Utterance Synthesis . . . 43

3.4 Summary and Discussions 45

I State of the Art 47

4 Collection and Annotation of Sign Language Data 48
4.1 Existing Sign Language Corpora 48
4.1.1 Video Corpora 49
4.1.2 *Motion Capture* Corpora 51
4.2 Data Annotation 56
4.2.1 Terminology 56
4.2.2 Manual Annotation 59
4.2.3 Automatic Annotation of Human Motion Data 61
4.2.4 Automatic Annotation of Sign Language Data 64
4.3 Summary and Discussions 69

5 Synthesis of Sign Language Content 71
5.1 Movement of Avatars 72
5.2 Isolated Sign Synthesis 73
5.2.1 Linguistic Representation of Signs 73
5.2.2 Scripting Languages for Sign Representation 80
5.2.3 Synthesis Techniques 84
5.3 Utterance Synthesis 94
5.3.1 Utterance Representation 94
5.3.2 Utterance Synthesis Approaches 103
5.4 Existing Systems 107
5.4.1 JASigning and AnimGen 107
5.4.2 SignCom and Sign3D 108
5.4.3 *Paula-Azee* 109
5.5 Summary and Discussions 113

II Contributions 117

6 *LSF-ANIMAL*: A Motion Capture Corpus in French Sign Language 118
6.1 Corpus Definition 119

6.1.1 Objectives . . . 120
6.1.2 Content of the *LSF-ANIMAL* Corpus . . . 121
6.2 Acquisition of the Data . . . 124
6.2.1 Technical Considerations . . . 124
6.2.2 Signers and Elicitation . . . 127
6.3 Data Post-processing . . . 128
6.3.1 Identification of the Markers . . . 128
6.3.2 From Raw Data to Standard Motion Format . . . 129
6.3.3 Playback on an Avatar . . . 130
6.3.4 Data Annotation . . . 130
6.4 Perceptual Evaluation of the Corpus . . . 131
6.4.1 Design of the Evaluation . . . 132
6.4.2 Results . . . 135
6.5 Summary and Discussions . . . 144

7 Automatic Annotation of Continuous Sign Language MoCap Data 145
7.1 Objectives and Challenges . . . 146
7.1.1 Terminology . . . 147
7.1.2 Challenges . . . 147
7.2 Annotation Scheme of the *LSF-ANIMAL* corpus . . . 149
7.2.1 Annotation Tool . . . 149
7.2.2 Annotation Scheme . . . 149
7.3 Automatic Refinement of the Segmentation of the Gloss Tracks . . . 153
7.3.1 Motivations . . . 154
7.3.2 Method . . . 154
7.3.3 Application . . . 159
7.4 Automatic Annotation of the Hand Configurations . . . 159
7.4.1 Motivations and Challenges . . . 159
7.4.2 Method . . . 162
7.4.3 Results of the Annotation of the Hand Configuration Tracks . . . 177
7.5 Automatic Annotation of the Hand Placement . . . 182
7.5.1 Motivations and Challenges . . . 182
7.5.2 Discretization of the Signing Space . . . 183
7.5.3 Application on the *LSF-ANIMAL* corpus . . . 188

7.6 Summary and Discussions 189

8 Motion Synthesis and Editing for the Generation of New Sign Language Content 193

8.1 Overview 194
8.1.1 Objectives 194
8.1.2 Approach 195
8.1.3 Our Synthesis System 195
8.2 Building New Signs with Phonological Recombination 199
8.2.1 Hand Placement Mechanisms 200
8.2.2 Hand Configuration Mechanisms 212
8.2.3 Hand Movement Mechanisms 222
8.2.4 Synchronizing the channels 226
8.3 Utterance Synthesis 231
8.3.1 Utterance as a Sequence: Generation of Believable Transitions . . . 231
8.3.2 Adding Simultaneity: Back to Phonology 242
8.4 Summary and Discussions 244

9 Conclusion 249

9.1 Contributions 249
9.2 Future Work 251

Bibliography 255

Appendices 281

A Annotation tracks of the *LSF-ANIMAL* database 281

B Inverse Kinematics Based on the Jacobian 282

C Comparison of the Transition Durations 285

Abstract 287

Chapter 1

Présentation du travail de thèse

Contents

1.1 Contexte 11
1.2 Objectif et problématiques de recherche 13
1.3 Contributions de la thèse 14
1.4 Conclusion 17

> Le français a le mérite de décrire objectivement ce que je veux exprimer. Le signe, cette danse des mots dans l'espace, c'est ma sensibilité, ma poésie, mon moi intime, mon vrai style.
>
> *Le cri de la mouette*
> Emmanuelle Laborit, 1994

Les langues des signes (LS) sont les langues naturelles des Sourds du monde entier. Ce sont des langues visuogestuelles qui possèdent leur propre vocabulaire et syntaxe. Cependant, l'absence d'un système d'écriture généralement accepté pour les langues des signes rend la production et la diffusion de messages en LS dépendantes du support vidéo ou de dessins statiques de signes. Les dessins, par l'absence d'informations temporelles et par leur imprécision, peuvent être difficiles à comprendre tandis que les enregistrements vidéo ne contiennent pas les informations de profondeur du mouvement humain. De plus, l'édition, le stockage et l'analyse du contenu d'un enregistrement vidéo sont complexes. En outre, étant donné l'importance des expressions faciales du signeur pour la compréhension du message, le maintien de l'anonymat sur une vidéo n'est pas possible.

Pour remédier aux limites des solutions existantes, des personnages virtuels communiquant en langues des signes, appelés *avatars signeurs*, sont des technologies prometteuses. Les avatars permettent de préserver l'anonymat du signeur tout en conservant les expressions faciales et de gagner en interactivité. De plus, contrairement aux vidéos, la production signée par un avatar peut potentiellement être éditée afin de créer un nouveau contenu. Cependant, pour que cette technologie soit compétitive, l'avatar doit être entièrement animé par des mouvements naturels, réalistes et signifiants car son rôle est d'être compris et accepté par la communauté sourde. Pour ce faire, l'animation de l'avatar doit répondre à deux contraintes :

1) **Précision** : le degré de précision de l'animation doit être suffisamment élevé pour transmettre le message souhaité. L'animation finale doit donc présenter des configurations manuelles correctes, des expressions faciales adéquates et des mouvements corporels précis. Elle doit également satisfaire les contraintes cinématiques requises pour conserver le sens du message souhaité.

2) **Réalisme** : pour que l'avatar soit cohérent et convaincant, il doit être animé de manière crédible. L'avatar est considéré comme plus engageant s'il est animé par des mouvements humains réels.

La plupart des moteurs d'animation se basent sur une spécification du contenu à signer et produisent, soit de manière procédurale, soit en utilisant des données capturées, un mouvement qui est utilisé pour piloter un avatar. Les techniques d'animation procédurale créent souvent des mouvements robotiques et irréalistes, mais tout signe peut être produit avec précision tant qu'il peut être décrit à l'aide d'un langage de spécification. Avec l'animation basée données, les mouvements de l'avatar proviennent de données réelles et sont donc plus naturels et plus fluides que le résultat d'une animation procédurale, mais la variété des signes qui peuvent être synthétisés est limitée et/ou biaisée par la base de données initiale.

Comme nous considérons que l'acceptation de l'avatar est un enjeu majeur, nous avons choisi d'utiliser une approche basée données en utilisant des mouvements capturés, mais, pour contrer la limitation de cette technique, nous cherchons à enrichir le corpus initial en synthétisant de nouveaux signes ou de nouvelles instances de signes existants afin d'apporter de la variabilité et d'élargir la gamme de contenu de langue des signes disponible dans la base de données initiale.

1.1 Contexte

Ce travail de thèse porte sur l'analyse, la synthèse et l'évaluation de mouvements pour l'animation de personnages virtuels communiquant en langue des signes française. Ces mouvements sont basés sur des données capturées et annotées. Ainsi, cette thèse se situe à l'intersection de trois grands domaines : les langues des signes, la capture de mouvement et l'animation de personnages virtuels.

1.1.1 Langues des signes

Les langues des signes sont utilisées par une grande partie de la population sourde pour communiquer. Contrairement aux langues orales, qui utilisent la voix pour émettre le message et l'ouïe pour le recevoir, les langues des signes sont visuelles et gestuelles, c'est-à-dire que les mouvements de l'ensemble du corps sont utilisés pour transmettre un message qui sera interprété par l'interlocuteur via son canal visuel. Cela entraîne une contrainte spécifique : il est nécessaire de voir la personne avec laquelle on communique ainsi que l'espace tridimensionnel autour du signeur. En conséquence, un rendu bidimensionnel de cet espace entraîne inévitablement des pertes de précision. Une application interactive avec un personnage virtuel tridimensionnel permet à l'utilisateur de changer de point de vue et ainsi de répondre à cette contrainte.

La langue des signes n'est pas universelle. Il existe de nombreuses langues des signes tout comme il existe de nombreuses langues orales. Tous les travaux de cette thèse sont basés sur des mouvements capturés en Langue des Signes Française (LSF), la langue utilisée par les Sourds de France. Cependant, bien que les travaux de cette thèse aient été appliqués à la LSF, la méthodologie et les algorithmes utilisés pour ce travail peuvent être directement étendus à d'autres langues des signes.

1.1.2 Capture de mouvements

La capture de mouvement (*Motion Capture* ou *MoCap*) est une technique d'acquisition de mouvement. Plusieurs procédés de *MoCap* existent :

— la *MoCap* basée sur des systèmes optiques mesure la position et le déplacement de marqueurs passifs (réfléchissants) ou actifs (LED) placés sur le corps et/ou le visage d'un acteur, à l'aide de caméras infrarouges ou de vidéo. La *MoCap* optique donne des résultats précis mais certains marqueurs peuvent être obstrués pendant

la capture, ce qui entraîne des pertes dans les données qui doivent être comblées dans une étape de post-traitement.

— la *MoCap* basée sur des systèmes non optiques tire parti de différents types de capteurs, qu'ils soient inertiels, magnétiques, mécaniques ou électromyographiques, pour calculer le mouvement de l'acteur. Contrairement aux systèmes optiques, ces systèmes ne sont pas soumis à des obstructions. Cependant, l'équipement peut être intrusif et les données sont généralement moins précises qu'avec les approches optiques.

Avec l'avènement des méthodes d'apprentissage profond, il est désormais possible de concevoir des systèmes de reconnaissance et de synthèse de mouvements basés directement sur des données vidéo, qui sont plus faciles à acquérir et moins intrusives que les données de *MoCap*. Cependant, ces techniques ne sont pas encore adaptées à l'étude de la langue des signes car elles ne permettent pas de reproduire les mouvements fins des doigts et du visage avec précision.

Pour cette thèse, nous avons choisi d'utiliser un système de *MoCap* optique utilisant des marqueurs réfléchissants et des caméras infrarouges, qui nous permet d'avoir des résultats précis et fiables. Bien qu'elle soit souvent considérée comme une technique coûteuse et intrusive, nécessitant un post-traitement lourd et une certaine expertise dans le placement des marqueurs, c'est encore une technique largement utilisée au cinéma et dans les jeux vidéo pour l'animation de mouvements réalistes et/ou de mouvements spécifiques difficiles à modéliser à la main. La *MoCap* permet ainsi d'obtenir des mouvements réalistes – puisqu'ils sont basés sur des mouvements réels – qui peuvent être rejoués sur des personnages virtuels de différentes morphologies. De plus, les mouvements de *MoCap* peuvent être étudiés à l'aide de modèles statistiques afin d'en extraire des lois et des invariants. Les mouvements capturés forment donc une base de données riche qui peut être analysée, consultée et éditée pour animer un avatar.

1.1.3 Animation d'avatars signeurs

Les avatars signeurs sont des humanoïdes virtuels animés par du mouvement de langue des signes. Dans cette thèse, nous considérons une animation du squelette des avatars : les avatars sont ainsi constitués d'un modèle géométrique tridimensionnel (un maillage $3D$) piloté par un squelette. Un squelette est une structure hiérarchique composée de segments rigides (os) reliés par des articulations, à partir d'une articulation racine située

au niveau du bassin. Un mouvement est défini comme une séquence de poses du squelette qui sont le résultat d'opérations de rotation sur les articulations et, potentiellement, d'une opération de translation sur l'articulation racine. Pour animer l'avatar, le squelette est lié au modèle $3D$ dans l'étape de *rigging* : chaque sommet du modèle est lié à une ou plusieurs articulations du squelette par une relation pondérée. Pendant l'animation, le maillage est déformé en utilisant un algorithme standard.

Les avatars signeurs sont une technologie prometteuse pour la communauté des sourds. Outre la préservation de l'anonymat du signeur, les avatars sont interactifs : la vitesse de la séquence signée et l'apparence de l'avatar peuvent être facilement modifiées par l'utilisateur. De plus, l'utilisateur peut déplacer la caméra autour de l'avatar ou décider de passer d'une perspective à la troisième personne à une perspective à la première personne, pour faciliter sa compréhension de l'animation.

1.2 Objectif et problématiques de recherche

Cette thèse se situe à la jonction de ces trois domaines de recherche que sont les langues des signes, la *MoCap* et l'animation d'avatars. Les besoins de la communauté LS concernant l'animation d'avatars et la limitation des techniques de synthèse basées données nous amènent à définir l'objectif de la thèse comme suit :

> Notre objectif est la création de nouveaux signes et énoncés en Langue des Signes Française en tirant parti des techniques de synthèse basées données et des mouvements annotés présents dans une base de données de mouvements capturés.

Cet objectif soulève quatre problématiques de recherche :

1. **Définition d'une base de données de mouvements pertinents.** Les bases de données de *MoCap* pour l'étude des LS sont rares et une infime partie d'entre elles sont mises à disposition du public. Comme notre objectif est de synthétiser un nouveau contenu en langue des signes à l'aide de techniques basées données, la définition et la capture d'une base de données de LS sont nécessaires. Cette base de données initiale sera utilisée à la fois pour étudier et analyser les caractéristiques cinématiques des mouvements des LS mais aussi comme matière première pour créer de nouveaux signes. Elle doit être conçue pour gérer cette dualité analyse/synthèse.

Un état de l'art des différentes bases de données de LS est réalisée dans la première partie du chapitre 4.

2. **Annotation des données.** Les données de mouvement doivent être annotées afin d'être utilisées pour la synthèse basée données. Annoter des données consiste à segmenter un flux continu de mouvement et, ensuite, à étiqueter ces segments avec un vocabulaire précis. Les segments étiquetés peuvent alors être récupérés afin d'étudier un mouvement ou d'animer l'avatar. L'annotation peut se faire à différents niveaux en fonction de la granularité nécessaire à l'analyse et à la synthèse du mouvement. Comme l'animation finale de l'avatar dépend de ces annotations, une définition précise des segments de mouvement et du contenu des annotations est nécessaire. La base de données de LS et les annotations correspondantes constituent un corpus en langue des signes. Les méthodes d'annotation existantes sont présentées dans la deuxième partie du chapitre 4.

3. **Création de nouveau contenu à l'aide de modèles basés données.** En utilisant la base de données entièrement annotée, des techniques de synthèse de mouvement peuvent être définies afin de créer de nouveaux signes et énoncés. Pour enrichir ainsi la base de données initiale, il convient de tirer parti des algorithmes de récupération (*motion retrieval*) et d'édition de mouvements. L'analyse du contenu initial est également une étape intéressante pour définir des techniques de génération de mouvement paramétrées pour être conformes aux lois du mouvement appliquées à la langue des signes. Une synthèse des travaux existants sur les techniques d'animation pour les avatars signeurs est faite dans le chapitre 5.

4. **Évaluation quantitative et qualitative des animations générées.** La qualité du contenu de langue des signes synthétisé doit être évaluée grâce à (i) des mesures objectives pour déterminer si les animations générées satisfont les contraintes de mouvement spécifiées et à (ii) des opinions subjectives de la communauté ciblée afin d'évaluer l'acceptabilité de l'avatar.

1.3 Contributions de la thèse

Les contributions suivantes constituent des réponses aux quatre problématiques présentées précédemment.

Conception, capture et évaluation perceptuelle d'une base de données de LSF

Notre première contribution est la conception, la capture et l'évaluation perceptuelle d'une base de données de *MoCap* en LSF dédiée à la synthèse de nouveaux signes et énoncés. Ce corpus, appelé *LSF-ANIMAL*, comprend différents mécanismes de LSF utiles à notre travail de synthèse : des configurations de mains isolées, des signes isolés, des signes contextualisés, des mécanismes iconiques tels que des spécificateurs de forme et de taille, des proformes ou des gestes de pointage, et des énoncés complets contenant tous les éléments précédents. Un *markerset* complet, adapté à la variabilité et à la précision des mouvements de la langue des signes, a été conçu et testé avant d'être appliqué sur deux instructeurs sourds de LSF pour l'enregistrement du corpus *LSF-ANIMAL*.

Les données ont été traitées, puis évaluées dans le cadre d'une étude perceptuelle impliquant 50 participants. Nous avons constaté que les signes étaient bien reconnus et que les participants avaient majoritairement trouvé les mouvements de l'avatar précis, naturels et crédibles lors de l'exécution des signes et des énoncés même si le manque d'expressions faciales de l'avatar était parfois regretté.

Cette première contribution est décrite en détail dans le chapitre 6.

Développement de techniques de segmentation et de reconnaissance automatiques pour différentes pistes d'annotation

L'annotation d'un ensemble de données est une étape de post-traitement importante pour faire de l'analyse ou de la synthèse de données. En effet, elle ajoute une couche sémantique aux mouvements bruts contenus dans la base de données qui peut ensuite être requêtée en fonction de contraintes linguistiques. Nous avons défini un schéma d'annotation composé de 18 pistes d'annotation avec une piste dédiée pour chacun des composants de la main (configuration, emplacement, mouvement et orientation), selon une approche phonologique, et quatre pistes supplémentaires pour le niveau "glose".

Pour limiter les imprécisions et les erreurs de l'annotation manuelle et pour réduire le temps d'annotation, nous proposons d'automatiser ou de clarifier la segmentation et l'étiquetage de certaines pistes : celles des gloses, des configurations manuelles et des emplacements.

Nous avons d'abord présenté un raffinement de la segmentation des **gloses** réalisé en analysant les propriétés cinématiques des mouvements de la LSF et, plus précisément, en

détectant les minima locaux dans la norme de la vitesse pour chaque main. Le raffinement a été réalisé en sélectionnant les minima les plus proches pour chaque segment défini manuellement. Cette segmentation est basée sur l'observation que chaque main a un comportement partiellement autonome. L'utilisation de caractéristiques quantitatives précises rend notre segmentation moins biaisée, plus précise et plus homogène que la segmentation manuelle.

Dans une deuxième section, nous avons présenté une technique pour automatiser la segmentation et la reconnaissance des **configurations de la main** basée sur 190 distances normalisées de la main. La segmentation est basée sur une mesure de la quantité de variation dans les distances entre les articulations des doigts et sur la comparaison de cette quantité avec un seuil fixe. L'étape de reconnaissance est basée sur des méthodes d'apprentissage automatique entraînées sur un sous-ensemble de données, annoté manuellement, et qui permet d'attribuer une étiquette à chaque segment de *configuration manuelle*. Cette méthode a été entraînée et testée sur une base de données spécifique, et appliquée sur *LSF-ANIMAL* pour réduire le besoin d'annotation manuelle.

Enfin, nous avons présenté une méthode de calcul de l'**emplacement de la main** par rapport à trois dimensions spatiales : *Hauteur*, *Distance* et *Orientation radiale*. L'annotation de l'emplacement de la main a été faite en fonction de notre discrétisation de l'espace de signation qui est dépendante de la position du signeur et de la portée de ses bras.

Cette deuxième contribution est présentée en détail dans le chapitre 7.

Développement de techniques de synthèse basées données

Une grande partie des travaux existants dans le domaine de l'animation d'avatars signeurs utilise exclusivement, pour la synthèse de signes, soit le rejeu de données de mouvements réels, soit de la synthèse procédurale pure souvent basée sur des systèmes de spécification phonologique tels que *HamNoSys*. Dans le premier cas, les mouvements de l'avatar conservent le réalisme de mouvements réels mais le nombre de signes pouvant être générés est limité par la base de données initiale. Dans le second cas, la variété des signes qui peuvent être synthétisés est beaucoup moins restreinte – l'expressivité du système de spécification étant presque la seule limite – mais les signes qui en résultent sont robotisés et irréalistes.

Nous proposons un système hybride qui tire parti des deux philosophies afin de créer de nouveaux signes, absents d'une base de données annotée, mais conservant les propriétés de réalisme des mouvements réels. Pour cela, nous nous appuyons sur une vision centrée

sur les éléments phonologiques tels que présentés dans différents travaux linguistiques et à la base de nombreux systèmes de représentation des signes. Nous proposons ainsi de constituer de nouveaux signes sous leur forme de citation ainsi que des phénomènes d'inflexion grâce à des techniques de (i) récupération de mouvement par composante phonologique et (ii) de recombinaison de ces éléments. Nous décrivons différentes techniques qui permettent de multiplier les possibilités d'une base de données en modifiant les valeurs prises par les différents éléments phonologiques. En plus de la simple recombinaison, nous avons mis en œuvre des techniques de synthèse pure, à savoir la cinématique inverse et des techniques d'interpolation que nous avons adaptées pour être au plus près de la vérité terrain résultant, par exemple, dans la définition d'une interpolation sigmoïde paramétrée.

En effet, en plus d'utiliser les données comme unité de base de notre synthèse, nous utilisons les connaissances extraites de l'analyse de ces données pour améliorer le résultat final. Les données constituent ainsi un matériau de construction et une base de connaissances sur le mouvement, grâce aux profils cinématiques des différentes trajectoires des articulations.

En ce qui concerne la synthèse des énoncés, nous avons d'abord travaillé sur la génération de transitions pour un moteur de synthèse concaténative. Dans ce cas, un énoncé est considéré comme une séquence de signes. Cependant, ce travail, combiné à nos analyses précédentes de la synchronisation des signes, peut être réutilisé pour créer des transitions réalistes dans le cas de signes exécutés simultanément (par exemple, dans une situation de description). De plus, comme nous travaillons à la fois sur les signes dans leur forme de citation et sur les mécanismes d'inflexion, nous avons la possibilité de construire des énoncés riches qui ne sont pas constitués exclusivement de signes non infléchis.

Cette dernière contribution est détaillée dans le chapitre 8.

1.4 Conclusion

Cette thèse porte sur la capture, l'annotation, la synthèse et l'évaluation des mouvements des bras et des mains pour l'animation d'avatars communiquant en langue des signes. Nous avons cherché à créer un nouveau contenu en langue des signes française (LSF) linguistiquement pertinent qui possède le réalisme des mouvements humains. À cette fin, nous avons proposé d'utiliser une base de données de mouvements capturés comme : (i) matière première pour notre système de synthèse et, (ii) matériel d'analyse pour identifier et recréer les caractéristiques qui rendent les mouvements de langue des

signes réalistes et sémantiquement significatifs.

À cette fin, nous avons mis en place une chaîne de traitement complète allant de la création d'un corpus LSF et de son annotation automatique à la synthèse de nouveaux contenus en langue des signes. Dans chaque contribution, nous avons placé au cœur de notre travail les éléments phonologiques que sont le mouvement, l'emplacement et la configuration des mains, en suivant des travaux linguistiques existants.

Des perspectives à ces travaux de thèse sont discutées dans le chapitre 9.

Chapter 2

Introduction

Contents

2.1 Context 20
2.2 Objective and Challenges 23
2.3 Contributions of the Thesis 24
2.4 Outline of the Thesis 25

> If you immerse yourself into a foreign language, then you can actually rewire your brain... It affects how you see everything.
>
> *The Arrival*
> Denis Villeneuve, 2016

Sign Languages (SL) are the natural languages of the Deaf around the world. They are visual-gestural languages with their own vocabulary and syntax but the lack of a generally accepted writing system for signed languages makes SL message production and broadcast dependent on the video medium or on static drawings of signs. The absence of temporal information and the imprecision of the drawings make them difficult to understand while video recordings lack the depth information of human motion and their editing, storage and analysis are complex issues. In addition, given the importance of the signer's facial cues for intelligibility, maintaining anonymity on a video is not possible. To solve those limitations, signing avatars, that is virtual characters using sign languages, are promising technologies. Avatars make it possible to preserve the anonymity of the signer, to gain interactivity and, unlike videos, the signed production can potentially be edited in order to create any new signed content. However, for this technology to be competitive, the

avatar must be fully animated with natural, realistic and meaningful motions as its role is to be understood and accepted by the Deaf community. To do this, the animation of the avatar must satisfy two constraints:

1) **Precision**: the degree of precision of the animation must be high enough to accurately convey the desired message. The final animation must therefore present correct manual configurations, adequate facial expressions and precise body movements. It must also satisfy the kinematic constraints required to maintain the meaning of the desired message.

2) **Realism**: for the avatar to be consistent and compelling, it must be animated in a believable way. The avatar will be considered more engaging if it is driven with human-like motions.

Most animation engines take as input a specification of the desired sign language content and produce, either procedurally or using captured data, a motion that is used to drive an avatar. Procedural animation techniques often create robotic and unrealistic motions, but any sign can be precisely produced as long as it can be described using a sign language representation. With data-driven animation, the avatar's motions come from real data and are thus more natural and smoother than the result of procedural animation but the variety of the signs that can be synthesized is limited and/or biased by the initial *Motion Capture* database.

As we considered the acceptance of the avatar to be a prime issue, we chose to use a data-driven process based on captured motions but, to counter the shortcomings of this technique, we seek to enrich the initial corpus by synthesizing new signs or new instances of existing signs in order to bring variability and broaden the range of SL content available in the initial database.

2.1 Context

This thesis deals with the analysis, synthesis and evaluation of motions for the animation of virtual characters communicating in SL. Those motions are based on annotated captured data. Thus, this thesis is located at the intersection of three broad domains: sign languages, *Motion Capture* and virtual character animation.

2.1.1 Sign Languages

Sign languages are used by a large part of the deaf population to communicate. Unlike oral languages, which involve the voice for emitting the message and the sense of hearing for its reception, sign languages are visual-gestural, i.e. the movements of the whole body are used to convey a message that will be interpreted by the interlocutor via his visual channel. This leads to specific constraints: it is necessary to see the person with whom one is communicating as well as the three-dimensional space around the signer. As a consequence, a two-dimensional rendering of this space inevitably leads to losses in precision. An interactive application with a three-dimensional virtual character allows the user to change its point of view and thus makes it possible to meet these two constraints.

Sign language is not universal. There are many sign languages just as there are many oral languages. All the work in this thesis is based on captured movements in French Sign Language (LSF), the language used by the Deaf of France. However, although the work of this thesis has been applied to LSF, the methodology and algorithms used for this work can directly be extended to other sign languages.

2.1.2 *Motion Capture*

Motion Capture, or *MoCap*, is a motion acquisition technique. Several *MoCap* processes exist:

— *MoCap* based on optical systems measures the position and displacement of passive (reflective) or active (LED) markers placed on an actor's body and/or face using infrared or video cameras. Optical *MoCap* gives precise results but some markers can be obstructed during the capture leading to gaps in the data that have to be filled in a post-processing step.

— Non-optical systems, also called active *MoCap*, take advantage of different types of sensors, whether inertial, magnetic, mechanical or electromyographic, to compute the motion of the actor. Contrary to optical systems, active *MoCap* is not subject to obstructions. However, the equipment can be intrusive and the data is generally less accurate than with optical approaches.

With the advent of deep learning methods, it is now possible to design motion recognition and synthesis systems based directly on video data, which is easier to acquire and less intrusive than *MoCap* data. However, this type of technique is not yet suited to the

study of sign language as the subtle movements of the fingers and face are not precisely reproduced.

For this thesis, we chose to use an optical *MoCap* system using reflective markers and infrared cameras which allows us to have accurate and reliable results. While it is often seen as an expensive and intrusive technique, requiring heavy post-processing and some expertise in marker placement, it is still a technique widely used in cinema and video games for the animation of realistic movements and/or specific motions that are difficult to model by hand.

MoCap makes it possible to obtain realistic movements – since they are based on real movements – at a high spatial and temporal resolution that can be replayed on virtual characters of different shapes and sizes. In addition, movements from *MoCap* can be studied using statistical models in order to extract motion laws and invariants. Captured motions can therefore form a rich database that can be analyzed, queried and edited to animate an avatar.

2.1.3 Animation of Signing Avatars

Signing avatars are virtual humanoids that are animated with sign language content. In this thesis, we consider a skeletal animation of avatars: avatars are thus constituted of a three-dimensional geometric model (*3D mesh*) driven by a skeleton. A skeleton is a hierarchical structure composed of rigid segments (*bones*) linked by *joints*, starting from a *root joint* located at the pelvis. A motion is defined as a sequence of skeleton poses which are the results of rotation operations on the joints and, potentially, a translation operation on the root joint. To animate the avatar, the skeleton is bound to the 3D model in the *rigging* step: each vertex of the model is linked to one or more joints of the skeleton by a weighted relation. During animation, the mesh is deformed using a standard linear blend skinning technique.

Signing avatars are a promising technology for the Deaf community. In addition to the preservation of the signer's anonymity, avatars are interactive: the speed of the signed sequence and appearance of the avatar can be easily changed by the user. Moreover, the user can move the camera around the avatar or decide to switch from a third-person to a first-person perspective which can ease the comprehension of the SL animation for education purposes for example.

2.2 Objective and Challenges

This thesis is located at the junction of these three research domains that are sign languages, *MoCap* and avatars animation. The needs of the SL community regarding avatar animation and the limitations of data-driven motion synthesis techniques lead us to define the thesis objective as follows:

> In this thesis, we aim to create novel signs and utterances in French Sign Language (LSF) by taking advantage of data-driven synthesis techniques and of the annotated motions present in a *MoCap* database.

This objective raises four research challenges:

1. **Definition of a relevant motion database.** *MoCap* databases for SL studies are rare and a very small portion of them are made available to the public. As our goal is to synthesize novel sign language content using data-driven techniques, it requires the definition and capture of an SL database. This initial database will be used both to study and analyze the kinematic features of SL motions but also as raw material to create new signs. It must be designed to handle this analysis/synthesis duality.
2. **Data annotation.** The motion data needs to be annotated in order to be used for data-driven synthesis. Data annotation consists in segmenting a continuous flow of motion and, then, in labeling those segments. The labeled segments can then be retrieved in order to study or animate the avatar in a specific way. The annotation can be done at different levels depending on the granularity needed for the analysis and synthesis of motion. As the final animation of the avatar depends on those annotation labels, a precise definition of the motion segments and labels is needed. The SL database and the corresponding annotations constitute a sign language corpus.
3. **Creation of new content using data-driven models.** Using the fully annotated SL database, motion synthesis techniques can be defined in order to generate variations in the initial data and, thus, to broaden the range of sign language content available. To enrich the initial database in such a way, motion retrieval and motion editing algorithms should be taken advantage of. The analysis of the initial content is also an interesting step to define parameterized motion generation techniques that comply to SL motion laws.

4. **Quantitative and qualitative evaluation of the generated animations.** The quality of the synthesized SL content must be evaluated thanks to (i) objective metrics to determine if the generated animations satisfy the specified motion constraints and to (ii) subjective opinions of the targeted community in order to assess the acceptability of the avatar since the aim is to synthesize the language of specific people.

2.3 Contributions of the Thesis

We have addressed the four challenges presented above with the following contributions:

1. A **French Sign Language *MoCap* database** dedicated to the synthesis of new content with a special focus on the phonological components of SL. The database is composed of isolated hand configurations, isolated signs, inflected signs and full utterances. It contains animal descriptions, storytelling sequences and grammatical LSF mechanisms. The motions of two LSF instructors were captured for a total duration of 1h. The corpus was perceptually evaluated.
2. **Different automated techniques and a refinement process for sign language data annotation**. Automated techniques were developed for the annotation of the hand configuration track using machine learning algorithms and the annotation of the placement track using a discretization of the signing space. We also defined a refinement of the manual segmentation at the gloss level based on the kinematic properties of the two wrists during SL production.
3. The **implementation of data-driven motion synthesis techniques** based on a parametric definition of SL. We describe the synthesis of new LSF content following the linguistic hypothesis that LSF production is the result of the combination of different discrete phonological components (sometimes called *parameters*). The proposed techniques are based on motion retrieval per annotation track and on the modular reconstruction of new SL content with the additional use of motion generation techniques such as motion interpolation and inverse kinematics. The specification of the parametrization of the motion generation techniques is done according to the kinematic features of real motions observed in the captured data.
4. **Quantitative and qualitative evaluations**. The contributions of this thesis

were evaluated either quantitatively or perceptually. For example, quantitative machine learning metrics were used to validate the annotations of the hand configurations while the quality of the corpus content was assessed thanks to questionnaires adapted to the specific needs of deaf participants by favoring visual questions translated in LSF.

2.4 Outline of the Thesis

The PhD thesis is organized in eight chapters: Chapter 3 gives detailed information about the social context of SL and the linguistic theories around LSF. The synthesis objectives of this thesis with respect to those theories are detailed. The five following chapters are divided into two parts: *State of the Art* and *Contributions*.

The *State of the Art* part contains two chapters that provide a review of the existing work related to the data-driven animation of signing avatars. Chapter 4 gives the current state of research in SL corpora and SL annotation while Chapter 5 is an overview of the different techniques to synthesize SL content. This part shows the limitations of the existing techniques and the significance of our approach.

The *Contribution* part contains three chapters. Chapter 6 presents the definition, construction and perceptual evaluation of the *LSF-ANIMAL* corpus, the French Sign Language *MoCap* database that was used for our work. Chapter 7 describes our contribution to the annotation of SL motion captured data. It introduces our different methods and gives a quantitative evaluation of the results. Chapter 8 introduces our work on the synthesis of SL content at the sign and utterance levels.

Finally, the last chapter concludes this thesis by providing a summary of the contributions as well as perspectives on future work.

Chapter 3

French Sign Language: Context and Challenges of the Thesis

Contents

3.1 Sign Languages 27
3.2 LSF Linguistic Theories 29
3.3 Synthesis Objectives 38
3.4 Summary and Discussions 45

> Je vois comme je pourrais entendre. Mes yeux sont mes oreilles. J'écris comme je peux signer. Mes mains sont bilingues. Je vous offre ma différence. Mon coeur n'est sourd de rien en ce double monde.
>
> *Le cri de la mouette*
> Emmanuelle Laborit, 1994

In this chapter, we present Sign Languages (SL) and, particularly, French Sign Language (LSF) from a socio-cultural and linguistic points of view in order to understand the scientific issues and challenges of signing avatar animation.

3.1 Sign Languages

Sign languages are visual-gestural languages used by an important part of the deaf population. While vocal languages rely on the voice to convey a message and on the audio

channel to receive it, sign languages use different sensory channels. The movements of the whole body and facial expressions are used to produce a message which is interpreted by interlocutors via their visual channels.

3.1.1 Status of Sign Languages

Sign languages are languages in their own right that have been developed naturally over time by Deaf communities to communicate. They have their own vocabulary, called *signary*, and precise grammatical rules. They are rich languages that, thanks to their visual and gestural aspects, can take advantage of the possibilities of space to tell stories, communicate information, describe situations with precision, date temporal events, make poetry, jokes, etc.

Contrary to a general belief, there is not just one universal sign language that would allow all deaf people around the world to communicate. This misunderstanding often comes from the fact that hearing people see sign languages as artificial languages created by hearing people to help people with a hearing handicap to communicate. This is not true: sign languages were born naturally from the need to communicate of the Deaf themselves [1]. As a result, there are as many sign languages as there are Deaf communities. Even if many countries present their sign language like French Sign Language in France or Libras in Brazil, there is no direct correspondence between countries and sign languages: several sign languages can be present in the same country and one sign language can extend beyond the borders of a country. And, like with vocal languages, there are regional variants. Moreover, an international signary exists called International Sign (IS) that can be used by deaf people who lack a common sign language.

As presented in Chapter 6, our initial database is in French Sign Language; therefore, our work targets particularly the Deaf of France. However, the work can be extended to any sign language provided that one is in possession of data in that language.

3.1.2 Deaf's Culture and LSF History

A distinction is often made in the use of the word "deaf" based on the presence or absence of a capital letter: "deaf" refers to a person with a medical condition of hearing loss while "Deaf" is used to talk about the cultural aspects claimed by the Deaf communities [2]. Sign language is the preferred language of the **D**eaf, while not all **d**eaf people use a sign language. Sign languages are objects of pride for the Deaf communities and are thus an

essential part of Deaf culture.

Sign languages, like any language, have their own history, which often helps to explain their diffusion, their limitations and their choices. We outline here the history of French Sign Language which has an indirect impact on our research and, more directly, on the reception of our work. LSF does not have a specific date of appearance. Until the middle of the 18th century and the work of L'Abbé de l'Epée, signed languages were seen more as senseless gestures than as means of communication. L'Abbé de l'Epée made it possible to bridge the gap between the deaf and the hearing by affirming that signed languages were worthy of interest and by creating schools throughout France to teach it to deaf children. In 1880, the Milan Congress ended a prosperous period for Deaf culture and LSF. It was decided that oralism, which advocates the learning and use of spoken French through lip reading and speech exercises, should be preferred for the communication of the deaf. Schools that were teaching the LSF had their funding withdrawn and the LSF was marginalized. However, in the 1970s, movements for the recognition of sign language began to emerge: it is the "Deaf awakening". As a consequence, in 1991, LSF was re-authorized in schools and, in 2005, it was recognized as a language in its own right [1]. In France, at present time, there are about 100,000 Deaf people communicating in LSF according to the *Fédération Nationale des Sourds de France* [2]. Since 2008, students in high school can choose LSF as an option for the Baccalauréat.

For a long time, the use of LSF has been repressed which can explain a certain resentment of the French Deaf community towards hearing people. This is less the case in other countries such as the United States, where the use of the ASL has not been prohibited: important Deaf institutions have emerged, including Gallaudet University, the only liberal arts university in the world specifically designed for deaf people.

3.2 LSF Linguistic Theories

A sign is a lexical unit of sign languages as is a word to oral languages. An utterance is close to the concept of "sentence" in oral languages: it is composed of signs, executed sequentially or simultaneously, and presents the statement of an idea.

In this section, we present three linguistics theories on LSF [3]. We first present sign languages phonology, then the iconicity mechanisms of Cuxac [3] that take place at a

1. `http://laboiteasaussure.fr/lsf_history.html`
2. `https://www.fnsf.org/`
3. The presented LSF theories can be extended to other SL.

lexical level and, finally, we try to reconcile those two antagonist points of view by presenting the linguistic proposition of Millet [1]. We chose those three models because we consider them interesting both from a socio-linguistic point of view and for our synthesis objectives: those three theories highlight different opinions on SL and their evolution over time while covering the linguistic mechanisms that we wish to synthesize.

3.2.1 Phonological Level: the Parametric Approach

Sign language phonology is born from the need to give a structure to SL and, this way, to make a parallel between signed and spoken languages. In spoken languages, *phonemes* are units of sound that compose words and make it possible to distinguish one word from another. *Minimal pairs* are two words that have an identical pronunciation except for one phoneme. Minimal pairs are used to determine the phonemes relative to one language. For example, the words "tad" and "dad" compose a minimal pair that illustrates the existence of two separate phonemes, /t/ and /d/ in English.

Moreover, the existence of a phonological system leads to the validation of the *double articulation* principle [4] for sign languages. This principle claims that human languages can be segmented on two levels: a first level linking an element with a meaning (in vocal languages, a word is the element of minimal size for the first level) and a second level composed of distinctive units without meanings (the phonemes).

In the case of SL, the parametric approach states that a sign is a sequence of discrete values taken by SL *phonological components*. Those phonological components often refer to 5 elements [5]–[7]:

1. the **hand configuration** corresponds to the overall shape of the hand characterized by the disposition of the fingers (see examples on Figure 3.1). Each configuration corresponds to a discriminating and meaningful posture of the hand. [*OISEAU*] (bird) and [*OIE*] (goose) are minimal pairs for hand configurations: the signs are done in the same way except for the hand configuration ('2-fingers beak' configuration for [*OISEAU*] and '3-fingers beak' for [*OIE*]). Researchers do not agree on the number and nature of hand configurations. For French Sign Language (LSF), Cuxac lists 39 configurations [3] [4] while Boutora identifies 77 configurations [8] and Millet 41 [1]. We chose to use 48 configurations after having compared five different sources (see Section 7.2).

4. His list identifies up to 41 hand configurations counting the hand configuration alternatives. He

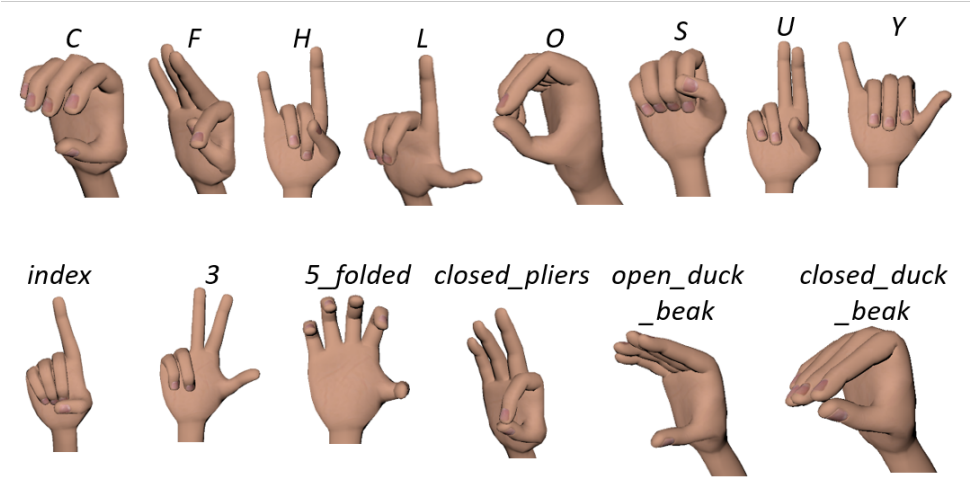

Figure 3.1 – Some hand configurations.

2. the **hand placement** is the location of the hand in the signing space or on the body of the signer. Depending on the field, it is defined differently. In linguistic studies, hand placement, also called *anchoring* (*ancrage*) [1], [9], is often defined as the global area where the sign is produced (e.g., neutral space in front of the signer, eyes, hand palm) in its *citation form* [10]–[12] (i.e. deprived of any syntactic context, also called "uninflected form" [13]). It seldom changes during the sign production.

   For the computer animation community, it designates the discrete area or the specific coordinates where the hand is positioned at a precise time. In this case, and depending on the discretization of the signing space, the hand placement can vary during the realization of a sign. We used the second definition.

3. the **hand movement** represents the trajectory of the wrist over time. Contrary to discrete placement or hand configuration which take a value in finite sets, hand movement is continuous and can represent any trajectory.

   We chose not to place the secondary movements (i.e. small movements of the fingers) into this category as we consider finger motions as changes in the hand configuration. Moreover, we consider that the movements performed by the hand with respect to the wrist correspond to changes in the orientation of the hands (like in the sign [*DEMANDER*] (to ask) [5]) and not to hand movements.

4. the **hand orientation** is defined by the direction of the hand palm and of the palm

specifies that it is not a "closed inventory".

5. See `https://dico.elix-lsf.fr/dictionnaire/demander`

normal (see Figure 3.2). It is strongly constrained by the hand movement and the human physiological limits. However some minimal pairs distinguished only by the orientation exist (e.g, [*MAISON*](house) and [*DEMANDER*](to ask)).

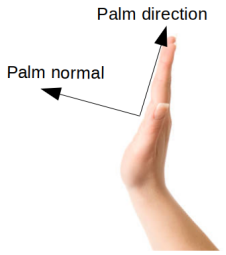

Figure 3.2 – Definition axes for the hand orientation.

5. the **non-manual features** (NMFs) include the facial expressions, the mouthing, the gaze and torso direction. Linguists are not unanimous on the phonological quality of the NMFs. Some linguists state that some signs can only be distinguished thanks to the facial expression of the signer thus making the facial expressions, at least, a phonological component [9]. [*GAGNER*](win) and [*DOMMAGE*](too bad) are such a minimal pair (the first is accompanied by a happy expression while the second by a sad expression). On the contrary, others argue that any facial expression can be done during the production of those signs depending on the expressive hue that the signer wants (a touch of weariness or irony can be added by doing a sad expression on the [*GAGNER*] sign and a happy one for the [*DOMMAGE*]) [1]. For those linguists, such signs are simply homonyms.

   We do not claim to have the linguistic background to make a decision on this matter. However, we found that the lack of facial expression in our work and on the animation of our avatar was often deplored making facial expressions an important element of SL animations [6].

A sign is therefore a sequence of values taken in parallel by each of these components in finite sets (hand configuration, hand placement) or infinite sets (hand movement, hand orientation). However, while in vocal languages, words are formed with a simple sequence

6. NMFs and, particularly, the facial expressions, have many functions in SL: (i) as we just mentioned, they are sometimes considered as a phonological component, (ii) they can also express the affect (surprise, fear or anger hues can be added to the message with the adequate facial expression), or (iii) have syntactic roles (e.g., a raise of the eyebrow can indicate a question while a swelling of the cheek can add information about the width of an object). Those functions can co-occur in signs and utterances and should be managed in complete synthesis systems [14]. However, this thesis focuses on arm and hand movements and not on NMFs.

of phonemes, in sign languages, components take on values simultaneously in addition to having a sequential aspect. To designate this particularity, we refer to sign languages as *multilinear* languages [15].

3.2.2 Lexical Level: Cuxac's Iconicity Theory

Sign language iconicity refers to the similarity between the sign and what it designates (resp. the *signifier* and *signified* of Saussure [16]).

In his work, Cuxac states that LSF [7] has two modes of production [3]: (i) a non-illustrative one based on signs in their citation form (called *standard signs* by Cuxac) whose gestural execution is relatively invariant and (ii) an illustrative one, called *structure of great iconicity* (SGI), using different spatiotemporal mechanisms to describe a scene, an object or an animal.

Cuxac describes various LSF mechanisms for SGI. He calls them *"transfers"* [8] but, in this thesis, we use other denominations that we find clearer and more accepted in the international community.

— **Size and Shape Specifiers** (≈ *transfers de taille et de forme*) consists in using the hand configuration, the wrist orientation and the amplitude of the motion to describe the shape and size of an object. For example, [TO GIVE] will not be performed with the same hand configuration and orientation in the sentence "I give you a book" (configuration of the *'open_duck_beak'* representing thick flat objects) and in "I give you a glass" (configuration of the *'C'* representing cylindrical objects) (see Figure 3.7). In addition, the amplitude of the motion is often used as a size specifier, as shown on Figure 3.3.

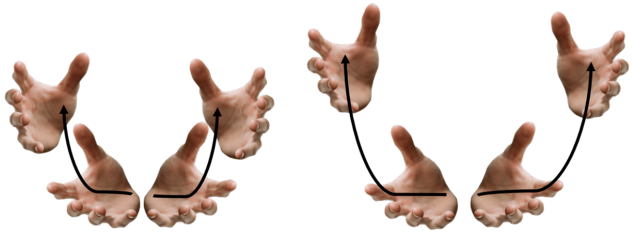

Figure 3.3 – Iconicity on the sign *[BOL]* (bowl): the size of the bowl corresponds to the amplitude of the motion.

7. Cuxac research on LSF can be applied to many SL.
8. Cuxac's work was complemented by Sallandre's, who proposed a classification of "transfers" in her doctoral thesis [17].

— In **Proforms** [18], also called *classifiers predicates* [19], [20] (≈ *transfers de situation*), the hand configuration and movement embody the depicted situation, object, or person. More precisely, it consists in using a particular hand configuration representing an object (e.g., a flat hand for a car) or a person (the index finger raised for a standing person or bent for a sitting person) and a movement (or a simple placement) of the hand to show the movement performed by the object or its location. For example, two flat hands moving forward with one hand behind the other one will depict two cars driving in a line. Proforms are used to describe a scene vividly, accurately and with few signs. Thus, a sentence like "Two persons are crossing a street" can be signed much more richly using the possibilities of proforms than with the corresponding uninflected signs: details on the precise nature of the walk, on the position of the two persons with respect to each other, or on the size of the street can be given much more naturally and intuitively. Another example of description using proforms is given in Figure 3.4.

Figure 3.4 – In this scene: a pedestrian ('index' proform with the left hand) waits for a car ('flat_hand' proform with the right hand) to pass.

— **Role Shift** (*transferts personnels*) designates the impersonation by the signer of the person, animal or object that he is talking about in order to describe its behavior. The whole upper part of the body of the signer is used in this case. For example, a character's gait can be accurately described by using the arms of the signer to represent the legs of the described thing. The whole body is involved and the movements performed can be very subtle (a subject with a slight or severe limp will not be signed in the same way).

These iconic structures are permanently present in LSF discourses and are essential in situations of scene description or storytelling. However, before Cuxac, little work existed

on the iconic aspect of LSF. Indeed, it was in direct contradiction with the principle of Saussure stating that "the link uniting the signifier to the signified is arbitrary" [21] and thus almost rejected by sign languages linguists. Cuxac's work is therefore an important step forward in LSF study.

While the parametric approach aims to draw a parallel between spoken and signed languages with the concepts of phoneme and double articulation, Cuxac tries to show that the iconicity of sign languages cannot be described using existing linguistic tools for spoken languages. Cuxac thus affirms that standard signs and SGI should not be studied in the same way. Moreover, in [22], Cuxac refutes the parametric models by stating that "a complete assimilation between LSF sublexical units and phonemes is not theoretically convenient." He affirms that it is incorrect to decompose LSF at a lower level than the smallest units with a meaning (i.e. the signs) and that the phonological level is thus irrelevant.

3.2.3 Phonology and Iconicity Reconciled Thanks to Millet

In her book *Grammaire descriptive de la langue des signes française*, Millet rejects the separation of Cuxac between iconic and non-iconic signs [1]. She states that iconicity is always present, even in the "standard signs" of Cuxac due to the visual-gestural nature and spatial aspects of LSF. According to her theory, all signs are motivated and, thus, are not arbitrary. However, the motivation link *(le lien de motivation)* between a sign and its signified can be more or less difficult to perceive. She considers phonology as a tool, developed by linguists for spoken languages at first but that can still be used to gain a better comprehension of signed languages.

Millet argues that the phonological elements that are hand configuration, hand placement and hand movement [9] can be analyzed on several levels: phonological, sub-lexical, lexical and syntactic. She thus decomposes what she calls "the iconic mechanisms of LSF" *(les dynamiques iconiques de la LSF)* according to the phonological elements. Table 3.1 summarizes this evolution for the three components.

For the **hand placement** component :

1. At a phonological level: placement is called **anchoring** *(ancrage)* for Millet. The anchoring is the placement of the sign in its citation form. It can be in the neutral

9. Millet does not consider NMFs as phonological elements but rather as phonetic elements. As for orientation, according to her, it is often too constrained by the movement and the physical limits of the body to have a semantic meaning.

Level	Hand Placement	Hand Movement	Hand Configuration
Phonological	Anchoring	Articulatory motion	Configuration
Sub-lexical	Derivative base	Iconic flexions	Derivative base
Lexical	Spatialization	/	Size and shape specifiers
Syntactic	Locus	Trajectories, pointing	Proform

Table 3.1 – Iconic mechanisms of LSF according to [1].

space in front of the signer or at a precise place location on the body of the signer.

2. At a sub-lexical level: placement can create a **derivative base** *(base dérivationnelle)*. A derivative base designates the signs with a similar component that have similar meanings. For example, many signs situated on the side of the forehead will have a meaning related to a psychic activity (e.g., [*PENSER*] (to think) or [*REVER*] (to dream), see Figure 3.5). A derivative base can also exist for hand configuration (e.g., the 'V' configuration is often used for activities where the eyes intervene like [*REGARDER*] (to watch), [*LIRE*] (to read) or [*DEVISAGER*] (to stare at)).

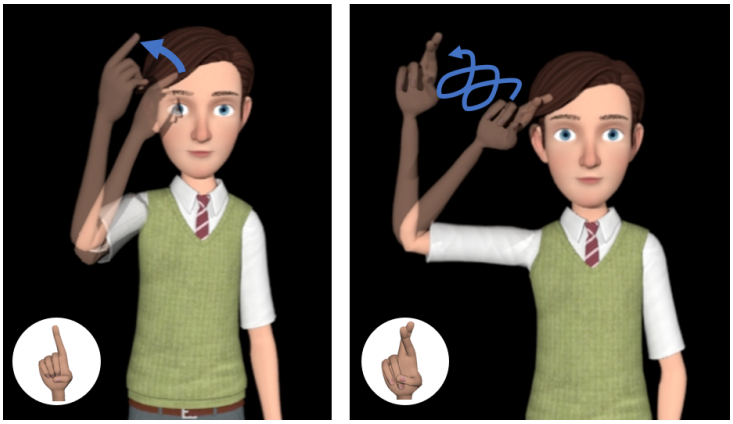

Figure 3.5 – Left: the sign [*PENSER*] (to think). Right: the sign [*REVER*] (to dream).

3. At the lexical level: when the anchoring is in the neutral space, it is possible to change the location of the sign according to its context in an utterance. This is called **spatialization**.
4. At the syntactic level: the **locus** is a $3D$ location in the signing space that is assigned to virtual objects and that can be used to refer to these objects. Associating virtual entities to $3D$ locations in the signing space makes it possible to give a relative placement of one entity with respect to the other (in the context of a description for example) or for a future referencing of these entities [13].

For the **hand movement** component :

1. At a phonological level: hand movement is the **articulatory motion** (*articulateur geste/sens* in [1]) that makes it possible to distinguish between two signs. For example, [*CAROTTE*] (carrot) and [*FILM*] (movie) have the same hand placement and hand configuration. Only the motion (circular for [*FILM*] and straight and repeated for [*CAROTTE*]) differentiate the two signs (see Figure 3.6).

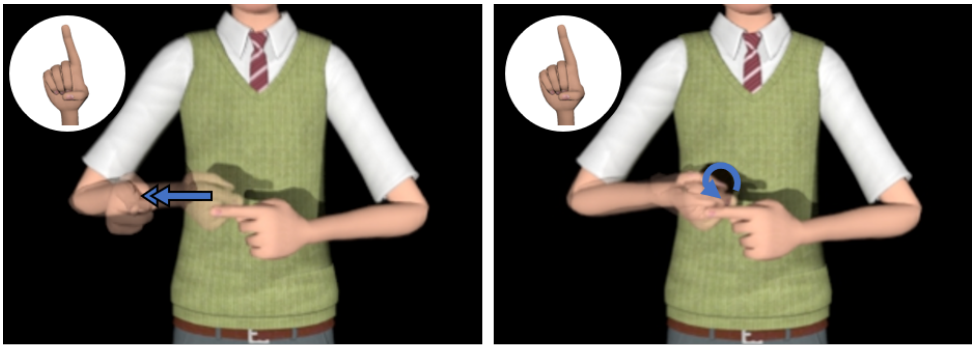

Figure 3.6 – The signs [*CAROTTE*] (carrot) and [*FILM*] (movie) can be distinguished thanks to their articulatory motion.

2. At a sub-lexical level: changing the movement of signs can create **iconic flexions** *(flexions iconiques)*. This is often used to describe the way of moving various objects/animals/persons: with the same configuration and placement but different movements, a signer can show the difference between the flight of an insect and the one of a plane.

3. At the syntactic level: the trajectory, or **motion path**, of some signs (notably the "indicating verbs" [13]) can change according to the relation between the described entities. The hand movement corresponding to the verb [*DONNER*] (to give) in the sentence "I give him" will not be performed in the same direction as the same verb [*DONNER*] in the sentence "you give me" (see Figure 3.7).

   The second syntactic function of hand movement is **pointing**. Pointing gestures consist in using the hand (often, the tip of the index) to designate an entity or a location. It can be used to indicate the subject(s) or object(s) of an action ([I], [YOU], [THIS BUILDING OVER THERE]) or to associate virtual objects to $3D$ locations in the signing space in order to initiate a locus or to (re-)activate one.

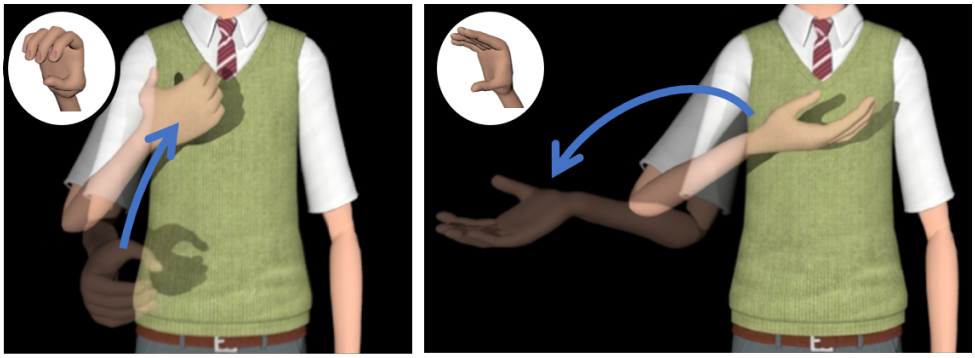

Figure 3.7 – Indicating verbs: hand movement and configuration for the sentences "you give me a glass" (left) and "I give him a book" (right).

> Pointing gestures are paramount in utterance production and in SL linguistics in general as it is often one of the first "signs" made by children [23].

The different functions of the **hand configuration** component (derivative base, size and shape specifiers, proforms) were explained in the previous paragraphs and in Section 3.2.1 and Section 3.2.2.

Millet thus proposes a classification of the linguistic mechanisms of LSF according to the phonological component and the granularity of the language study (phonology, morphology, syntax) without a separation between standard and illustrative signs. We find that this theory satisfactorily encompasses the parametric and iconic aspects of the LSF. We now show that our synthesis work is in line with this work.

3.3 Synthesis Objectives

The final goal of this thesis is to overcome the main limitation of data-driven synthesis techniques by using annotated motions present in an LSF *MoCap* database to synthesize novel LSF movements, signs and utterances absent from this initial database. In other words, we seek to increase the expressive capacity of our initial LSF database with synthesized motions corresponding to linguistic objectives.

With this purpose in mind, this section briefly presents the synthesis objectives of this thesis in accordance to the LSF linguistic theories described above. We therefore show that our objectives are linguistically relevant as they correspond to linguistic realities. Two

main synthesis approaches are described here: (i) the sign construction from phonological recombination, including the generation of inflected signs, and (ii) the utterance synthesis. We succinctly describe our approaches and list the main challenges that we address in this thesis.

3.3.1 Sign Construction from Phonological Recombination

The construction of signs by phonological recombination consists in creating signs, whether inflected or in their citation form, by combining, on one virtual character, the phonological components whose values have been fixed. In this case, we consider that a sign is the exclusive result of the values taken by its phonological components (see Figure 3.8).

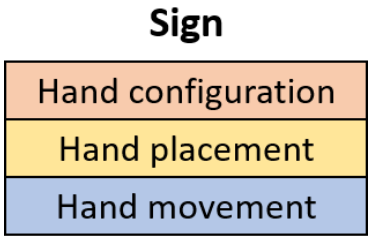

Figure 3.8 – A sign is the result of the values taken by its phonological components. The distinct components can be directly retrieved from a *MoCap* database or synthesized from scratch. They are then recombined to form a sign.

For example: the sign [*CAROTTE*] (carrot) is the result of assembling the 'index' configuration for both hands with a placement in front of the signer and with a rectilinear and repeated motion (see Figure 3.6, left). Note that a component can take several successive values in one sign. Thus, for the [LSF] sign, the hand configuration takes three values: 'L', 'S' and 'F' (see Figure 3.9).

3.3.1.1 Main Challenges of Phonological Recombination

The parametric approach has a definite advantage for movement synthesis and signing avatar animation because (i) the decomposition of language into "atomic" elements allows each component to be treated independently before synchronizing the different channels, (ii) this theory offers a way to represent a language production in the form of a set of targets to be reached, and (iii) the fact that some sets are finished allows to create movement databases whose elements can be recombined for sign formation (for example,

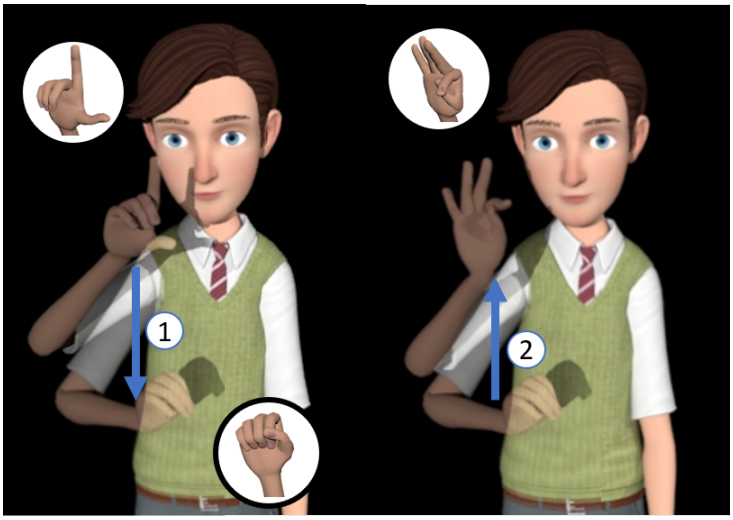

Figure 3.9 – The sign [LSF]: 'L', 'S' and 'F' configurations with a descending (1) and then ascending (2) hand movement.

it is possible to capture all the hand configurations if they have been listed beforehand). Still, the placement and the movement components can take an infinite number of values and must be synthesized from scratch for each new sign or situation.

In practice, considering a database annotated at a phonological level (i.e. the hand configuration, placement, movement and orientation are annotated on different annotation tracks), the sign that we wish to synthesize must first be decomposed into a sequence of values taken by the phonological components. Then, the numerical values taken by the joints are extracted from the database for each channel [10] taking full advantage of motion retrieval techniques from an annotated database. Each channel is processed independently, then, the different channels are synchronized to obtain the sign.

When implementing a phonological recombination system, we can therefore note several challenges:

1. **Presence of real data.** The construction of signs by recombination of the phonological components assumes the presence of a database containing realistic LSF movements from, preferably, more than one distinct signers to gain in variability and have more representative samples. As we are indeed seeking to reassemble the phonological components present in the database to form new signs, the components must be precisely defined and various values of each component must be present in

10. We call "channel" the set of joints of the animated model corresponding to a phonological component. Thus, the hand configuration channel corresponds to the joints of the fingers. The channels are detailed in Section 8.1.3.3.

order to have a significant coverage of signs.

2. **Data annotation**. To be used for synthesis, the captured data stream must be segmented and labeled in such a way that the movements of a channel corresponding to a specific value can be retrieved. In order to be used in phonological recombination, the annotation of the data must be made at a phonological level.

3. **Definition of a mapping** between the phonological elements and the joints of the model to be animated (i.e. the channel). The use of a captured database and the objective of animating an avatar forces the definition of a mapping between each phonological component and a set of joints of the avatar. Indeed, if we want to extract the value taken by a component from the database to replay it on an avatar, it is necessary to define a mapping between the component in question (e.g., the hand configuration) and the joints related to this component (e.g., the joints of the fingers).

4. **Representation of the objective**. In order to specify the sign to be synthesized, the values taken by the different parameters of the sign must be made explicit. This representation can, in addition, contain information on the synchronization of the elements between the body channels.

5. **Intra-channel coarticulation**. During the realization of a sign, the change from one component value to another (e.g., a change of hand configuration) must be managed. These articulatory movements are often absent from the database given the huge number of possible combinations [11], it is thus necessary to synthesize them in order to obtain a realistic movement. Different synthesis techniques can be applied as we will see in Section 8.2.2.2.

6. **Synchronization of the body channels**. To obtain the desired meaning, the channels must be synchronized precisely. On a given channel, the determination of the precise timing to reach the different values is important. In addition, relative synchronization between the different channels is also essential to obtain both a semantically correct sign and a realistic movement. The study of LSF data allows to find empirical laws of human movement that can be implemented to synthesize credible movement.

11. Considering the configuration channel only, if we set the number of possible configurations to 40 and assume that each one is always executed in the exact same way, there are $39 * 40 = 1560$ possible combinations.

The challenges 1 and 2 are the subject of separate chapters (Chapter 6 for the data, Chapter 7 for the annotation). In Chapter 8, we detail the challenges 3 to 6 and present the solutions implemented to address them. Those challenges are common to the synthesis of signs in their citation form and to the synthesis of inflected signs.

3.3.1.2 The Specific Case of Sign Inflections

The transformation of the citation form of some signs to take into account the contextual information of an utterance is called *sign inflection*.

The sole synthesis of signs in their citation form mainly aims at building signs for bilingual dictionaries or educational applications while translation or storytelling applications require the generation of full utterances. However, for the synthesis of utterances, the addition of inflection mechanisms is needed.

An analogy with spoken languages can help to understand what inflected signs actually are. In many spoken languages, words have a different form when they are isolated and uncontextualised than when they are used in a sentence: verbs are conjugated and some words are put to the singular/plural form, for instance. Inflected signs are similar to those contextualized words: their form is impacted by the context. Under the "sign inflection" label, we regroup the three iconic mechanisms of Cuxac [3] as well as the spatial referencing mechanisms of Millet, namely the spatialization, locus, motion path and pointing mechanisms [1]. Figure 3.10 shows two examples of inflections for the sign [*BOL*] (bowl).

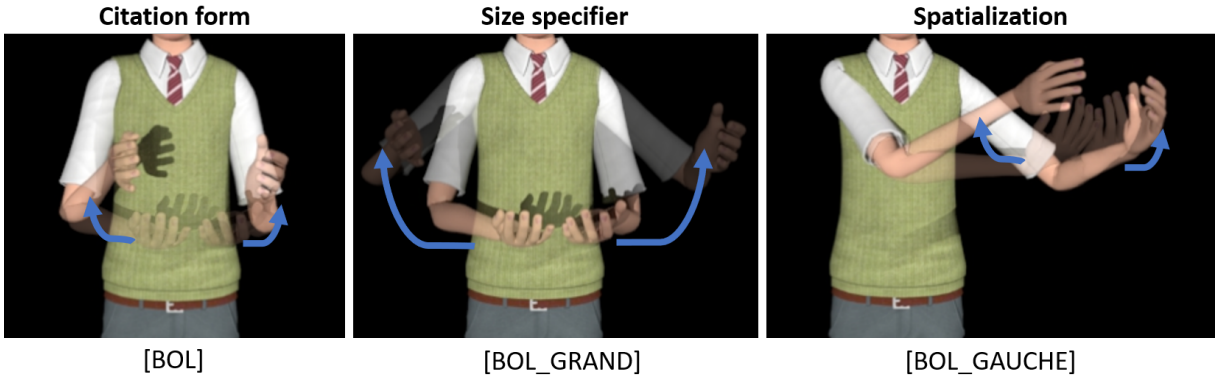

Figure 3.10 – The sign [*BOL*] (bowl) in its citation form (left) and two of its inflected forms. A big bowl can be signed with a size specifier (middle) and a bowl can be located at the left of the signer with a spatialization (right).

Starting from the content of a database from which we will be able to extract knowledge (empirical motion laws) and numerical data to be re-used, it is possible to modify captured signs to build their inflected version or new inflected content absent from the database using phonological recombination. Here again, the modification of some of the values taken by the phonological components for different signs leads to the creation of inflections.

Different LSF mechanisms defined in Section 3.2.3 are targeted here (the main component involved is inside the parenthesis): **spatialization** (hand placement), **pointing gesture** (hand placement), **motion path** (hand movement), **size and shape specifiers** (hand configuration and movement) and **proforms** (hand configuration and movement).

The challenges listed in the previous section apply to the synthesis of inflected signs. Other challenges of the synthesis of inflection mechanisms are specific to the mechanisms involved. However, we list here two issues that affect most inflection mechanisms:

1. **Context awareness:** by definition, the context of the signs must be taken into account to build inflected signs. For instance, as mentioned above, indicating verbs take a different form when put into an utterance. Another example are proforms which take advantage of the hand configuration and movement to embody an object or a person. They are used to describe an infinite number of situations and are not suited for isolated sign synthesis but are very interesting in utterance synthesis.
2. **Space representation:** the signing space is an area in front of and around the signer in which he/she can place the entities of his speech. In this space physically limited by the signer's reaching capabilities, he/she will be able to describe an infinite and constantly changing space. Interestingly, space will also be the carrier of temporal information: a movement of the torso backwards or forwards makes it possible to place events in the past or the future. Space is very important for the synthesis of inflection mechanisms as a large part of these mechanisms relies on the specification of spatial targets. When doing motion synthesis, the appropriation of the signing space results in the definition and naming of $3D$ areas around the signer. The creation of sign language content for avatar animation requires this discretization of the signing space

3.3.2 Utterance Synthesis

An utterance is constituted of a set of signs performed sequentially or simultaneously. Both signs in their citation form and inflected signs must be used to compose utterances.

Indeed, the signs in their citation form are not enough for utterance synthesis: a simple concatenation of those signs to create an utterance does not do justice to the richness of LSF and can generate incorrect utterances. The multiple variations of illustrative signs must be taken into account just as well as the fixed citation form of signs.

In the signing avatar community, utterances are often built by concatenating synthesized signs (see Section 5.3). This solution, if rightly managed, is suited to the sequential aspect of utterances and can be sufficient to synthesize simple, factual utterances. However, translation or storytelling tasks require more complex sentences involving the simultaneous aspects of sign language utterances. Both hands, for instance, can be used in parallel to perform two different signs. An example of two signs performed at the same time is visible on Figure 3.11.

Figure 3.11 – Example of two signs performed at the same time for the description of a scene: here, a car, symbolized by a 'flat_hand' proform (left hand), drives past a building (right hand).

We propose to use the knowledge extracted from the captured data to improve the synthesis of the sequential and simultaneous aspects of utterances. The focus of this thesis is the synthesis of a correct and realistic motion and, by no means, machine translation. We thus consider that we possess the knowledge of the nature and ordering of the signs that compose an utterance. However, some challenges specific to the generation of meaningful utterances still remain:

1. **Analysis material:** in order to improve the realism of the synthesized utterances, real data is again required. A *MoCap* database containing various utterances can

be exploited for this purpose.

2. **Transitions:** an utterance is, at least partially, a sequential phenomenon: signs are executed one after another. As such, the motion marking the transition from one sign to the next must be generated carefully to preserve the dynamics of the utterance.
3. **Coarticulation:** the form of a sign will be impacted by the previous and next signs. In spoken languages, when pronouncing words, speakers make a *liaison* between the end of some words and the beginning of the following. More generally speaking, the pronunciation of a word will slightly impact the pronunciation of the neighboring words. This same coarticulation mechanism exists for sign languages and is expressed both in the transitions between the signs and in the form of the sign itself. To have a more natural sign language flow, the transition motion between signs must be generated carefully.
4. **Simultaneity:** some signs can be performed in parallel with one another. A simple concatenation of signs is thus not sufficient for the creation of many utterances, particularly when describing a scene or in situations of storytelling that involve more visual phenomena. It can thus be interesting, in some situations, to manage the hands (and the non manual features) independently from each other.

Those challenges, whether at a sign or utterance levels, will be addressed in the remainder of this thesis.

3.4 Summary and Discussions

Our thesis lies in the computer science field: it consists in animating virtual characters with believable movements. In this sense, we seek to develop annotation techniques and data-driven synthesis methods that provide acceptable quantitative and qualitative results for our field of application: *signing* avatars. Thus, an additional constraint relating to this specific field appears: the synthesized movements must respect the linguistic realities of LSF.

This chapter has laid the linguistic foundations of our work to highlight the main concepts and issues that we will meet when animating LSF avatars. First, we presented the socio-cultural context of our work. Indeed, knowing that the final field of application of our work targets a specific audience, the members of Deaf communities, it is important to

have a general knowledge of its history. Then, three different linguistic theories of LSF were summarized in order to present the linguistic basis of our work. Finally, we presented the main challenges of sign and utterance synthesis. We make a parallel between the linguistic theories and our synthesis objectives to assert the relevance of our work.

At the sign level, our work is based on the phonological components defined in Section 3.2.1 whether by combining them to create new signs or by modifying them to add iconic information. In this sense, we approach Millet's linguistic theory, which gives phonological components a central place [1]. We name this approach "phonological recombination".

At the utterance level, we propose to use the knowledge extracted from a database of captured motion to improve the synthesis of transitions between signs and to add simultaneous phenomena to our utterances.

In the two discussed approaches, the sign construction and the utterance synthesis using inflected signs, an accurate database annotated at a phonological level is needed. Chapter 6 deals with the challenges related to captured data and the presence of analysis material. Chapter 7 presents our choices and solutions for annotating this database. Finally, Chapter 8 deals with all the challenges relative to the motion synthesis itself.

We have chosen to focus our study on the manual phonological components that are hand configuration, hand placement and hand movement but we do not minimize the importance of non-manual components. Facial expressions and other body movements could be incorporated into our work.

Part I

State of the Art

Chapter 4

Collection and Annotation of Sign Language Data

Contents

4.1 Existing Sign Language Corpora 48
4.2 Data Annotation 56
4.3 Summary and Discussions 69

> Nous sommes comme des nains assis sur des épaules de géants. Si nous voyons plus de choses et plus éloignées qu'eux, ce n'est pas à cause de la perspicacité de notre vue, ni de notre grandeur...
>
> Bernard de Chartres, 12th century

In order to animate sign language avatars from real human data, an annotated *Motion Capture* database is needed to provide the raw data to be analyzed and reused for motion synthesis. Section 4.1 lists and discusses the existing sign language corpora. Then, Section 4.2 presents the different manual and automatic techniques for annotating a SL database.

4.1 Existing Sign Language Corpora

Given the lack of exhaustive and widely accepted written systems for sign languages, only video cameras or *Motion Capture* (*MoCap*) technologies can provide sign language recordings accurate enough to be used for analysis and/or synthesis.

4.1.1 Video Corpora

Video recordings constitute the most common source of data. The subjects are filmed with RGB cameras from one or more points of view, data is stored using a standard video format and is annotated *a posteriori* following a pre-defined annotation template. Videos can be used for the study and analysis of SL but do not provide reusable data for SL synthesis.

Video corpora are often the base material for statistical studies to highlight a particular gestural phenomenon or to verify a given hypothesis on human motion. Various SL video databases have been designed by linguists in order to study a specific **linguistic mechanism** to compare sign languages with each other or simply to store and archive signed statements.

Directional verbs and the use of space in SL are often studied and some corpora have been collected for this purpose [24]. Coarticulation is also a recurring study topic ([25] for the coarticulation of Dutch Sign Language hand configurations, [26] to compare the effects of coarticulation in Finnish Sign Language and oral language). For French Sign Language, iconicity and role shift were studied in the LS-COLIN corpus [27], pointing gestures made by small children were studied in the corpus *Illana* [23], [28] and classifiers in narratives/stories were analyzed in Millet's corpora [1], [29].

Linguistic corpora can also be used to compare SL with each other or to detect regional variations. In [30], an approach to detect variations in the execution of the same sign in German Sign Language is proposed. In [31] and [24], five Australian Sign Language dialects were analyzed and the same was done in Swedish Sign Language in [32]. In [33], the French Sign Language used by children and adults is compared. In [34], the prosodic cues of French Sign Language and Quebec Sign Language native signers and interpreters are studied.

In other cases, the corpora have been collected in the sole purpose of preserving the language without any linguistic analysis in mind. This is the case of [35] for the Sign Language of the Netherlands.

Table 4.1 lists some video corpora designed for linguistic studies. A more thorough comparison of the video corpora for Sign Languages until 2012 is available on the website of the University of Hamburg [1].

1. `https://www.sign-lang.uni-hamburg.de/dgs-korpus/files/inhalt_pdf/SL-Corpora-Survey_update_2012.pdf`

On the other hand, some corpora have been designed specifically by computer science communities for **natural language processing** (NLP) and **computer vision** tasks. While many linguistic corpora attempt to capture a large number of different signs to try to cover the whole language, NLP corpora are focused on a limited set of signs repeated numerous times with many variations (for example, by varying the location of the sign in space). Moreover, the need for a natural flow of SL is less important: to compose NLP corpora, elicitation of specific signs is often preferred to free conversation in order to provide robust training data for recognition and statistical machine translation tasks.

Some corpora were designed for computer vision tasks, particularly sign language recognition [36]. The RWTH-BOSTON-400 [37] and the RWTH-PHOENIX-Weather-2014 [38] corpora can be easily divided into training and test data sets for recognition tasks. The SIGNUM corpora [39], in German Sign Language, contains 450 elicited signs performed by numerous different signers in order to carry out a cross subjects recognition task of isolated signs. For more recent SL recognition corpora [40], [41], depth information is added to the video recordings using Kinect devices.

Other corpora aim to solve Statistical Machine Translation (SMT) tasks. The number of corpora for SMT between a sign language and a corresponding spoken language is quite important. Among them, we can list corpora for the translation between written Chinese and Taiwanese Sign Language [42], Arabic and Arabic Sign Language [43], English and Irish Sign Language [44] or Spanish and Spanish Sign Language [45].

However, few corpora exist for SMT between two different sign languages. The ATIS corpus [46] focuses on a small set of sentences to develop SMT systems between 5 languages (English, German, Irish Sign Language, German Sign Language and South African Sign Language). The European Dicta-Sign Project [47], [48] aims to build an "inter SL" dictionary and to perform SMT tasks between 4 sign languages: German, French, Greek and British sign languages. This project also seeks to develop new annotation tools for sign language analysis.

Hybrid corpora designed for both NLP and linguistic studies exist. For Greek Sign Language, the corpora of [49] is as broad and representative as possible in order to meet the requirements of both communities.

Table 4.2 lists and compares different video SL corpora designed for NLP tasks.

The cost of a video recording session is quite cheap as a single video camera com-

mercially available may be sufficient. However, a video recording alone eliminates the third dimension of space. Pose estimators, such as *OpenPose* [50], which can be badly impacted in the case of SL by occlusions between the two hands, or additional cameras are needed to compute the depth information during the realization of the signs. Besides, video recordings rarely possess a spatial resolution and a frame rate high enough to allow a precise data segmentation and analysis.

4.1.2 *Motion Capture* Corpora

Motion Capture, or *MoCap*, is a motion acquisition technique which can be based on passive or active systems. In passive systems, the position and displacement of reflective markers placed on an actor's body and/or face are measured using infrared cameras. In active systems, different types of sensors, whether inertial, magnetic or mechanical, are used to compute the motions of the actor.

MoCap technologies offer a higher spatial and temporal resolution than $2D$ video cameras in exchange for the need for a greater technical expertise and a rigorous post-processing. Data resulting from a *MoCap* session can be used for SL analysis: precise quantitative motion descriptors can be computed from the $3D$ data to confirm or reject existing linguistic hypotheses or motion laws. As the $3D$ positions of the human skeleton joints can be inferred from the *MoCap* data in a systematic way, avatars can also be animated from the captured data which constitutes a major advantage compared to video data.

While a vast majority of video corpora are designed for linguistic analysis, *MoCap* corpora purposes are more evenly distributed. Some corpora were designed with **linguistic and kinematic analysis** objectives in mind – coarticulation analyses [51], [52], study of the signing space and of the indicating verbs [53], prosody [54], syntactic analyses [55], [56] or studies on motion laws [56], [57]. Others have been used for SL **automatic annotation** and SL **recognition** [58]. Contrary to NLP video corpora that mainly aim at doing statistical machine translation, few *MoCap* corpora were done with this sole purpose [59]. Finally, some *MoCap* corpora were designed specifically for **data-driven synthesis** (e.g., using concatenative synthesis for French Sign Language [60] or a deep neural network model for Japanese Sign Language [61]). The CUNY corpus of American Sign Language [53] aims both to study linguistic mechanisms and to build a linguistic representation that can be used to animate an avatar. The sign language *MoCap* corpora are listed in Table 4.3.

However, even though their number is steadily growing, *MoCap* databases for SL studies are still rare and only a small portion of them are made available. Besides, *MoCap* databases are small compared to video databases: they rarely exceed one hour of data and contain the signed utterances of few different signers (often only one signer), while a video footage can last hundreds of hours and gather the data of various persons. As a consequence, each of the existing *MoCap* corpus has been designed for a specific purpose.

In this thesis, we aim to study, annotate and synthesize the different phonological components of signs in order to build signs and utterances in French Sign Language. We need an available *MoCap* French Sign Language corpus, composed of isolated phonological elements, isolated signs and full utterances. No corpus met all those constraints so we designed the *LSF-ANIMAL* corpus to meet our specific needs. This corpus is presented in chapter 6.

Corpus Name (SL)	#Signers	Duration	Content	Objectives	Annotation	Available
LS-COLIN [27] (French SL)	13	2h12	Narratives, expression on a given topic, teaching a course	Study of classifiers	Glosses, phono. elements	Yes
Corpus NGT [35] (SL of the Netherlands)	73	12h	Narratives and elicited lexical items	Preservation of the language	Glosses, translation	Yes
The Auslan SignBank (ELDP & SVIAP) [24], [31] (Australian SL)	100 + 211	300h + 140h	Free conversation, interviews, elicited lexical data, and narratives	Study of the sociolinguistic variations, cross-linguistic data for comparison, study of indicating verbs	Glosses, classifiers, pointing gestures	Yes
CREAGEST [33] (French SL)	Approx. 150	Approx 130h	Dialogues between children/adults, between two adults, free conversation, narratives	Study of child signing, of coverbal gestures, preservation of the language	Iconic mechanisms, pointing gestures, gaze, unit of meaning	NC
BSLCP [62] (British SL)	249	Approx. 1h per participant	Free conversation, interview, lexical elicitation, narrative.	Studies of lexical frequency and sociolinguistic variation in BSL	Glosses, phono. components, translations	Yes
SSLC [32] (Swedish SL)	42	24h	Free conversations, elicited narratives	Research on language structure and dictionary	Glosses, translations	Scheduled
DGS Corpus [30], [63] (German SL)	330	560h	Free conversations, elicited lexical items, narratives	Linguistic studies, detect regional variants	Glosses	Yes
LSF corpus A/B [1], [29] (French SL)	More than 20	NC	Conversation on broad subjects, free conversation, narratives, elicited grammatical mechanisms	Linguistic studies of grammatical phenomena	Glosses, phono. components	No

Table 4.1 – Some video SL corpora designed for linguistic studies.

Corpus Name (SL)	#Signers	Duration	Content	Objectives	Annotation	Available
TSL Corpus [42] (Taiwanese SL)	NC	NC	2000 utterances	Statistical machine translation (Chinese, Taiwanese SL)	Glosses, utterances, phono. components	No
GSLC [49] (Greek SL)	4	18h	Free narratives, manual and non manual features, elicited utterances	SL recognition and linguistic studies	Glosses, utterances, phonological levels, translation	No
SIGNUM [39] (German SL)	25	More than 40h	450 elicited signs	SL recognition	Glosses, phono. components, translation	Yes
RWTH-BOSTON-400 [37] (American SL)	4	NC	843 utterances	Automatic SL recognition and machine translation	Glosses	On request
ATIS [46] (Irish SL, German SL, South African SL)	6	NC	595 sentences on air travel in 5 languages	Statistical machine translation and sign language analysis	Glosses, spatial references	On request
Dicta-Sign [47], [48] (German SL, Greek SL, British SL, French SL)	16 to 18 per language	Approx. 6h per participant	Utterances and elicited signs	Multilingual lexicon, annotation tools, machine translation, synthesis material, SL recognition	Glosses, translations	Yes
RWTH-PHOENIX-Weather [38] (German SL)	9	Approx. 10h	Weather reports	Statistical machine translation and SL recognition	Glosses, phono. components, translation	Yes
BosphorusSign [40] (Turkish SL)	10	NC	Elicited signs and utterance samples from three domains	SL recognition	Glosses, phono. components, translation	Samples available
SMILE [41] (Swiss-German SL)	30	NC	300 elicited signs per participants	SL recognition for educational purpose	Glosses, phono. components, translation	Scheduled

Table 4.2 – Video SL corpora designed for NLP and computer vision tasks.

Corpus Name (SL)	#Signers	Duration	Content	Objectives	Annotation	Available
TRAIN [54] (French SL)	1	10 min	Elicited utterances about train departure and arrival	Synthesis by sign replacement	Glosses	Scheduled
METEO [54] (French SL)	1	10 min	Elicited utterances about the weather	Prosody analysis	Glosses	Scheduled
SignCom [64] (French SL)	2	1h	Elicited utterances	Synthesis by modifying phonological components	Glosses, phono. components	Scheduled
CUNY ASL [65], [66] (American SL)	8	3h30	Free conversation on 12 different subjects	Analysis of linguistic mechanisms for data-driven synthesis	Glosses, spatial referencing	Partially
KWS avatar [67] (Swedish SL)	1	10 min	150 elicited signs	Study of *MoCap* applied to sign language recordings, hand reconstruction	Glosses	No
MOCAP1 [55], [56], [68] (French SL)	8	1h	Description of images	Syntactic analysis, kinematic analysis	Glosses, gaze, right hand motion	Yes
Sign3D [69] (French SL)	1	10 min	Elicited utterances about buildings	Synthesis by sign replacement, SL analysis	Glosses, phono. components	No
BP-Libras [59] (Brazilian SL)	1	8h	Elicited science textbook recordings	Machine translation	Glosses, phono. components, inflections, translations	No
HRI JSL [61] (Japanese SL)	1	Not communicated (10.384 signed utterances)	Elicited utterances including inflected signs, indicating verbs, syntactic NMFs	Study of mechanisms to "augment" a corpus of sign language	Glosses	No

Table 4.3 – *MoCap* sign language corpora.

4.2 Data Annotation

Whenever data is used, whether for analysis or synthesis, it is necessary to be able to manipulate it. Data annotation is the process of associating a piece of data with one or more labels describing its content, often with words of natural language. Annotation thus adds a semantic layer between the raw data and the human and allows to group together, under the same label, elements whose similarities are not obvious from the data alone. These labels can be queried to access the corresponding raw data which can then be used for analysis or synthesis.

4.2.1 Terminology

The annotation is a three-step process. First, an *annotation scheme* is defined to specify the structure of the annotation. Then, a second step, called *segmentation*, consists in dividing the stream of data into segments or regions of interest. Those segments are then identified in a third step, called *labeling*.

4.2.1.1 Annotation Scheme

The annotation scheme is a high-level framework which gives the overall annotation structure. The level of details of the annotation scheme greatly influences the way an avatar will be controlled. The annotation scheme defines:

— The different **tracks** of the annotation and, therefore, the **granularity** (phonological, semantic, syntactic...) of the annotation. For example, for SL annotation, a typical annotation scheme will define multiple tracks:

Track #1 - Gloss (semantic/morphological granularity),

Track #2 - Right hand configuration (phonological granularity),

Track #3 - Left hand configuration (phonological granularity),

Track #4 - Right hand placement (phonological granularity),

...

— If needed, the **relationship between the tracks**. Dependency relationships or a hierarchy between the tracks can be clarified. In a dependency relationship, the temporal limits of the higher level segment must encompass or exactly correspond to the limits of the lower level segments.

— The **vocabulary** that is used to label each track. It can be a closed vocabulary if the track corresponds to an element taking discrete values in a finite set (like the hand configuration), or it can be an infinite or very large vocabulary for other tracks (like the gloss track that can take virtually any value). In the first case, the vocabulary is defined in the form of an enumeration of the possible values while, in the second case, the vocabulary is defined in the form of rules describing the content of the label (for example, "glosses must not be conjugated" or "the placement of the hand must be indicated with numerical values, in centimeters and relative to the placement of the hips").

— The **format** of the labels. Depending on the track, the labels must be written following different formalization rules. For example, glosses are often written within brackets and with uppercase letters while hand configurations use lower-case letters. For automatic processing of the annotation, spaces between words in a label are often proscribed and replaced with underscores. For the same reason, the use of accents or other special characters is not recommended.

As an example, the technical report of Jonhston provides a thorough annotation scheme for the annotation of sign language data [70].

4.2.1.2 Segmentation

Two types of segmentation exist depending on the nature of the data to annotate. *Temporal segmentation* consists in segmenting time-dependent data – such as videos, sound or any other time series – along a time axis. *Spatial segmentation* allows the segmentation of images or other non-time-dependent data by defining regions of interest (detection of trees, animals or cars on an image) (see Figure 4.1).

Depending on the type of the input data, SL segmentation can involve both types of segmentation. For video data, computer vision techniques are usually needed to first perform spatial segmentation and tracking on the different body parts (hands, face, arms...) to extract the positions of the joints over time (i.e. skeleton-based approach). Then, the data, originating either from video or *MoCap*, can be considered as one (or more) time series requiring a temporal segmentation that will result in the definition of labeled time segments. As computer vision tasks are not the focus of this thesis, the remainder of this section is about temporal segmentation.

The segmentation of human motion is the process of breaking a continuous sequence of motion data into smaller and meaningful components that range from actions to move-

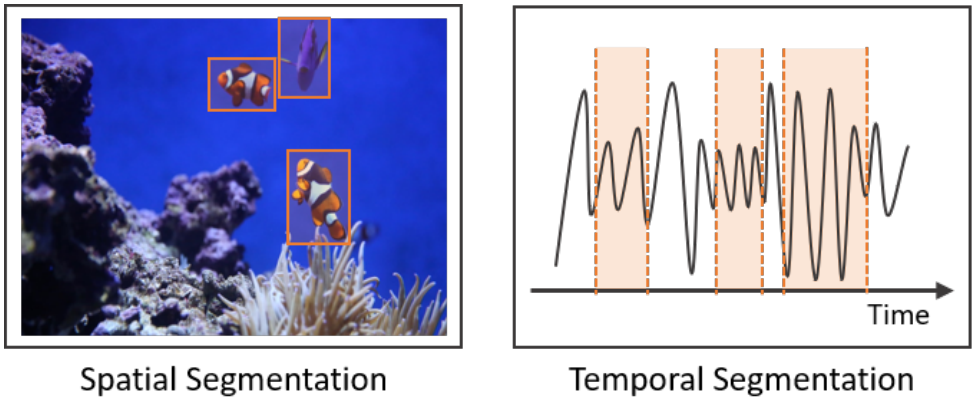

Figure 4.1 – The spatial (left) and temporal (right) segmentations. The segments or regions of interest are in orange.

ment primitives. Specifically, it consists in identifying the starting and ending frames of each segment. For sign language movements, the segments depend on how the linguistic element boundaries are defined, following phonetic, phonological and semantic rules, as well as coarticulation between signs. The granularity and thus the selected tracks to be segmented depend on the final use of the data. For SL synthesis by phonological recombination, a segmentation at a phonological level is required, while the synthesis of some inflected mechanisms asks for a higher level segmentation (gloss or part of utterance). The generation of new sign language content heavily relies on the quality of the segmentation.

4.2.1.3 Labeling

The segmentation allows to define the segments of interest. The labeling step is a recognition step: the content of the segments is identified (manually or automatically) and named according to the annotated track with a tag for which the value and format are specified in the annotation scheme. Confusion between tags is the main source of errors of the labeling step. For gesture and SL recognition, on $2D$ video data, the errors can be caused by the lack of depth information, the color and lighting of the background and of the actor. For automatic SL recognition, the inflections of signs, the change in position and orientation of the hands and the coarticulation effects are other recurrent sources of confusion.

4.2.2 Manual Annotation

The annotation is often done manually. In this case, annotation experts use an annotation tool to visualize the data and to define and identify the segments of interest. The resulting annotations can be used both to analyze the data and to synthesize new content.

4.2.2.1 Tools

Different tools like ELAN [71], [72] [2] and ANVIL [73] [3] are freely available to annotate a stream of temporal data. They provide a graphical user interface to create, visualize and edit the annotation. Both ELAN and ANVIL can be used to create annotations on multiple tracks (called *tiers* in ELAN and *tracks* on ANVIL). Constraints like track dependencies or closed vocabulary are directly implemented in the tools reducing the risk for human mistakes during the annotation process. Both tools are generic as they can be used to annotate different types of time series like sound recordings or video footage following different annotation schemes. They are both XML-based and platform independent. Spreadsheets or homemade software are also used for data annotation.

4.2.2.2 Limitations

Manual annotation is a fastidious, time consuming and expensive task as it requires annotation experts that are skilled both in annotation and in the domain of the data. Manual annotation is subject to inaccuracies and mistakes as the experts may not have exactly the same segmentation criteria. Finally, the annotators can be subject to fatigue resulting in confusions in the labels.

4.2.2.3 Specificities of Sign Language Annotation

Sign language annotation requires the definition of an annotation scheme providing tracks for all the needed linguistic elements of SL. The precise definition of each track (including the often problematic gloss track) at a gesture and linguistic level is needed [74].The vocabulary used for sign language data annotation is often larger than for general motion annotation as it corresponds to a whole language. Therefore, the annotation process, including the definition of the annotation scheme, the segmentation and

2. http://tla.mpi. nl/tools/tla-tools/elan/
3. `https://www.anvil-software.org/`

the labeling, needs to be done by annotators both experts in sign language and gesture annotation, a combination that can be hard to find.

In addition, sign languages are expressed simultaneously on multiple tracks (hand configuration, hand orientation, body posture, facial expression, etc.), thus complicating the task of the annotators (see Figure 4.2). When comparing the duration of the annotation process to the duration of the data to annotate, Dreuw *et al.* [75] 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).

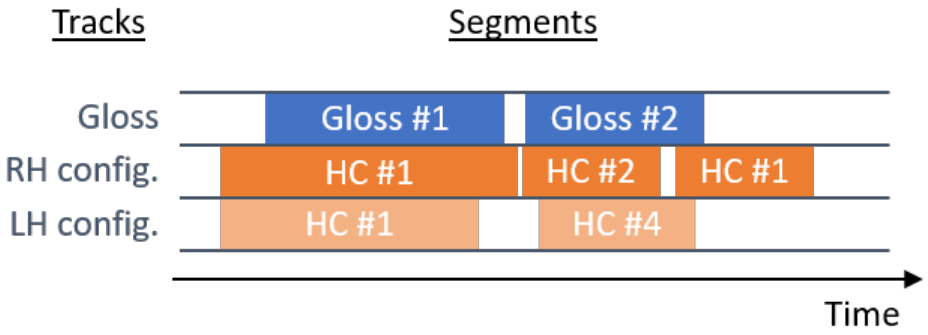

Figure 4.2 – Example of a multi-track annotation with three tracks (RH = Right Hand and LH = Left Hand).

Moreover, even with a precise annotation scheme, the segmentation is subject to variability due to the fact that all annotators do not agree with the starting and ending frames of the segments [76].

Finally, as an utterance consists in a continuous stream of signs, one signer may start the subsequent sign before fully completing the previous one. This coarticulation effect makes the segmentation and labeling more complex. Likewise, the contextual dependency causing sign inflections is also a source of errors in the labeling step.

Some of the challenges and possible solutions for annotating a sign language corpus were developed in a series of conferences about the *CREAGEST* project [33] [4].

The numerous limitations of manual annotation motivate the development of automatic annotation techniques. Automatic annotation of human motion data and, specifically of sign languages, could reduce the annotation costs but is still a challenging and yet to be solved task.

4. https://www.sourds.net/2017/11/29/enjeux-constitution-dun-corpus-moderne-de-langue-signes-francais

4.2.3 Automatic Annotation of Human Motion Data

For the remainder of this section, we consider the motion data as a set of joint positions over time. Each joint provides a time series corresponding to the joint position in the $3D$ space (see Figure 4.3). The processing step to obtain this data is not under the scope of this chapter.

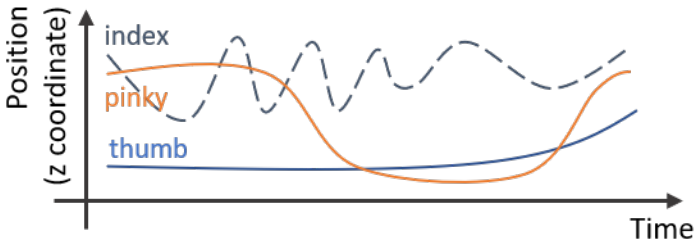

Figure 4.3 – Example of position time series for 3 joints: index, pinky and thumb fingers.

Whether we want to annotate general body motion or sign language motions, the goal is to annotate the movements performed by the human body. Most of the challenges are therefore common to both fields and, as a result, the techniques used are often similar. This section describes the work done for general motion data annotation while the annotation of sign language data is described in the following section.

4.2.3.1 Automatic Segmentation of Human Motion

The segmentation of human motion along a temporal axis consists in determining the temporal extents of motion segments in a continuous stream of data. The automation of such a task is challenging due to the high-dimensional nature of human motion data that comes from the high number of joints of the human body and the number of degrees of freedom of each joint, usually 3. Indeed, even with a simplified model of the human body, the number of degrees of freedom must remain high to preserve the distinctive features of human motion. Moreover, a given motion, for example a jumping motion, will vary when performed by different subjects or even by the same subject over time depending on the previous and following motions, the level of fatigue, the physical characteristics or specific abilities of the subjects. Human motions thus possess temporal and spatial variations that complicate their automatic segmentation.

After having defined the nature of the motion segments, automatic segmentation techniques can be taken advantage of to divide the continuous motion into motion primitives.

For motion primitive segmentation in general, a very complete framework is developed in [77]. 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 multiple track segmentation or the particular application of sign language processing. We propose a classification of some techniques and work on automatic human motion segmentation in the list that follows:

a) Techniques Based on Kinematic Features Computation These identify the segment boundaries by detecting changes in the variation or threshold crossings of kinematic features, such as position and orientation and their derivatives (velocity, acceleration, curvature, jerk). Zero crossings in the speed of specified joints are often used as it corresponds to a pause in the motion and thus often to the beginning of a new motion. [78] segments by detecting the zero crossings in the angular velocity of the arm joints to detect changes in the direction of a movement. [79] analyzes the changes in variation in the acceleration feature to determine the segment extremities.

If these methods are easy to implement and can be efficiently applied to various data sets, they may give over or under segmentation boundaries, they depend on *a priori* knowledge of the data and of the segments for the choice of features, thresholds and of joints of interest.

To reduce the need for *a priori* knowledge, the threshold techniques can be applied not directly on the kinematic features derived from the data but on **distances** between kinematic features. The Euclidean distance between joints [80]–[82], normalized or not, and the variance of the data [83] are two distances used for segmentation. Unsupervised Dynamic Time Warping (DTW) algorithms can be performed on a sliding window to compare the distance between time series in order to minimize the impact of temporal variations in a motion segment [84].

b) Unsupervised Statistical and Probabilistic Models Boundaries can be determined by statistical or probabilistic models, using data analysis principles such as Principal Component Analysis (PCA) [85], [86], Probabilistic PCA [85], Gaussian Mixture Models (GMM) [85], entropy based Hidden Markov Models (HMM) [87], deriving Bayesian methods [88], temporal application of Hilbert space embedding of distributions [89] or jointly using probability density functions (PDF) with HMM and the Viterbi algorithm for the resolution of the HMM [90], [91].

Barbič *et al.* [85], for example, use a sliding window on the data to compute its N principal components with PCA. Then, the following frames are reconstructed using those principal components: the error between the original and reconstructed data is computed and a segment boundary is defined if this error is higher than a threshold. This technique depends on this threshold and on the chosen N.

Those statistical and probabilistic models are often sensitive to a sliding window size, a threshold value and the model parameters but are less correlated to the specific data to annotate and are unsupervised; they do not require an *a priori* knowledge about the data.

c) Supervised Techniques The unsupervised models and kinematic approaches presented above are used to divide a motion into sub-motions without giving any information on the nature of the sub-motions. A labeling step is needed, in a second phase, to identify the segments. Other segmentation approaches use supervised techniques, such as template matching or classification approaches, to recognize the sub-motions from a continuous stream of motion data. In this case, the segmentation and labeling are done simultaneously given an *a priori* knowledge of the nature of the motions contained in a data set.

Knowing the profiles and labels of the motions present in a database, template matching algorithms like DTW make it possible to compare the known profile with the data despite temporal variations and thus to segment and label parts of the data. Zhou *et al.* [92] use Aligned Cluster Analysis (ACA), a combination of k-means clustering techniques with DTW to segment motion data. Müller *et al.* [93] use motion templates to identify related motions in a data set.

The segmentation by classification using machine learning approaches trained on a manually labeled data set is also a common way to automatically annotate data. Machine learning techniques, as they are trained on real data, make it possible to take into account the semantic content of motion data. Among the proposed approaches for segmentation by classification, we can find HMM techniques with known labels [94], a combination of Support Vector Machine (SVM) and HMM [95], Linear Discriminant Analysis (LDA) [96] or genetic algorithms to identify characteristic keyframes [93].

Supervised techniques of segmentation make it possible to segment and label the data at the same time at the expense of a higher computational cost than simple kinematic tech-

niques. Supervised approaches are often offline or semi-online techniques – in this second case, the training is done offline and the segmentation can be done online. Unsupervised techniques for segmentation can often be applied online depending on the computational cost of the underlying algorithm but a labeling step is required to annotate the segments.

4.2.3.2 Automatic Labeling of Human Motion

To obtain a complete annotation, the temporal segments detected in the segmentation step must be tagged with a label. This can be seen as a multiclass classification problem where the classes are the labels of each segment. The supervised techniques for segmentation presented on the previous section can also be considered as labeling techniques as both the segmentation and the labeling are done at the same time. Other **machine learning approaches** such as random forests [97] or GMM [98] can be used to recognize human motion. Applying those algorithms on pre-segmented data instead of directly recognizing the motion labels in a continuous data stream simplifies the classification problem and can increase the performance of the recognition. For supervised machine learning algorithms, manual annotation is used as ground truth to train and test algorithms for automatic annotation.

Deep learning approaches are massively used for spatial annotation and deep learning techniques for temporal segmentation and labeling are emerging. Liu *et al.* [99] use deep convolutional neural networks for temporal motion recognition in the context of human-robot collaboration. To the author knowledge, deep learning approaches are not used yet in SL annotation supposedly due to the high amount of data they require.

4.2.4 Automatic Annotation of Sign Language Data

These methods give effective results, but the amount of general human motion data available is often higher than the amount of data available for sign language annotation. Moreover, the number of classes used to describe general human motion is generally very low compared to the numerous classes needed to describe sign languages. The complexity of the signs, characterized by many features (including hand movement, hand configuration and facial expression) and the challenges specific to SL data require the development of dedicated approaches.

4.2.4.1 Annotation at a Gloss Level

The annotation problem at a gloss level can be stated as: (i) segmenting the continuous signing into isolated signs and (ii) recognizing the isolated signs to assign them a label. Both parts are challenging because the form of a sign differs from one utterance to the next due to the inflection and coarticulation effects and the labels to annotate a whole language are numerous.

a) Segmentation of Continuous Signing A large part of the existing work on the automatic segmentation of a continuous stream of sign language data is based on video recordings and seeks to segment at a gloss level.

For example, in their work, Yang *et al.* [100] perform sign segmentation of continuous signing by distinguishing between transitions and signs in American Sign Language using Conditional Random Fields (CRF) while Kim *et al.* [101] take advantage of Hidden Markov Model (HMM) to segment a continuous flow of Korean Sign Language into sign segments. Ye *et al.* [102] simultaneously segment and recognize American Sign Language signs using a sliding window and a recurrent convolutional neural network on a continuous signing sequence.

Other approaches rely on the detection of motion features. Lefebvre-Albaret *et al.* [103] present a semi-automatic segmentation of sign language sequences based on the detection of a set of features including symmetry, repetition of movements, hand velocity and stability of the configuration. A human operator is still required to assist the algorithm by specifying a frame belonging to each sign segment. As for Gonzalez *et al.* [104], they propose to define an automatic two-level segmentation process: the first step uses the minima in the velocity of the hands to provide segmentation points which are confirmed or removed, in a second step, based on the variation of the hand configurations.

Those segmentation approaches do not take into account the multilinear aspect of sign languages and result in a high-level segmentation, highly dependent on the context of the segmented utterance, i.e. the segments implicitly contain the coarticulation and inflection effects of the sequential signs. Even if techniques have been developed to limit the impact of inflections in the form of signs by automatically detecting the parts of signs not inflected by coarticulation [105], most of the resulting annotation is hardly reusable in a different context, for example to synthesize new utterances. A lower-level segmentation, based in particular on phonological components would facilitate sign composition in various contexts in order to produce new signs and utterances.

b) Gloss Recognition After the automatic segmentation and given the high number of possible labels for the segments (the labels correspond to the possible glosses which are equivalent to words in spoken languages), the recognition step is sometimes done manually by experts in a second phase. Work on automatic labeling techniques exists but much of the work simplifies the problem by drastically limiting the number of glosses that are recognizable by the system and/or by removing any inflected or coarticulated signs from the data set that had to be recognized. A review of SL recognition techniques can be found in [106] while the recent survey of Ibrahim *et al.* [107] gives a complete overview of the challenges and solutions for the recognition of continuous sign language data. Ye *et al.* [102] use $3D$ recurrent convolutional neural networks on RGB, motion, and depth SL data to recognize 27 signs from American Sign Language (ASL) from sequences of isolated signs. Zaki *et al.* [108] describe an HMM process to label 50 ASL signs. In an early work using HMM, Grobel *et al.* [109] recognize 207 signs of Sign Language of the Netherlands but they settle for the recognition of signs deprived of any contextual variation, removing the difficulty of managing sign inflections.

To generalize the recognition process on a higher number of glosses, Vogler and Metaxas [110] break signs into "phonemes" (close to the five phonological components of section 3.2.1) and use HMM on the combination of the values of the phonemes to recognize signs. However, due to the difficulty of capturing the finger movements, the hand configuration track was not processed and the authors of the article also chose not to deal with non manual features. Later, following the same intuition, Dilsizian *et al.* [111] propose to add some linguistic knowledge about the hand configurations composing lexical signs in order to increase the recognition performance. In a later work, they use conditional random fields to combine the information about hand configuration, orientation, placement, motion trajectories and linguistic knowledge to perform sign recognition [112].

Kong *et al.* [113] use a two-layer system to label 107 different signs coming from continuous signing. The first layer is a segmentation layer that classifies motion segments into two categories: *transition* or *sign* using a combination of the probabilities given by a conditional random field (CRF) and a support vector machine classifier. The second layer, like in the work of Vogler and Metaxas [110], recognizes the precise label of the *sign* segments using CRF classifiers on (i) the separated phonological elements to find their individual values and, then, on (ii) the fusion of the results (see Figure 4.4). Similarly, Koller *et al.* [114] propose a system that can perform recognition tasks involving multiple signers and a large vocabulary of more than 500 signs using features describing the states

of the hand configuration, hand position, hand movement and face.

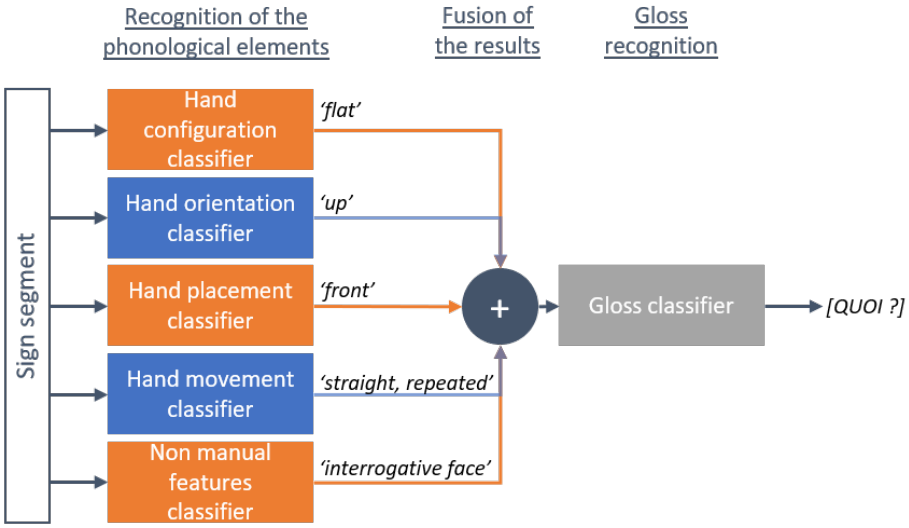

Figure 4.4 – The two-step gloss recognition approach of [113]: (1) Recognition of the phonological elements composing the sign segment and (2) Recognition of the gloss using a fusion of the results of the first step. Example of the LSF sign [*QUOI ?*] (what?).

These works perform annotation at a gloss level using low-level linguistic knowledge. Yet, the values of the phonological components are not considered relevant enough to be retained after the sign recognition. For the analysis of signs and utterances as well as for the synthesis by phonological recombination, it may however be interesting to annotate each track (phonological components and gloss) independently.

4.2.4.2 Annotation at a Phonological Level

Although several studies address the issue of the annotation of sign language data at a gloss level, little attention has been given to the automatic annotation of the different linguistic tracks of sign languages.

Lower level segmentation should facilitate the composition of signs in a variety of contexts. Many phonological structures use the five phonological components that are hand configuration, hand placement, hand orientation, hand movement and non manual features [5]–[7] to define signs which can be used as a basis for annotation: the segments are of a finer level than the gloss segmentation and retain a linguistic value.

a) Segmentation of the Phonological Tracks The segments of a phonological component track correspond to stable states of the component value. Phonological com-

ponents can be segmented using the temporal series derived from a subset of joints (e.g., the joints of the hand for the hand configuration segment). Figure 4.5 shows the segmentation of the hand configuration track using thumb and index motion features with the hypothetical example of the LSF fingerspelling alphabet. The segmentation at a phonological level uses the human motion segmentation techniques of section 4.2.3.1. For the particular application of SL, Héloir *et al.* [115] perform Principal Component Analysis (PCA) to segment manual configurations of the French Sign Language fingerspelling alphabet.

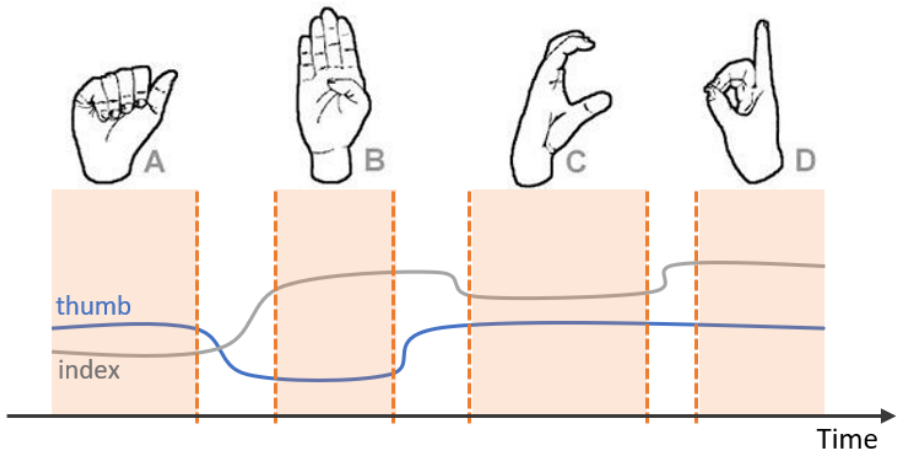

Figure 4.5 – Hypothetical example of a hand configuration segmentation using features on the thumb and index finger for the fingerspelling alphabet.

b) Recognition of the Values taken by the Phonological Tracks The recognition task on SL data was mostly performed on the hand configuration and on the facial expression tracks. Some works, like [113], include movement, orientation and placement labeling but the final purpose of those recognition tasks is gloss identification and not phonological track annotation.

Hand configurations are static poses of the hand. The fingerspelling alphabet gathers a high number of such hand configurations. SVM [116], neural networks [117], k Nearest Neighbors (kNN) [118] and DTW [119] are common techniques for the recognition of the letters of the fingerspelling alphabet. Their accuracy results are presented in Table 4.4. Symeonidis [120] used neural networks for the recognition of hand configurations of ASL in general, not limited to the fingerspelling alphabet.

Reference	Data Source	Classifier	Accuracy
[116]	Myo armband	SVM	From 4% for the letter 'A' to 95% for 'W'. Avg: 41,15% (std: 30,12%)
[117]	Static photos	Neural network	83.03%
[118]	Static photos	K-Nearest Correlated Neighbors	91%
[119]	Static photos and dynamic videos	DTW	86.3%

Table 4.4 – Accuracy results of the recognition of the letters of the fingerspelling alphabet.

Facial expressions are non-manual features of SL. Reverdy *et al.* [121] attempt to segment and annotate the LSF facial expressions from *Motion Capture* data using different classifiers such as random forests, kNN and SVM.

4.3 Summary and Discussions

This chapter reviews the existing work regarding the collection and annotation of sign language data. The first part is about the existing sign language databases which can be constituted of video or *MoCap* sequences. Sign language data sets composed of video recordings are mainly used for linguistic analyses, natural language processing or computer vision tasks. *MoCap* data sets are less widespread but the frame rate, spatial precision and $3D$ data provided by *MoCap* are interesting for gesture and linguistic analyses but also for motion synthesis tasks. The result of a *MoCap* recording is suited to our needs for the analysis and synthesis of precise and natural sign language motions. However, as no available *MoCap* data set met our needs, we collected our own data set and constructed a new corpus described in Chapter 6.

Concerning the content of *MoCap* corpora, two main trends are often confronted. In Duarte's PhD thesis [57], they are named *breath* and *depth*. Breath consists in capturing a large number of different signs to try to cover the whole language, while depth consists in capturing a limited set of signs numerous times with many variations (for example, by varying the location of the sign in space). As sign languages do not contain a finite number of signs due to mechanisms such as proforms or role-shifts, and because capturing data is costly, covering all the possible signs in all the possible contexts (the *breath* solution) is not a viable design solution. Moreover, the motion data, after being captured, is not directly workable. In the case of passive *MoCap*, the signer whose motions are recorded is covered by numerous markers (sometimes more than 100). Those markers are often not

or wrongly identified by the *MoCap* system leading to a task of manual verification and relabeling. Then, potential gaps due to the occlusion of markers have to be filled. The resulting data (either from active or passive *MoCap*) may be filtered to remove unwanted noise. Finally, the joint orientations of the skeleton have to be reconstructed from the sensor data or from marker positions in order to obtain workable skeletal motion data in the form of a motion file. Moreover, the data must be annotated at a more or less fine-grained level depending on the final application of the corpus. This annotation work can be tedious and time-consuming, especially when done manually.

In the second part of the current chapter, we therefore define data annotation and present the challenges specific to sign language data annotation. We describe the different techniques used to automatically annotate general motion data and sign language data. We distinguish the annotation of the gloss track from the annotation of lower-level tracks such as the hand configuration or facial expression tracks. Work on the automatic annotation of the phonological tracks of SL is scarce and focus on only two different tracks. Yet, this lower level annotation is important for a finer analysis and synthesis of sign language content. In Chapter 7, we propose a new technique for the automatic annotation of the hand configuration and hand placement tracks.

Chapter 5

Synthesis of Sign Language Content

Contents

5.1 Movement of Avatars 72
5.2 Isolated Sign Synthesis 73
5.3 Utterance Synthesis 94
5.4 Existing Systems 107
5.5 Summary and Discussions 113

... c'est parce que nous sommes élevés par eux.

Bernard de Chartres, 12th century

This chapter is based on an article submitted to Computers & Graphics in 2020.

Depending on the final application, the approaches to generate sign language animations will differ. The creation of an online bilingual dictionary will require the synthesis of isolated signs in their citation form, i.e. not inflected by a sentence context. On the contrary, an application of translation into sign language will require a mastery of grammatical rules and the use of contextualized and coarticulated signs to generate correct utterances. In both cases, the isolated signs, like the utterances, must be as precise and natural as possible.

Sign and utterance synthesis are two inherently different processes as sign synthesis is the process of generating the skeletal animation of an isolated sign while utterance

synthesis consists in the animation of a whole sign language sentence and therefore requires a more extensive knowledge of sign language linguistics and involves different mechanisms than the generation of isolated signs.

This chapter surveys the different steps and various methods that are currently used to animate SL avatars whether at a sign or at an utterance level. We focus on the animation of the manual features. A complementary survey, dedicated to the animation of facial expressions for sign language avatars, was proposed by Kacorri [122].

It is organized as follows: a definition of motion is given in section 5.1, the process of isolated sign synthesis is described in section 5.2. Utterance synthesis is presented in section 5.3. Section 5.4 lists the existing signing avatars with a special focus on three of them. Finally, section 5.5 compares and discusses the presented techniques.

5.1 Movement of Avatars

In $3D$ traditional animation, an avatar is represented by a complex $3D$ mesh in the shape of a virtual humanoid. It can be animated thanks to a *skeleton* which is a tree structure composed of rigid segments (*bones*) connected by *joints*. A *pose* or *posture* is the state of the skeleton at a given time or frame, described by the position and orientation of each joint. The pose q_i of the skeleton at frame i is therefore defined as:

$$q_i = \{(pos_i^1, orient_i^1), (pos_i^2, orient_i^2), ..., (pos_i^n, orient_i^n)\} \tag{5.1}$$

where n is the total number of joints in the skeleton, pos_i^j is the position of the joint j at frame i and $orient_i^j$ is the orientation of the joint j at frame i.

A *motion* M is a sequence of k poses: $M = \{q_1, q_2, ..., q_k\}$. Its duration is equal to $k * \Delta t$ where Δt is the timestep between two poses.

Forward and Inverse Kinematics (FK and IK resp.) are the main techniques used to synthesize the poses of the avatar. FK consists in animating a skeleton by specifying the joint positions and orientations of the avatar's skeleton for chosen poses. IK, the inverse problem, consists in computing the angles of the joints of an articulated chain knowing the position/orientation of some joints (often its end-effectors). Constraints on the angles must be added to the system to eliminate physiologically impossible solutions. In order to have a constant Δt, in-between poses for a given frequency can be interpolated.

Standard file formats, like *BVH* or *FBX*, are designed to record motions. Generally, the skeleton's hierarchy is specified and the corresponding sequence of poses is stored in

the form of a sequence of numerical values. The *FBX* format can, in addition, store the $3D$ mesh of the avatar as well as other features.

This chapter focuses on the skeleton motion synthesis in the case of sign language generation. The rigging and rendering steps, which consist in computing the deformation of the $3D$ model and in displaying the resulting animation are beyond the scope of this thesis as they are not specific to sign language synthesis.

5.2 Isolated Sign Synthesis

Isolated sign synthesis consists in generating the motion corresponding to signs deprived of contextual information. Isolated sign synthesis mainly aims at building uninflected signs for bilingual dictionaries, educational applications or very simple utterance generators with the concatenation of signs.

In the field of sign synthesis, signs in their citation form are often preferred to inflected signs for several reasons: they exist in a limited number and are listed in dictionaries, their description in a notation system often already exists (e.g., *HamNoSys* [123], see Section 5.2.1) and they have an almost direct equivalent with words in oral languages. These signs can be defined by a set of fixed values taken by phonological components of SL following a parametric approach.

Three techniques are used to synthesize isolated signs: the keyframe, procedural and data-driven techniques. The first consist in a specification of the key poses of the skeleton, the second in an automatic computation of the motion while the last uses *Motion Capture* data to produce realistic gestures. To specify the features of the sign to be generated, the three techniques need a representation of the synthesis objective.

We first present, in Section 5.2.1, visual representations and parametric notations of signs that can be used to manually specify keyframes. Then, in Section 5.2.2, we survey the different computer-friendly representations of signs that are directly used in procedural animation and, potentially, in data-driven processes. Finally, Section 5.2.3 presents the existing synthesis techniques.

5.2.1 Linguistic Representation of Signs

In order to synthesize a sign, a representation of the synthesis objective is needed. For isolated signs which are mainly synthesized following a parametric approach, the represen-

tation should highlight the structure and, possibly, the values taken by the phonological elements during the sign production. The spatiotemporal aspect of sign languages makes exhaustive representation of signs a complex problem involving researchers both in the linguistic and in the computer animation fields. We present here the representations mainly used by the linguistic communities to study signs.

5.2.1.1 Visual Representations

Visual representations are straightforward ways to represent signs. It consists in representing the signs on a $2D$ canvas by being as faithful as possible to the actual sign. Drawings and video recordings are two common visual representations.

Signs are motions specified both in the $3D$ space and in time: in a **drawing**, the use of arrows and of different types of contour (e.g., dotted line or fine line) allow for a partial representation of those dimensions on a $2D$ paper. However, this schematic representation depends on the interpretation of the user and on the skill of the artist. Drawings are an ambiguous representation that often needs to be clarified with annotations. Figure 5.1 is extracted from a French Sign Language textbook. It shows a drawn representation of the sign [HELLO] complemented with annotations about the hand motion, the facial expression and the hand configuration. As a consequence of this ambiguity and of the impractical aspects of $2D$ drawing for computerization, this type of representation cannot be directly used as a computer formalization for animating an avatar.

Figure 5.1 – The sign for [HELLO] in LSF. To remove any ambiguity of the $2D$ drawing, some annotations are added to describe the motion, the facial expression and the hand configuration.

Videos recordings are another, more precise and exact representation of signs that is very popular in the linguistic community to store and study SL signs and utterances [1].

1. Elix is an example of video-based dictionary for French Sign Language `https://dico.elix-lsf.`

Videos do not allow for signer anonymity but are very efficient to record the dynamics of signs. Nevertheless, a video recording alone eliminates the depth information and imposes a point of view on his viewer. Like drawings, it lacks flexibility and its format is not suited for automatic computer synthesis.

Those visual representations may be attractive due to the intuitive understanding of the sign structure and dynamics that they provide. Their visual format is suited to the manual definition of keyframes by graphic designers. However, their ambiguities and format makes them hardly suitable to be used as a sign representation in an automatic animation engine.

5.2.1.2 Gloss Representation

Another straightforward way to describe sign language is to use a gloss representation or "glossing". Glossing consists in associating one or more words of an oral language to a sign. It is used by linguists to annotate SL videos. No gloss standard exists (which is a recurrent impediment to obtain consistent annotated corpora [124]) but, as a convention for this thesis, we will designate glosses using brackets and uppercase letters (i.e. [GLOSS]). A gloss is not a translation but the spoken language description of the signed language utterance: for example, a sentence glossed as "[YOU][LIVE][WHERE?]" will be translated as "Where do you live?". Furthermore, a sign can have no one-word equivalent in a spoken language. In this case, the corresponding gloss can contain more than one word (e.g., the difference between the utilization of [TO GIVE] can be indicated in the gloss description : "[TO-GIVE-A-BOOK]" vs "[TO-GIVE-A-GLASS]").

The way to execute a sign is not indicated in the gloss description. Therefore, few animation systems can choose to rely exclusively on a glossing description of the desired SL production since it implies the presence of a motion database annotated on the same gloss-level as the specification like in [69] for *Motion Capture* data or in [125] for hand-crafted animation. As a consequence, isolated sign animation systems need another, lower-level sign representation to achieve the actual motion synthesis.

5.2.1.3 Parametric Notation and Writing Systems

The most obvious way to transcribe a language is writing. For oral languages, alphabets or syllabaries are common ways to describe the sounds that can be produced in a spoken

`fr/`

utterance. Each character is assigned a sound and the concatenation of characters forms new sounds that result in words and sentences when spoken out loud. The transcription of an entire language is possible using a finite number of letters (using the native writing system or other transcription alphabet such as the Pīnyīn notation for Chinese language).

The conception of a similar transcription system for signed languages is the focus of many studies. The written representation of the linguistic production is called a *notation system*. As mentioned in Section 3.2.1, signs can be decomposed into linguistic components that can be seen as phonemes. Parametric notations propose to decompose signs into a combination of those components and to assign a value for each one from a finite set of possible values. However, due to the spatiotemporal aspect of the language, the assignment of a finite number of characters to the description of the whole set of possibilities of the language is not an easy task. The parametric notations that are presented in this section aim at discretizing the continuous concepts that are space and time in order to find the optimal transcription of sign language. Some questions are often raised, such as the partitioning of the signing space, the number of possible hand configurations, or the way to represent kinematic behaviors (acceleration, deceleration, etc.).

In 1825, Auguste Bebian, looking for a notation system for sign language, is the first to define the signs as a combination of elementary components [126]. However, due to "The Milan Conference" of 1880, which promoted the oral education of deaf people in Europe at the expense of sign languages, research concerning sign language writing was stopped until the important work of Stokoe in 1960. Stokoe [5] defined three linguistic components: *hand configuration*, *hand placement* and *hand motion* (see Section 3.2.1). His work on a sign language transcription system resulted in the **Stokoe notation** [127] which describes the ASL signs using a combination of those three components: the sign location is called *tabula* or *TAB*, the hand configuration is *designator* or *DEZ* and the hand motion is *signation* or *SIG*. Each component is specified using a limited set of symbols. *Hand orientation* and *non-manual features* (NMF) (mainly facial expressions) were later added to complete the definition [6] but only hand orientation was incorporated to the notation. The notation merely describes the signs without taking into account the intent of the signer: even if the two hands perform a symmetric gesture, the hand configuration of each hand will be described (in the *Stokoe* box of Figure 5.2, we can see the symbols B^ repeated twice, one for each hand).

Later, the **Hamburg Notation System** or **HamNoSys** [123] was developed to

palliate those limitations. It includes the four components of the *Stokoe notation* and some facial expressions. The variety of possible values for each of the components makes *HamNoSys* a much more complete notation system. It was designed to be language-independent and extensible to new linguistic mechanisms. It can better handle some SL mechanisms like symmetry (the symbol ¨ at the beginning of the transcription in the *HamNoSys* box of Figure 5.2) and can be transcribed in a linear way using computer Unicode symbols. However, it is still limited in terms of non-manual features.

SignWriting [128] is based on a dancing gesture transcription and constitutes a more graphical and intuitive notation of SL. It integrates some facial expressions, body movements and even iconic signs (see Figure 5.2). Hand configurations and orientations can be defined using a limited set of symbols but the placement of the hands with respect to the body is indicated only in a relative manner thus leaving some ambiguities. It differs from the two previous notations in the sense that it was designed to be used by deaf people as a writing system and not by linguists or researchers as a notation system. It is SL independent and is taught in deaf classes around the world [129]. Other writing systems exist for different sign languages (like "SEL" for Brazilian Sign Language [130]) but none as popular as *SignWriting*.

Stokoe notation, *HamNoSys* and *SignWriting* focus on dealing with the representation of the static components of SL but fail to represent the dynamics of signs and synchronization of the different components.

The temporal aspects of SL were thoroughly studied in the work of Johnson and Liddell. In 1989, they introduced the **Movement/Hold model** [131] where signs were defined as an alternation of static poses (*Hold*) and dynamic transitions (*Movement*) between two consecutive poses. Then, between 2010 and 2012, they defined the **Sign Language Phonetic Annotation (SLPA)** [132] which relies on the Posture-Detention-Transition-Shift (PDTS) classification. This classification uses two timing features to distinguish between the four classes, called *segments* or *timing units*. The first timing feature is the *static/dynamic* nature of the segment. In static segments, one or more SL component(s) (hand configuration HC, orientation FA, placement PL, non-manual features NM) is stable during a finite amount of time while *dynamic* segments are transitions from one static segment to the following. The second timing feature is the *transient/deliberate* quality of the motion during the segment. This feature mainly impacts the duration of the segment. A deliberate segment will have a significantly higher duration than a transient segment. More precisely, the four types of timing units are the *Posture* (static and transient), the

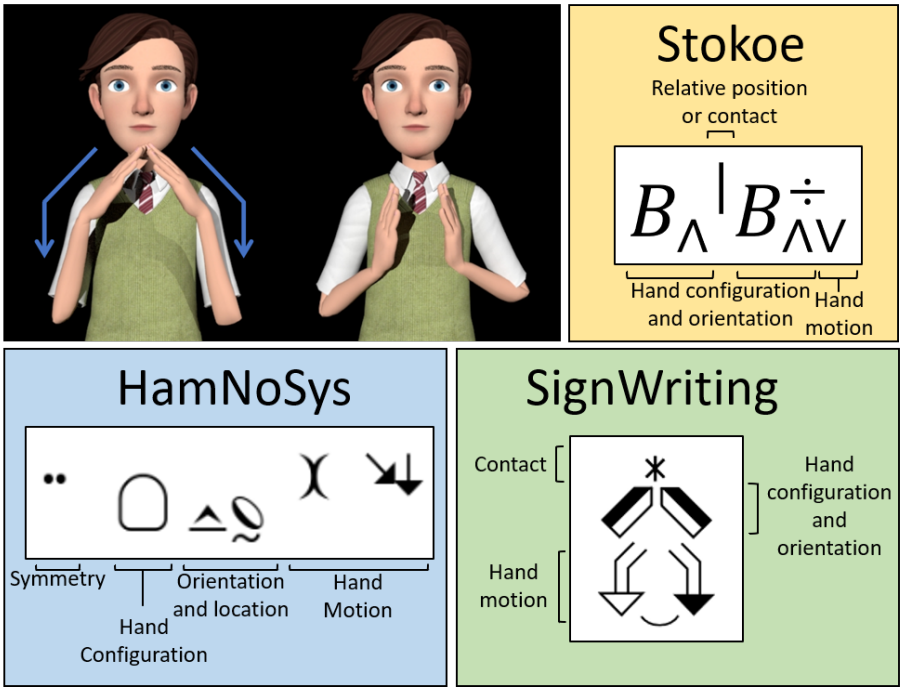

Figure 5.2 – The sign for [HOUSE] in American Sign Language using the *Stokoe*, *HamNoSys* and *SignWriting* notation systems. The sign is shown on the avatar on the top-left corner.

Detention (static and deliberate), the *Transform* (dynamic and transient) and the *Shift* (dynamic and deliberate) segments. The *SLPA* has the particularity of transcribing a sign using a table: the timing units are represented in the columns whereas the articulatory components are described on the rows. An "∞" symbol designates a change in the articulatory features. Figure 5.3 shows the *SLPA* transcription of the word [CHICAGO] in ASL.

5.2.1.4 Phonetic Codings of Hand Configurations

To describe the hand configuration, the phonetic and the phonological systems must be distinguished. In phonological systems such as *Stokoe notation* or *SignWriting*, the hand is seen as a whole and a name or a symbol is assigned to a hand configuration. In phonetic systems, the disposition of each finger or even each finger joint is specified to describe a particular hand configuration. Phonetic codings have been defined by the linguistic communities to describe hand configurations in an exhaustive and generic way. For the same reasons, such low-level descriptions are interesting to automatically synthesize hand configurations. Two examples of such codings are presented hereinafter.

Timing units / Articulatory features	Posture	Transition	Posture	Transition	Detention
Manual features	HC1 PL1 FA1	∞	PL2	∞	PL3 FA2
Non-manual features	NM1	∞	NM2	∞	NM3

Figure 5.3 – The sign for [CHICAGO] in American Sign Language using the SLPA notation system (table based on [132]). The sign [CHICAGO] draws a "7" in the signing space with a 'C' hand configuration. The three placements PL_i correspond to the three inflection points of the "7".

In addition to the temporal aspects, the **SLPA** introduces a phonetic coding for the hand configurations [133], [134]. Indeed, in Figure 5.3, *HC1* represents a hand configuration that stays stable during the execution of the sign but it gives no indication on the nature of the hand configuration. The *SLPA* coding of the hand configuration was defined in order to precisely describe those hand configurations. The position of each finger is described by indicating if each joint (3 joints of each finger + 2 joints of the thumb) is flexed (f or F with respect to the intensity of the flexion) or extended (e or E). The position of the fingers with respect to each other are noted with a particular symbol (e.g., "=" if the fingers touch each other, "<" if they are separated). The description of the thumb position with respect to the other fingers is placed at the beginning of the specification (U for "unopposed", O for "opposed" and L for "lateral"). Additional notations describe the type of contact between the fingers. The *SLPA* phonetic coding is thus exhaustive but its exhaustiveness makes it a complex system, hard to use in practice. A more user-friendly version of the *SLPA* could be defined by reducing the number of degrees of freedom and introducing anatomical knowledge to eliminate anatomically impossible hand configurations [135].

Another phonetic coding of the hand configurations, the **Prosodic Model Handshape Coding (PMHC)**, was proposed by Eccarius *et al.* [136]. Each configuration is coded based on (i) the detection of sets of *selected fingers* – groups of fingers that are the most relevant for the configuration and that share a state (or *joint configuration*) – more than one group can be identified (separated by ";"), (ii) the determination of the state of those sets of fingers (e.g., curved "@", bent "[", crossed "x" or extended by default), (iii)

the determination of the state of the non-relevant fingers (extended "/" or flexed "#") and, (iv) the specification of the thumb position. It uses the standard ASCII characters to represent the hand features and different sign languages were studied to design the system making it a generic and extendable system. Figure 5.4 shows the coding of two hand configurations with the *SLPA* and the *PMHC*. *PMHC* is more compact than *SLPA* but it lacks its precision.

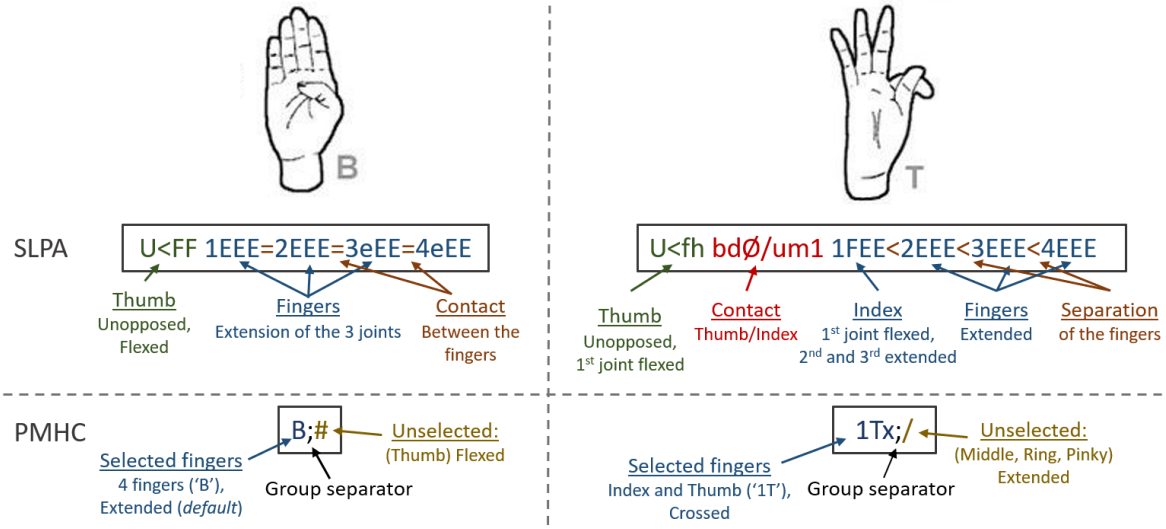

Figure 5.4 – The coding of the 'B' and 'T' hand configurations with the *SLPA* and the *PMHC* phonetic codings.

While the visual representations can be taken advantage of by graphic designers to manually define the key poses of a skeleton, the notation and writing systems are rarely used as such by the computer animation community. They are useful systems for SL linguistic analyses but lack the precision needed to synthesize motions. Phonetic codings of hand configurations, also defined in the linguistic fields, provide detailed descriptions of the SL hand configurations and could potentially be integrated in a specification language. In order to animate SL avatars, SL notation systems have therefore been used as a basis to design SL specification languages and scripts that can directly be exploited to create signs automatically.

5.2.2 Scripting Languages for Sign Representation

Scripting languages have been designed to specify the signs in a way directly understandable by a computer. Those languages have been developed by the computer anima-

tion community but are often based on linguistic sign representations.

5.2.2.1 Descriptive Languages based on Existing Notation Systems

Descriptive markup languages are used to describe and structure a document or a data set. Among them, the eXtensible Markup Language (XML) describes the data in a way that is understandable both by the humans and the computers. It is commonly used to describe natural languages and is suited to the specification of signs as it can integrate the information of existing notation systems in a computer-readable language.

The **SignWriting Markup Language (SWML)** [137] is an XML version of *SignWriting* and has been designed to allow storage and processing of sign language files. The structure corresponding to a sign, called a *signbox*, contains the exact same information as the *SignWriting* transcription of the sign. Consequently, it does not overcome the limitations of *SignWriting*, namely the ambiguities of the hand placement and the time management.

Similarly, **SiGML** was initially an XML version of the *HamNoSys* transcription system [138]. This language was developed within the European projects *ViSiCAST* and *eSIGN* that promote Deaf access to information [139]. Contrary to *SWML*, it was designed with the prospect of animating virtual signers, this is why some aspects of SL, like timing or very precise orientations, can be specified in *SiGML* and not with *HamNoSys*. Moreover, the PDTS classification of Johnson & Liddell was later added to *SiGML* in an extended version of the language [140] additionally providing an explicit timing control, synchronization between elementary motions and a direction specification in various contexts. It is one of the most advanced existing sign language specifications for SL synthesis.

A parametric description of signs based on an XML version of the *Movement/Hold model* of Johnson & Liddell studies was also designed by Amaral *et al.* [141] in order to animate an avatar using Brazilian Sign Language to translate textbooks for educational purposes [142].

However, XML version and extension of existing notation systems do not have the monopoly of sign representations. Dedicated programming languages can also be used to depict signs.

5.2.2.2 Programming Languages for Sign Synthesis

The definition of a programming language for synthesizing SL implies the specification of a dedicated lexicon and syntax to be used in subsequent instructions. Such languages are often defined for a specific animation engine and are less generic than descriptive languages but offer more freedom and flexibility to the programmer.

In an early work, Lebourque *et al.* [143] defined **QualGest**, a high-level specification language dedicated to LSF that takes into account the four manual parameters (hand configuration, placement, motion and orientation), called *gestems*.

To specify the hand placement, they use a discretization of the signed space, including the definition of: (i) a set of directions (defined from the three main planes – sagittal, frontal and horizontal–, plus two intermediate planes), (ii) of amplitudes (proximal, medial, distal and extended), and, (iii) of body positions. Movements are defined using a finite set of predefined primitives (pointing, straight-line, curve, ellipse, wave, or zigzag), parameterized by a set of starting, ending and, if needed, intermediary locations. Hand configurations are defined, using 5 basic hand configurations (angle, hook, spread, fist, stretched), completed by *modifiers*. Hand orientation can be specified in a relative or absolute manner from two hand directions (palm and metacarpus). All the different values taken by the parameters can be specified by meaningful terms or numerical values. Even if non-manual channels are not taken into account in this description, other interesting information about the symmetry of the two arms, the synchronization between the dominated/non-dominated arm or the number of repetitions of elementary gestures can be added, thus indicating the intent of the signer. Moreover, coarticulation mechanisms can be added to the synthesized motion.

Similarly, Losson defines a sign description where signs are divided into atomic gestures called *shifts* [144]. Again, the four Stokoe's parameters are used to determine the characteristics of the *shifts*: initial and final configurations of the hand, orientation and placement, and nature of the movement. Hand motions are specified using displacement primitives (straight line, arc or circle) and the location of the targeted destination of the hand (plus the equation of a plane in which the displacement takes place for arc and circle trajectories). The primitives can be augmented with secondary movements, contact zones or modifiers. To describe the hand configurations, the behavior of the thumb is considered separately from the other fingers (similarly to the hand configuration coding of *SLPA*). Repetitions in the movement, synchronicity properties of the hands, symmetry

or anti-symmetry characteristics, or relative placement of the hands can be specified. In addition, Losson's representation uses a parametric computer language that allows the specification of sign inflection mechanisms (size and shape specifiers, spatial referencing). Figure 5.5 shows an example of Losson's specification for the sign [CHIMNEY] in LSF using two *shifts*.

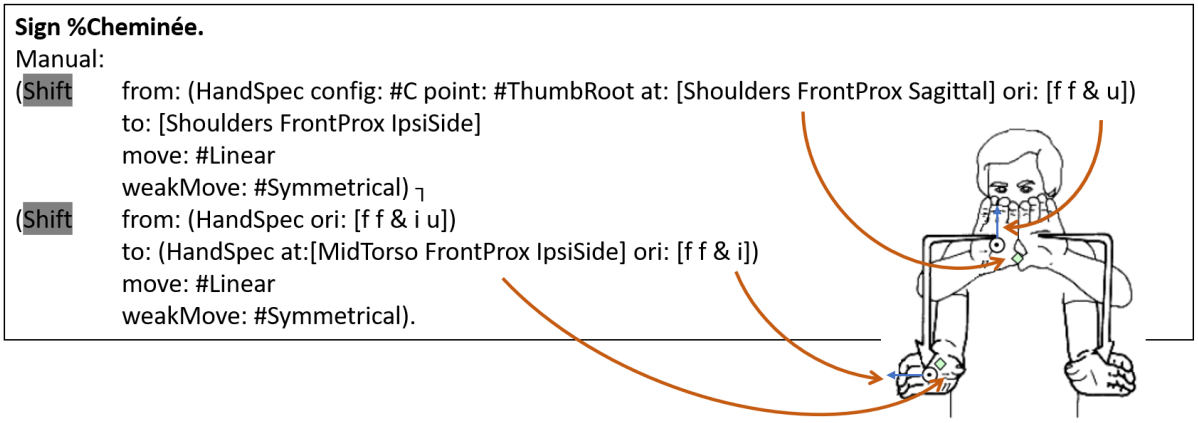

Figure 5.5 – Example of Losson's description for the sign [CHIMNEY] in LSF (avatar and code extracted from [144]).

In *QualGest* and Losson's approaches, the time does not explicitly appear in the SL specifications, whereas the **EMBRScript** is based on the explicit specification of key poses in absolute time, allowing a fine temporal control of the animation [145]. It was originally designed to describe the motions of Embodied Conversational Agents (ECA) and was extended to be applied on SL avatars by adding new hand configurations, facial expressions and gaze directions [146]. It specifies the low-level animation data of the *k-pose-sequence*, a sequence of key postures corresponding to a sign.

While the previous specification used a phonological definition of signs, **Zebedee** is a sign specification language based on a geometrical definition of signs [147]. Geometric constraints on points, vectors or surfaces replace the set of parameters usually used (hand configuration, orientation, placement). The *Zebedee* model separates signs into two types of temporal units, following the Movement/Hold model of Liddell and Johnson [131]: the *key postures* when the parameters of the motion reach a stable state and *transitions* in-between two consecutive key postures. One of the advantages of this description model is that it can also represent signs with inflections by modifying the geometrical constraints

describing the sign (e.g., the difference between [BIG BALLOON] and [SMALL BALLOON] in *Zebedee* will be done by changing the radius parameter). *Zebedee* captures both the temporal and the spatial constraint of sign languages. Furthermore, the geometric nature of the language can highlight the structure of signs. Figure 5.6 shows the representation of the sign for [BALLOON] in LSF with the LIMSI avatar performing the sign.

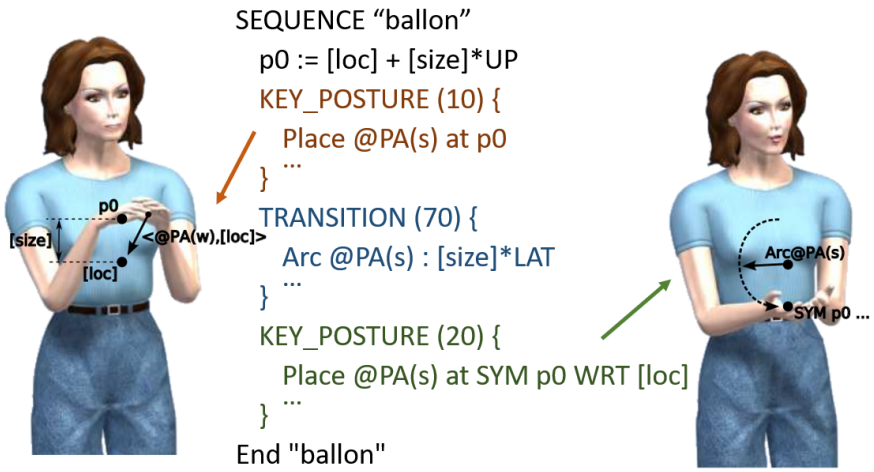

Figure 5.6 – Example of the representation of the sign for [BALLOON] in LSF with the *Zebedee* language. The left (resp. right) image is the initial (resp. final) position of the LIMSI avatar (avatar and code extracted from [147]).

All the different sign representations are compared in Table 5.1.

5.2.3 Synthesis Techniques

The sign representations described in the previous sections constitute the first step of the synthesis of isolated signs. The choice of a synthesis technique is the next step to achieve a meaningful animation.

5.2.3.1 Keyframe Techniques

Keyframe synthesis is the most straightforward technique to generate the animation of an isolated sign. An animation is a sequence of avatar poses displayed at a given frequency. Some poses may be more relevant than others – e.g., the state of the avatar at the beginning and end of a sign, or the pose describing an inflection point in the

Category	Name	Fidelity	Temporal aspects, synchronization	Non manual features	Flexibility	Readable by a computer
Visual Representation	Drawings	✓	✗	✓	✗	✗
	Video recordings	✓✓	✓✓	✓✓	✗	✗
Parametric Notation	Stokoe [127]	(✓)	✗	✗	(✓)	✗
	HamNoSys [123]	✓	(✓)	(✓)	(✓)	(✓)
	SLPA [132], [134]	✓	✓	✗	(✓)	(✓)
	SignWriting [128]	✓	(✓)	(✓)	(✓)	✗
Scripting Language	SiGML [140]	✓	✓	(✓)	(✓)	✓✓
	QualGest [148]	✓	✓	✗	✓	✓✓
	Losson [144]	✓	✓	(✓)	✓	✓✓
	Zebedee [147]	✓	✓	✗	✓✓	✓✓
	EMBRScript [146]	✓	✓	✓	(✓)	✓✓

Fidelity: absence of ambiguity, fidelity to the original movement, precise description of the sign, preservation of the intent of the signer.
Temporal aspects, synchronization: the dynamics of the movement is specified, the synchronization between the different channels is managed.
Non manual features: the status of non-manual components (facial expressions, gaze, etc.) is specified/visible.
Flexibility: the ease with which the representation of a sign is modified to take into account the context of the sentence. A purely visual representation will make the transformation fastidious while some linguistic representations are highly flexible.
Readable by a computer: it can be reused as it is at the input of an automatic synthesis engine.

✓✓: Good management of the functionality
✓: Management of the functionality
(✓): Partial management of the functionality
✗: Functionality not managed

Table 5.1 – Comparison of the sign representations.

hand configuration. Those key poses, associated with a time tag, are called *keyframes* and a special attention should be paid to their description. Keyframe animation consists in describing the pose of an avatar for each keyframe, the transitions between those keyframes is then automatically computed by interpolation.

a) Hand-Crafted Animation In hand-crafted animations, the specification of the keyframes is done manually.

3D traditional animation consists in setting the avatar in a specific pose at different frames of a timeline using a specialized animation software such as Autodesk Maya [149] or Blender [150] and the knowledge of human anatomy and motion. Different joint angles values are tested and the best values are selected for a particular pose of a particular sign. The process is fastidious and cannot be generalized to other signs. The early work

of Shantz [151] in 1982 which is considered to be the first work on sign language avatar animation [152], used this technique.

Rotoscoping is a particular instance of $3D$ traditional animation. It is used to produce realistic movements from video footage. It consists in posing over a projection of a video recording of the animated scene using a $3D$ animation software.

Examples of avatars relying on hand-crafted keyframes:

— *Paula* of DePaul University is an American Sign Language avatar that partially relies on traditional animation and on the PDTS classification [153], [154].

— The Italian Sign Language avatar of the University of Torino is animated using hand-crafted keyframes [155], [156].

— *Elsi*, a French Sign Language avatar designed by the LIMSI laboratory to be exhibited in French train stations, is based on rotoscopy [125], [157].

— The work of Irving *et al.* [158] proposes to synthesize signs by defining its keyframes in a parametric way according to the Stokoe parameters. Each hand location, configuration, movement and orientation can be described very precisely using sliders in a graphical interface.

— The Turkish Sign Language avatar of Yorganci *et al.* [159] uses an original approach: the torso and arm motions, the facial expressions and the hand configurations of the avatar are manually modeled separately and are combined in order to generate signs, leading to a parametric and somewhat generic creation of signs.

Hand-crafted techniques can give precise results depending on the skill and choices of the artist. Indeed, he or she is the one in charge of determining the keyframes to be reproduced, the missing frames being deduced using interpolation techniques. However, this is a laborious task and, for the particular application of sign language synthesis, the designers must be expert both in $3D$-modeling and in sign language, a combination that can be hard to find.

b) Automatic Keyframing The tiresome $3D$-modelling work can be avoided by using the sign specification of key postures of Section 5.2.2 to automatically compute the key poses of the avatar. Indeed, a lot of those specifications define signs as a sequence of static and dynamic segments intuitively leading to a key postures/interpolation definition of signs. They can be the basis for the definition of spatiotemporal targets for the avatar joints leading to keyframe animation.

A typical approach is to define the hand configuration(s) of a key pose using forward kinematics (FK) by referring to a look-up table where the joint angles of the hand corresponding to each hand configuration are listed. The position of the wrist or the palm of the hand is determined using the placement descriptors of the chosen sign representation and the angles of the arms are computed using the result of an inverse kinematics (IK) algorithm (see Figure 5.7 and Figure 5.8). The motion between two keyframes is synthesized using different interpolation methods applied to the angles of the joints.

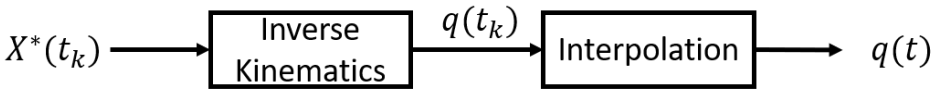

Figure 5.7 – The hand motion synthesis process for automatic keyframing techniques. An inverse kinematics system is used to compute $q(t_k)$, the state of the skeleton at discrete time t_k corresponding here to the timestamp of the keyframes, from $X^*(t_k)$, the desired position of the hands at t_k. To obtain $q(t)$, the state of the skeleton at time t, the intermediary poses of the skeleton between each t_k are interpolated.

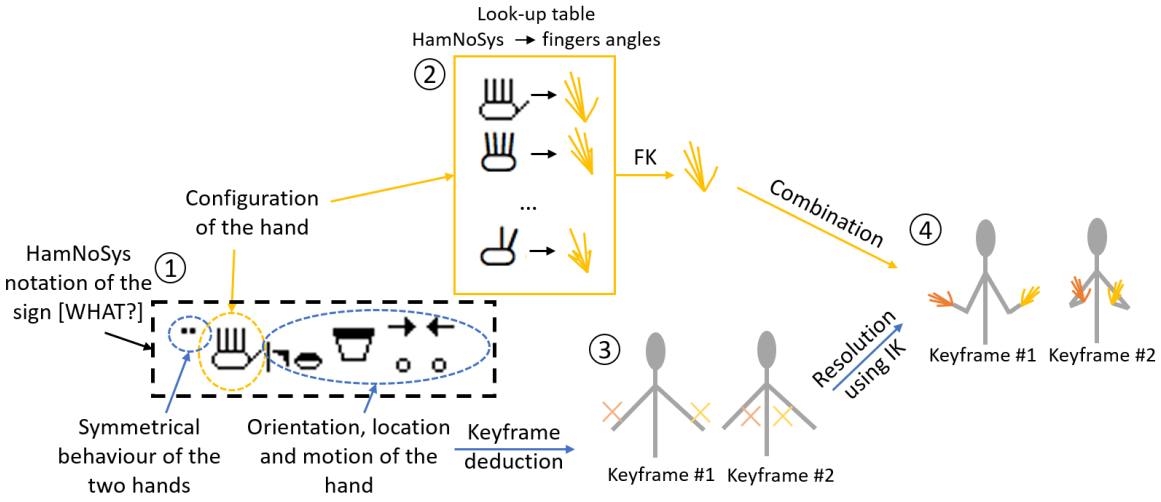

Figure 5.8 – Automatic keyframing typical process for the sign [WHAT?] in ASL. ① : *HamNoSys* notation of [WHAT?]. ② : Resolution of the hand configuration with FK. ③ : Computation of the keyframes and placement of the hands over time based on the *HamNoSys* notation. ④ : Combination of the hand configurations and of the hand placement using IK.

Examples of avatars relying on automatic keyframing:

— Grieve [160] used an adaptation of the Stokoe notation for the ASCII characters (the *ASCII-Stokoe* notation) to place its targets.

— The work of Papadogiorgaki *et al.* [161] is based on the SignWriting Markup Language (*SWML*).

— The avatars of Krnoul *et al.* [162] and Fotinea *et al.* [163] are based on the *HamNoSys* notation.

— Symbolic representation is also used by the VCom3D company to animate the commercial avatars of their Sign 4 Me application [164].

— The avatar of Delorme [165] relies on a segmentation of the motion in terms of key postures and transitions, as defined in the *Zebedee* specification system.

— In the EMBR avatar of Kipp *et al.* [145], [146], the skeleton key poses are described at a gloss level using the *EMBRscript*. The transition between two poses is smoothened by enhancing the interpolation with temporal modifiers.

— Losson & Vannobel [152], [166] used an analytical approach to compute the hand configuration, placement and movement of their avatar using Losson's specification of signs.

Keyframe animation creates a precisely controlled motion which is very important for sign language animation. Indeed, signs have to be generated carefully to keep their meaning. Moreover, keyframing techniques, either hand-crafted or automatic, provide consistent animations but they are often characterized by robotic motions as interpolation between keyframes does not always convey the kinematic properties of natural motion.

5.2.3.2 Procedural Techniques

Instead of relying on keyframes techniques where only key poses are computed, procedural techniques automatically synthesize every pose of an avatar resulting in the generation of a continuous motion. Procedural techniques are automatic techniques using definitions and rules based on mathematical and physical tools. The procedural models involved are also driven by a sign representation and solve either inverse kinematics or dynamics problems. Moreover, procedural tricks can be used to add realism to the generated animation regardless of the motion synthesis technique used.

a) Continuous Motion Synthesis To counter the limitations of keyframing/interpolation techniques which do not guarantee fluid and human-like motions, procedural approaches

use kinematics or dynamics control loops in order to generate a continuous motion. Those approaches are still based on sign representations which define discrete spatiotemporal targets at the task level. A continuous motion is created in order to reach those targets.

Automatic keyframe animation uses IK algorithms as a way to obtain joints angle for a given keyframe without taking into account the possible intermediary solution of the IK problem; only the output of the IK model is exploited. However, in **procedural techniques based on IK**, the motion of the skeleton is generated by the intermediary positions given by iterative IK method through an IK-based control loop with optional biomechanical or neuromimetic constraints (see Figure 5.9).

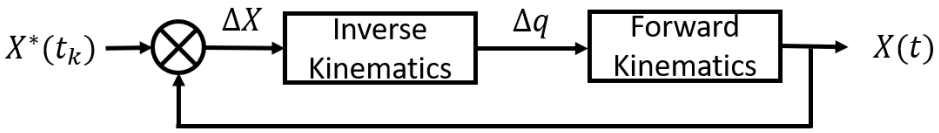

Figure 5.9 – The hand motion synthesis process for IK-based procedural techniques. At each time step, the difference ΔX between the targeted $X^*(t_k)$ and the actual position and orientation of the hand $X(t)$ is computed to serve as input to an inverse kinematics system. A small displacement of all the skeleton joints Δq is computed, thus estimating the state $q(t)$ of the system at time t, and applied to the skeleton, thanks to a forward kinematics system.

While kinematic animation focuses on the motion and trajectories themselves, dynamic animation concentrates on forces. Instead of guiding the gesture like in the kinematic case, dynamic animation consists in modelling the forces that lead to the motion. Therefore, it is a powerful tool to obtain realistic reaction in the case of an interactive application. Sign language synthesis, however, needs more precision than interactivity with an environment. Therefore, most techniques developed for signing avatars have focused on kinematic animation which allows to reach both the precision of the animations essential for the hands and the fluidity of the movements. To our knowledge, no work has been dedicated to exclusive dynamic animation for sign language synthesis but some work takes advantage of the realism provided by **dynamic controllers** combined with IK (see Figure 5.10).

Examples of avatars relying on continuous motion synthesis:

— *GessyCA* [148], [167], the synthesis engine based on *QualGest*, takes as input a discrete sequence of weighted spatiotemporal targets to produce the motion. While the hand configuration is classically described by forward kinematics, the motion itself is generated with a sensory-motor **IK-based** control loop. The targets are not

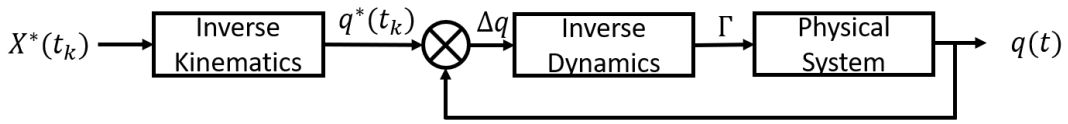

Figure 5.10 – The hand motion synthesis process for dynamics-based procedural techniques. First, $q^*(t_k)$, the desired state of the skeleton at discrete times t_k is computed from $X^*(t_k)$, the desired position and orientation of the hands, using an inverse kinematics system. Then, at each time step, the difference Δq between the desired state of the skeleton $q^*(t_k)$ and the actual state of the skeleton $q(t)$ is computed to serve as input to an inverse dynamics system. A force Γ is computed and applied to the skeleton, thanks to a physical system, to obtain $q(t)$.

necessarily reached at all time allowing for a slightly less precise but more human-like motion as coarticulation effects are implicitly taken into consideration in the optimization process.

— In the *VISICAST* project [168], a **dynamic** model coupled with IK targets is defined: each joint is represented as a control system for which the controlled variable is the angle of the joint. Joints are virtual masses whose acceleration is proportional to the force computed to reduce the error between the current angle and the desired angle of the joint. The trajectory of the arm motion is computed using IK and the angles deduced by IK are then fed to the controllers as reference angles.

b) Adding Realism Through Procedural Means Automatic keyframing and continuous motion synthesis techniques are a convenient way to produce sign language animations. However, even using kinematic or dynamic controllers, they are often characterized by unrealistic motions as sign specification languages seldom take into account the non-semantically relevant part of sign language which conveys the natural-looking aspect of motion.

In order to add some realism to the animation, some mechanisms can be implemented. One of the preferred tricks is to add some small human imperfections to the avatar animation to avoid too stiff postures or too perfect performances. Adding noise, ambient motions, autonomous behaviors ensuring breathing motions, or modifying the timing of the motion on the different channels are recurrent methods used to improve the realism of avatar motions. Signal processing techniques based on *Motion Capture* data analysis can also be used to improve the realism of avatar motions [154].

Examples of avatars using procedural means to add realism:

— In the ViSiCAST project [168], noise, damping and ambient motions (small random eye / head / torso motions) have been added to reduce the robotic and unnaturally stiff movement inherent to their dynamics-based model.

— A small de-synchronization of the hands during the performance of symmetrical signs has been added to the Irish Sign Language avatar of Smith *et al.* [169].

— In the EMBR project of Héloir & Kipp [145], an autonomous behavior ensuring breathing and some natural movements like blinking as well as blushing effects has been implemented.

— To liven the avatar *Paula*, noise has been added to its movements [170].

However, even if various techniques are implemented to overcome the limitations of synthesis techniques based on synthetic models, they cannot compete with data-driven synthesis techniques in terms of realism.

5.2.3.3 Data-Driven Sign Synthesis

Data-driven synthesis techniques use captured motions to animate an avatar. They are less exploited than hand-crafted or procedural synthesis techniques even though they provide a high level of realism that can hardly be achieved by procedural means.

Motion Capture (*MoCap*) is used to record the position of markers placed on a human being performing a motion. The position and orientation of the human's joints are then deduced from the *MoCap* data during the post-processing step. The previous chapter, Chapter 4, reviews the existing database and annotation methods. In this section, we are only interested in the synthesis methods based on a *MoCap* database.

Synthesizing signs using a limited set of pre-recorded signing sequences is the major challenge of data-driven techniques. Three main approaches exist to obtain isolated signs using *Motion Capture* data:

1. **Playback**: the captured data can be played back without modification,
2. **Editing**: new motion can be created by editing existing motion data,
3. **Machine Learning**: new data can be synthesized using knowledge from MoCap data via machine learning approaches.

Regardless of the chosen synthesis technique, data-driven synthesis involves motion retrieval. It consists in choosing and extracting the best motion(s) for a particular application among a set of motions. It can be done by (i) directly querying the motion

features (e.g., see Kapadia et. al. [171] for motion retrieval of non linguistic motion using Laban movement analysis), or (ii) using a textual search in the database containing the annotation of the motion [64], [172]. In this second case, the retrieval process relies on a management system database, separating the two levels of representation (semantic symbols and raw motion data) (see Figure 5.11). This is the most straightforward technique as the motions of sign language are semantically meaningful. However, motion feature query remains interesting to decide between several motions with the same annotation label [173], or to extract motions at a finer level than gloss.

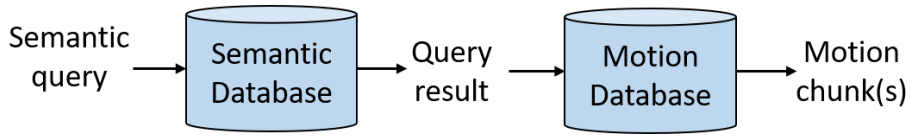

Figure 5.11 – Semantic motion retrieval.

Assuming that the quality of the captured database and post-processing is correct, the **playback of a pre-recorded sign** will give an accurate and realistic motion. Indeed, the main advantage of MoCap technologies is that it preserves the human qualities of motion. The motion and dynamics of the generated sign will therefore be perceived as natural and credible, thus increasing the acceptance of an avatar driven by such a method [65]. However, while this technique can be used to generate new utterances by concatenating existing signs as we will see in Section 5.3.2.1, the generation of new signs, not existing in the original database, or the editing of a sign in order to adapt it to a new context are not the objectives of the playback approach. The utterances produced using this technique are therefore limited by the number and variety of the recorded data.

The existing database can also be augmented by **editing** the pre-recorded motion using synthetic animation techniques. For example, in [64], some signs are created by temporally inverting the direction of the hand motion. For example, the sign [*PRENDRE*] (to take) can be changed to [*DONNER*] (to give) or [*AIMER*] (to like) to [*NE PAS AIMER*] (to not like) by inverting the hand trajectory of the sign. Gerard, the avatar of [54], can generate the same sign with different styles by modifying the dynamic properties of the recorded motions.

Motion warping is also a motion editing technique that can be applied to add variability in sign language generation. It consists in altering a motion of the database by changing its trajectory while keeping the kinematic properties of the motion [176]. This is a promising technique for data-driven synthesis of SL, but so far little work has been dedicated to

Figure 5.12 – The Sign360 application of MoCapLab [174], [175].

the development of this technique. Other approaches use Dynamic Time Warping to synthesize the style of the signs in terms of temporal variability of the motion, whether at a phonological level [177], or at an utterance level [178].

Machine learning methods are another way to reuse and generalize motion data. It relies on the captured data to extract knowledge and models that can provide new plausible and contextualized movements. Carreno *et al.* [179] present a very thorough survey on motion synthesis based on machine learning methods. For the particular case of sign language synthesis, Huenerfauth *et al.* [13] studied and synthesized directional verbs like [ASK], [MEET] or [SEND] using *MoCap* data of SL performances from fluent signers as a training data set to learn their synthesis models.

Examples of data-driven avatars:

— *Tessa* [180], a British Sign Language avatar, and *Simon* [181], a Sign Supported English [2] avatar both take advantage of the playback technique.

— Playback of signs in Swedish Sign Language was also done by Alexanderson *et al.* [67], but the main concern of their work was the study of *Motion Capture* and skeleton reconstruction.

— The *Sign3D* project combines both playback and sign synthesis editing techniques [69].

2. Sign Supported English (SSE) : the signs from British Sign Language are used in the order that the words would be spoken in English. It is a code and not a language as the grammar used is the one of English.

— For commercial applications, *Sign360* of the French company MoCapLab presents a French Sign Language avatar driven by pre-recorded gestures [174], [175] (see Figure 5.12).

Data-driven synthesis techniques are effective ways to produce natural looking motions. However, the quality of the resulting avatar animations depends on the granularity of the annotation and on the size and content of the initial corpus. Editing techniques to generate new signs are still rare in the SL animation field.

5.3 Utterance Synthesis

In the previous section, we reviewed the processes and techniques for isolated sign synthesis. This section is dedicated to the synthesis of full utterances. An utterance is a set of signs, performed simultaneously or sequentially, that represents the statement of an idea. It is close to the concept of "sentences" in oral languages. Utterances are composed of signs in their citation forms and of inflected signs (see Section 3.3.1.2).

5.3.1 Utterance Representation

Since an utterance is composed of a set of signs, it is possible to take advantage of the representations of the signs seen in Section 5.2.1 and in Section 5.2.2 to specify an utterance. However, a simple concatenation of these signs in their citation form would be incorrect in the same way that a sentence without conjugating the verbs would be incorrect in oral language. It is important to take into account the grammar and semantics of SL in order to inflect the signs that need to be inflected and to coordinate the different body channels of the avatar. First, we describe four challenges of the representation of an SL utterance and show, using examples, how the representation by a sequence of glosses may be inadequate. Then, we describe and show the advantages and disadvantages of four utterance representations.

5.3.1.1 Limitation of the Representation with a Sequence of Glosses

Many machine translation systems describe an utterance as a simple sequence of glosses. As the order and nature of the signs is given by this representation, it is suited to concatenative synthesis provided the presence of a sign database annotated on the same gloss-level as the specification.

However, utterance representation is a complex and non-sequential problem for which a sequence of glosses is not the appropriate solution. The main challenges of utterance representation are listed here. We illustrate each challenge with an example that highlights the limitations of the gloss representation.

a) Non Manual Features (NMFs) synchronization Facial expressions, gaze and torso directions are channels that are sometimes not coordinated with manual movements in order to add syntactic or contextual information to an utterance.

The direction of the torso, shoulders and gaze, for example, are often used in **role-shift** cases (see Section 3.2.2) and are not synchronized to a particular sign. In this case, the signer takes the role of the person, animal or object he/she is describing or whose words he/she is telling. If he/she repeats a dialogue he/she has heard between a person A and a person B, A's statements can be transmitted with a movement of the shoulders, chest and gaze to the left, while B's statements will be reported with a movement to the right. These movements are associated with the whole statements and not with a particular sign. It is therefore an overlay at the statement level.

Similarly, **negation** can be made with a repeated movement of the index finger (sign [NOT]) but is often supported by negative headshake that can begin before and end after the sign [NOT] itself. For example, the sentence "I don't dance" can be represented by the following gloss sequence:

$$[\text{I}][\text{DANCE}][\text{NOT}] \tag{5.2}$$

A signer will often make the negative headshake from the beginning of the [DANCE] sign until the end of [NOT] (see Figure 5.13). However, the representation by a sequence of glosses as shown in (5.2) does not provide this information. The [DANCE] sign is not affected by the negative aspect of the sentence, which can result in incorrect sentences in synthesis. One solution is the use of *parameterized glosses* in which contextual information is provided. Parameterized glosses are used by various researchers [1], [182], [183] but no standard exists.

Thus, (5.2) would become:

$$[\text{I}][\text{DANCE\_not}][\text{NOT}] \tag{5.3}$$

Where [DANCE_not] and [NOT] include the negative headshake. But even so, the movement of the head would be synchronized on each of the signs and not on the whole

{[DANCE][NOT]} as it should be. Glosses, simple or parameterized, result in an *over-synchronization* (the synchronization is carried out on too many synchronization points) at the gloss level.

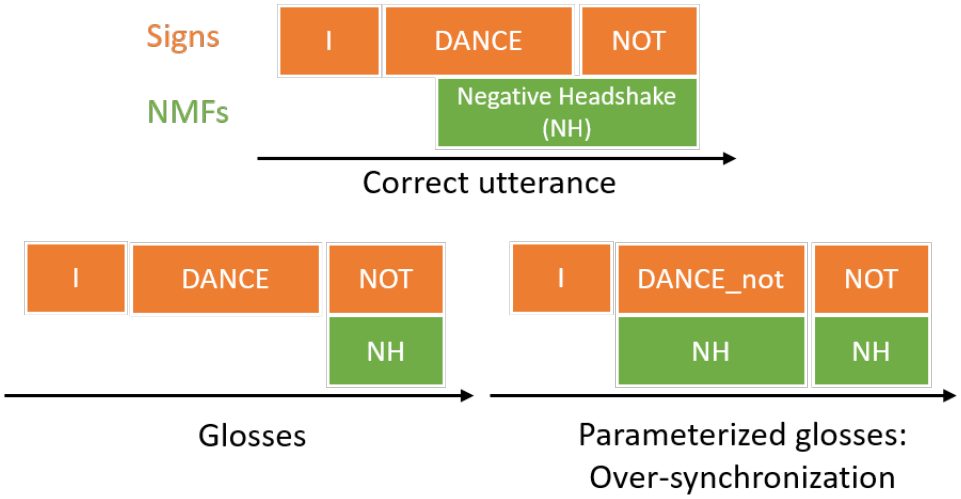

Figure 5.13 – Example of the sentence “I did not dance”. Top: the correct synchronization between the signs performed with the hands and the NMFs. Bottom left: the result of a simple sequence of glosses. [DANCE] does not include the negative headshake. Bottom right: an example of over-synchronization with parameterized glosses.

b) Sign Simultaneity In addition to the desynchronization of the NMFs, two signs can be performed at the same time with one sign or part of a sign being held while the next sign is performed. This situation often occurs when using **proforms** to describe a scene. Indeed, the aim is to show the position and action of one entity in relation to another.

It is a slightly different case from the synchronization of NMFs where the channels involved had little or no impact on the realization of the signs. In terms of animation, in the case of the NMFs synchronization, the channels involved could be animated by an independent controller. In the case of **scene description**, the entities, often impersonated by the two hands, move and act in relation to each other. The required animation is more precise and controlled.

Once again, the sequence of glosses, precisely because of its sequential aspect, is insufficient.

Let us take the example of the sentence "The frog jumps into the lake". This sentence should be signed with two proforms: one for the lake with the non dominating hand

performing a wide C-shaped configuration to show the outlines of the lake, and one for the frog with the dominant hand. The latter jumps into the lake, which results in a jumping movement of the dominant hand towards the space delimited by the non dominant hand.

A parameterized sequence of glosses where the proforms are indicated by the suffix "_pr" would give :

$$[\text{LAKE}][\text{LAKE\_pr}][\text{FROG}][\text{FROG\_pr}][\text{JUMP}] \tag{5.4}$$

Again, a sequence of glosses does not transcribe the simultaneity of the signs.

c) Sign Inflection The contextualization of the signs leads to modifications of the citation form of some signs in order to insert them into the utterance. Inflections can be of several types (see Section 3.3.1.2) and characterized by a change in hand configuration (e.g., proforms), trajectories (e.g., directional verbs) or amplitudes (e.g., size specifiers). Simple glosses do not provide such information.

For instance, in French Sign Language, the sequence "I give a small balloon to him" encompasses the three phenomena and will be done with only two signs : (i) a sign to specify the given object ("the small balloon") with a size specifier and (ii) the sign [GIVE] with the hand configuration corresponding to a small balloon ('O' configuration) and a motion going from the signer to the side.

It can be represented using different gloss sequences which will have an impact on the resulting animation. With a sequence of simple glosses, this would give:

$$[\text{I}][\text{HE}][\text{SMALL}][\text{BALLOON}][\text{GIVE}] \tag{5.5}$$

This would certainly result in the concatenation of many isolated signs in their citation form. The utterance would not be grammatically correct.

However, the use of parameterized glosses can cover this type of inflection:

$$[\text{BALLOON\_small}][\text{GIVE\_}\textit{'O' config}\text{\_I} \rightarrow \text{he}] \tag{5.6}$$

The representation (5.6) only contains two glosses, a first gloss naming the specified object and a second gloss corresponding to an inflected sign that can be retrieved from a database or generated on-the-fly. This parameterized representation will certainly achieve a result more similar to what a real signer would do. However, it implies a very large

database containing an important number of inflected signs or a synthesis engine capable of understanding the syntax of the representation and of creating the corresponding sign.

d) Timing Information There are two philosophies in timing management for signing avatars. This timing can be (i) indicated explicitly, in a relative or absolute manner, in the representation of the utterance at the input of the synthesis model (in this case, it is an additional constraint for the model) [146], or (ii) computed by the synthesis model according to the constraints of the system [69], [156]. For example, if the sentence is a concatenation of movement segments present in a database, the timing will be constrained by the length of these segments. A sequence of gloss representations, whether parametric or not, does not account for the timing but only for the linear order of the signs.

5.3.1.2 Examples of Representations

We describe here four ways of representing the SL utterances, each addressing different problems.

a) EMBRScript: Representing Absolute Timing The *EMBRScript* represents non-contextualized signs in the form of *k-pose-sequences*. With this script, an utterance is considered as a sequence of glosses, except that the timing of the signs is explicitly indicated in an absolute way [146]. Apart from this timing information, the shortcomings of this representation are the same as those of the simple gloss sequence representation (no consideration of the context and synchronization of non-manual channels).

Example:
I 300 → 1190
DANCE 1220 → 2620
NOT 2650 → 3000

b) ATLAS and HLSML: Representing Inflections For the *ATLAS* project of the Italian team of Turin, a focus is done on the inflected signs: a sign is defined as the combination of a base sign (the citation form) and context-dependent modifiers. These modifiers can be of different types: sign relocation, speed modification, trajectory modification, sign resizing, sign iteration or hand configuration modification [155], [156].

Their article [155] takes the example of the sentence "Cloudy at north-east. During the evening, cloudiness increases at north-west" which is interpreted in their animation

language as:

```
north-east;
zone (relocated top-left);
cloud (relocated top-left);
instead;
evening;
cloud (repeating and shifting from top-left
to top-right);
more (relocated top-right);
zone (relocated top-right);
```

In the same way, HLSML was developed by López-Colino *et al.* [184], [185] for their Spanish Sign Language avatar. It uses an XML-based notation to represent inflected signs (see Figure 5.14) or to change the location of a sign. Moreover, time information can be included in the representation to constrain the realization of the signs.

```
<!DOCTYPE hlsml SYSTEM "hlsml.dtd">
<hlsml>
<sentence value='sentence2' language='lse'
                                        tag ='standard'>
 <sign name="TO_GIVE">
  <signclassifier value ="configuration">
   <sign name="clBOOK"/>
  </signclassifier>
 </sign>
</sentence>
</hlsml>
```

Figure 5.14 – The representation of the inflected sign [TO GIVE] in the expression "to give a book" in HLSML. The hand configuration used for the sign [TO GIVE] is the one of the sign [BOOK] named "clBOOK" ("book classifier").

An utterance is therefore defined as a sequence of contextualized glosses which are a form of parameterized glosses with the same limitations.

c) P/C Model: Representing Synchronization A common way to analyze a sentence structure in natural language processing for oral languages is to display it in the form of a syntax tree [186]. However, the multichannel aspect of SL makes it difficult to

describe an SL utterance as a syntax tree. In his work, Huenerfauth proposes to represent an SL utterance in the form of a $3D$ syntax tree [187]. He thus defines a formalism, called *Partition/Constitute (P/C) model* which is a $2D$ representation of a $3D$ syntax tree to which restrictive rules have been added. This representation allows to visualize from left to right the temporal axis and from top to bottom the body channels. The nodes of the $3D$ tree are represented by rectangles (leaves are the "atomic" rectangles that surround a text while the root is the rectangle that surrounds the whole, the other rectangles are the different branches). This representation allows to manage the coordination of the different channels between themselves as well as the use of certain proforms (see Figure 5.15 and 5.16).

There is no precise timing information but the "sequential ordering within channels and coordination relationships across channels" are made explicit.

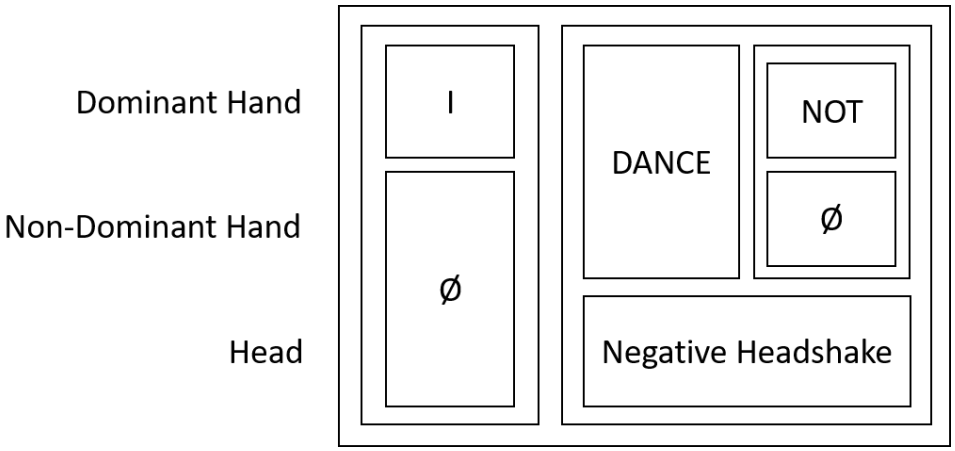

Figure 5.15 – Example of the P/C modeling of the sentence “I did not dance” in ASL. The coordination of the negative headshake with the signs [DANCE] and [NOT] is explicit (figure done using the representation described in [187]).

d) Azee: Representing Function-to-forms Associations With the *Azee* representation, the common assumption that an SL sentence is defined as a sequence of glosses is questioned. The originality of *Azee* is that it is based on the minimal linguistic assumption that the language is a system where observable forms are associated, in a systematic way, to a meaning. To capture those systematic associations, *Azee* implements *production rules*: invariant function-to-forms correspondences where a *function* is the desired semantic meaning produced by the *forms*, the "*visible states and movements of the body articulators*" [188].

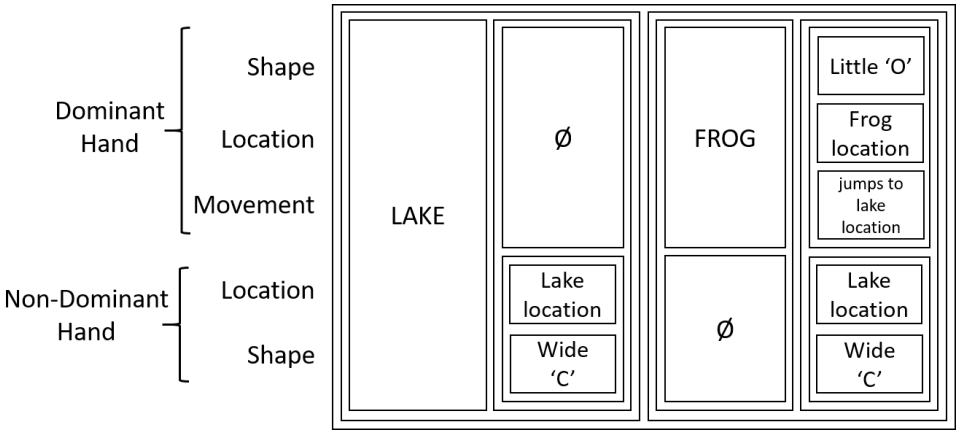

Figure 5.16 – Example of the P/C modeling of the sentence "The frog jumps into the lake" in ASL. The ∅ symbol for the non-dominant hand during the realization of the [FROG] sign makes it possible for the non-dominant hand to hold the classifier of [LAKE] (figure done using the representation described in [187]).

Those production rules capture, using the same process, the formation of isolated signs (in this case, the states of the different channels for the production of this sign will be the *forms*, while the meaning of the sign, the gloss, will be the *function*) and higher-level syntactic mechanisms such as the relationship between entities (e.g., "A is an instance of B") [189], [190]. Some rules can thus be parameterized according to the context (parameters A and B of the previous examples) and, since all mechanisms are put on the same level, the nesting of production rules is done in a natural and direct way. A and B can be isolated signs or complex syntactic functions. A rule tree allows to visualize this type of nesting (see example of rules in Figure 5.17 and a rule tree in Figure 5.18).

In practice, the production rules are derived from the analysis of the content of a corpus until an invariant *form* is found for a large number of instances of the same *function*. For example, if a left-to-right headshake is found with many instances of negative utterances, a production rule associating the form *left-to-right headshake* to the function *negation* will be created. These production rules can be written in the form of temporal scores (see Figure 5.19) and are unambiguously interpretable by a synthesis system. The rules are defined without *a priori*: the lexical sequence as the base of an SL production is therefore questioned. The sequence of signs is seen as one possible *form* in the same way as an eye blink or a headshake. In addition, if timing information has been found consistently for many instances of the same *function*, this timing is added to the *forms* of the *function*.

Moreover, there is ongoing work in order to associate *Azee* formalism with a pictogram

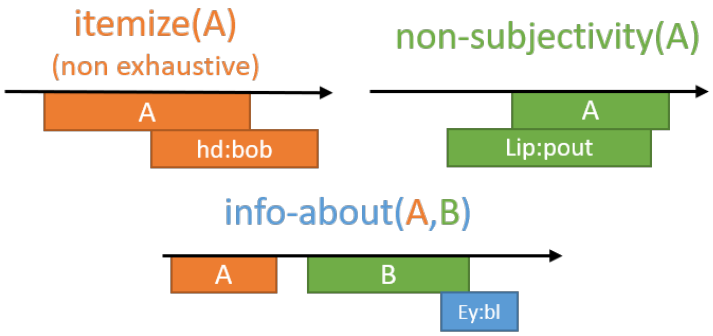

Figure 5.17 – Individual production rules of Azee (figure inspired from [191]). The *itemize(A)* production rule, for instance, designate the function-to-form association used to do a non-finite enumeration. It makes it possible to create expressions like "theaters, restaurants, etc." in which case, the *itemize(A)* rule is used twice with the parameter "A" taking successively the values "theater" and "restaurant". When performing the enumeration, a head movement (hd:bob) is done at the end of each item.

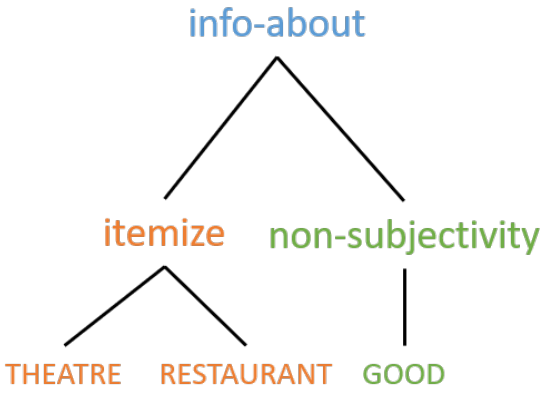

Figure 5.18 – The rule tree of the sentence "Theaters, restaurants, etc. are usually deemed good" (figure inspired from [191]).

Figure 5.19 – The *Azee* representation of the sentence "Theaters, restaurants, etc. are usually deemed good" in the form of a sign language score (figure inspired from [191]).

representation to make it more accessible to the deaf community and obtain a writing system that can be directly interpreted and synthesized by an animation system [192].

Table 5.2 provides an overview and a comparison of the utterance representations.

Representation	NMFs synchronization	Sign Simultaneity	Sign Inflection	Timing Information
Sequence of glosses	✗	✗	✗	✗
Sequence of parameterized glosses	✗	✗	✓	✗
EMBRScript [146]	✗	✗	✗	✓
ATLAS [156], [193]	✗	✗	✓	✗
HLSML [184], [185]	✗	✗	✓	✓
P/C Formalism [187]	✓	(✓)	✗	✗
Azee [188], [189]	✓	✓	✓	✓

✓: Good management of the functionality
(✓): Partial management of the functionality
✗: Functionality not managed

Table 5.2 – Comparison of the utterance representations.

5.3.2 Utterance Synthesis Approaches

Two main utterance synthesis approaches are often distinguished: the **concatenative** and the **articulatory** synthesis [160] (see Figure 5.20). In the case of concatenative synthesis, the utterance objective is considered as a set of smaller objectives. Concatenative synthesis involves a motion/sign database and consists in concatenating small chunks of existing data while articulatory synthesis computes the sign language utterance directly from the gesture specification. Hybrid avatars with animation systems taking advantage of both techniques are also emerging.

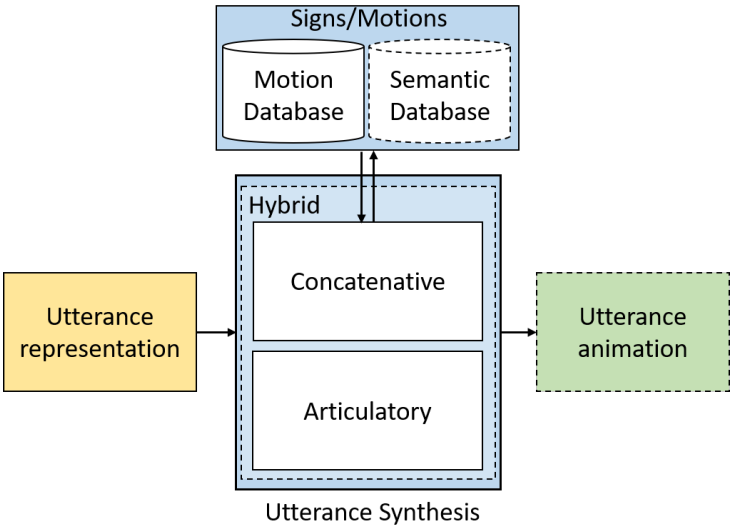

Figure 5.20 – Overview of the utterance synthesis techniques.

5.3.2.1 Concatenative Synthesis

It is the process in which chunks of pre-recorded or pre-synthesized motions are concatenated successively. Motion interpolation or blending techniques are used to build the transitions between two chunks of motions. Motion blending consists in doing, at each frame, an interpolation of motions present in the database to create new motion having the characteristics of the initial motions. The quality of the result strongly depends on the length and granularity of the chunks of motion. The simplest and most common approach is to consider motions at a gloss-level, but smaller motion chunks can also be used, leading to a more precise control of the avatar animation. Motion synthesis is therefore based on an annotated database of signs (or finer grained motions) that is queried to compose the utterance. The annotation of the data is essential: the granularity of the synthesis is restricted by the granularity of the annotation, and the presence or absence of artifacts in the final animation depends partially on the quality of the segmentation of the data. The database is built and annotated offline while the sign concatenation can be done online. Utterance representations based on sequences of glosses are particularly suited to this synthesis technique. The following paragraph details the different avatars technologies that make use of concatenative synthesis for utterance synthesis depending on the nature of the content of the original database.

a) Using Hand-Crafted Animations In this case, concatenative synthesis relies on a database of hand-crafted motions and on motion interpolation to soften the transitions between the concatenated gestures. *Paula* [153] and *Elsi* [125], the avatars presented in Section a), rely on this principle. *Paula*'s motion is created in real-time by automatically combining and possibly modifying the hand-crafted key postures in accordance to the desired sign language utterance. *Elsi* can sign full utterances created by concatenation of isolated signs designed by rotoscopy. Unfortunately, she can create very few novel utterances as rotoscopy does not provide for context-dependent signs. However, this method suits the application of *Elsi* who does not need to convey novel messages since she has to provide information in train stations to deaf passengers by combining pre-recorded fill-in-the-blank sign language utterances and isolated signs.

Examples of avatars using concatenative synthesis with hand-crafted data:

— *Paula* [153].

— *Elsi* [125].

— The Turkish Sign Language avatar of Yorganci *et al.* [159].
— The Spanish Sign Language avatar of López-Colino *et al.* [184], [185].

b) Using Automatically Generated Signs It is also possible to build a sign language utterance by editing and concatenating signs extracted from a database of automatically keyframed or of procedurally synthesized signs. Annotation tags are queried to retrieve the relevant keyframes/signs from the database. The avatars relying on automatic keyframing for the generation of isolated signs (Section b)) often synthesize utterances by simple concatenation of whole signs.

Examples of avatars using concatenative synthesis with automatically generated signs:
— The EMBR avatar of Kipp *et al.* [145], [146].
— The avatar of Krnoul *et al.* [162].
— The parametric avatar of Irving *et al.* [158].
— The *eSIGN* avatar [194].

c) Using MoCap Data Concatenative synthesis on *MoCap* data consists in combining and concatenating previously captured motion chunks (often corresponding to a sign or gloss segment). The transition between the motion chunks can be done using different kinds of interpolation and blending methods [52]. This concatenative synthesis produces realistic motions on playback sequences but the quality of the utterance as a whole strongly depends on the quality of the synthesized transition segments. Moreover, the signs will be played by the avatar the exact same way as they were recorded. It means that the objects described in the new utterance will have the same aspect as the ones recorded. The creation of novel utterance is thus limited by the motions available in the database.

Examples of avatars using concatenative synthesis with MoCap data (see section 5.4.2 for more details):
— The Sign3D avatar [69]. This system, characterized by high fidelity captured motion (both corporal and manual data, facial expression and gaze direction), proposes data-driven synthesis at the sign/gloss level.
— The SignCom avatar [64]. This system allows for concatenative synthesis at a phonological level.

5.3.2.2 Articulatory Synthesis

One of the drawbacks of concatenative synthesis, regardless of the nature of the database, is that the input is often a sequence (of glosses or motions) while simultaneous spacial phenomena (e.g., proforms or iconicity) are often present in sign languages and cannot be fully captured by concatenation alone. Therefore, another way to build utterances in sign language is to create them on-the-fly following an utterance specification and not depending on a predefined database of fixed motion chunks. Iconicity, proforms, transitions between the signs or simultaneous phenomena can be incorporated directly into the sign description so as to take the context into account.

Sentences build this way can be precisely controlled but the work of describing the utterance using a sign or a gesture specification (taken as input of such a system) can be extremely tedious. Furthermore, the resulting animation lacks realism and can sometimes be rejected by the Deaf community. To our knowledge, only the *GessyCA* system [167] uses the articulatory synthesis alone to build utterances. The sequence of signs described with *QualGest* is assembled by combining sequentially and in parallel the atomic *gestems* using synchronization operators. Procedural approaches to animate an avatar at a sign level from a sign representation are commonly used but they are rarely extended to the utterance level. However, hybrid approaches that mix articulatory and concatenative principles are more developed.

5.3.2.3 Hybrid Synthesis

Hybrid models take advantage of the strengths of concatenative and articulatory synthesis. The CUNY American research group worked on both procedural animation techniques and data-driven synthesis approaches using *MoCap* data and compared the two approaches in [66]. The sign language research team of the University of East Anglia also studied both technologies, first with *Tessa* (concatenative) [180], followed by the eSIGN project (articulatory) [195]. They proposed to use *MoCap* data to add realism to their procedurally animated avatar.

DePaul University research group implemented this idea on *Paula*, their avatar animated with hand-crafted keyframes [154]. Moreover, *Paula* was recently improved with IK solvers that procedurally modify the hand-crafted animations in order to synthesize context-dependent mechanisms such as proforms making it a fully hybrid avatar [189], [196]. The Italian Sign Language avatar of Lombardo *et al.* [155], [156] defined a contex-

tualized sign as the combination of a base sign (defined mainly by manual and automatic keyframing but also by *MoCap*) and context-dependent modifiers (procedural methods or hand-crafted poses). It showed great promises but the project seems to have been frozen as no subsequent article has been published on the subject. In the same philosophy, the ASL avatar of Adamo *et al.* [197] relies on a multilayer system with a concatenative synthesis engine to query individual signs that are altered by procedural prosodic modifiers.

5.4 Existing Systems

In this section, we present three SL avatars using different sign and utterance synthesis approaches. To have a more exhaustive overview, Table 5.3 lists and describes the existing avatars.

5.4.1 JASigning and AnimGen

The Java Avatar Signing (*JASigning*) system is a multiplatform tool for the synthesis of any sign language [198]. It is freely available for research purposes [3]. It integrates *AnimGen*, an animation engine for SL synthesis developed for the ViSiCAST [168] and, later, the eSIGN [194] projects.

For isolated sign synthesis, *AnimGen* takes as input the *SiGML* notation [138] of a sign and translates it into low-level parameters for the animation based on inverse kinematic controllers.

For utterance synthesis, the concatenative approach is preferred: the signs are assembled to form utterances. Some inflections can be added to the isolated signs to take the context into account: the location, direction, gaze and facial expressions of signs can be changed. However those modifications can only be applied at a sign level creating a possible over-synchronization.

The *AnimGen* module was used internationally in numerous works from 2001 until today [139], [168], [169], [195], [198]–[202]. Notably, Hanke *et al.* [200] whose work focuses on the study of timing differences on the SL channels, used *AnimGen* to test their hypothesis on the synchronicity between the hand configuration with respect to the hand location. In more recent work, *JASigning* constituted the basis of Ebling's animation module to test her machine translation system [199].

3. `http://vh.cmp.uea.ac.uk/index.php/JASigning`

The *AnimGen* module is suited for simple translation tasks as it allows the creation of new signs and utterances. Indeed, it only depends on a sign representation (*SiGML*) and not on annotated data. However, the motions generated lack the naturalness of human motion. In addition, it depends on the concatenation of signs to create utterances: simultaneous mechanisms cannot be captured by *AnimGen* and the inflections that can be added to the signs are limited by *SiGML*.

5.4.2 SignCom and Sign3D

The avatars *SignCom* [64] and *Sign3D* [69] aim to animate French Sign Language avatars with natural and realistic movements from captured human data. They rely on two databases: (i) a raw motion database in which the captured motion is stored as a hierarchical structure (the skeleton) with a set of transformations applied to each joint, and (ii) a semantic database that serves as a mapping between the gloss level annotations and the movements contained in the first database. The *Sign3D* system operates at a sign/gloss level. Following a stereotyped syntactic scheme, it allows to precisely synthesize novel utterances by replacing signs or groups of signs. The building of utterances is done following an interactive graphical language [69]. The *SignCom* system operates at a phonological level on the different channels (hand configuration, hand movement, facial expression or gaze direction) (see Figure 5.21). A hierarchical scripting language allows for taking into account the sign modifications while the animation of the skeleton integrates the various parallel controllers.

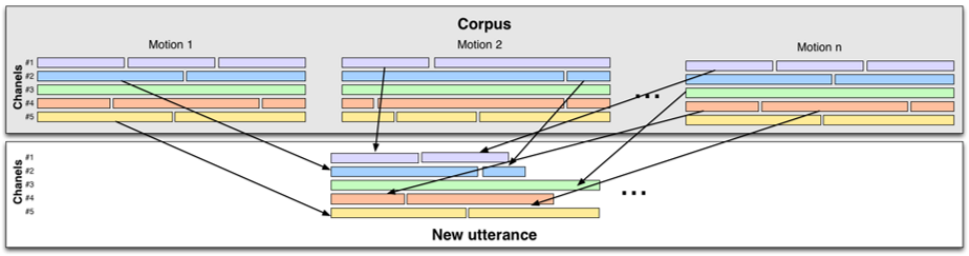

Figure 5.21 – Multitrack concatenative synthesis (image extracted from [64]).

For the synthesis of isolated signs, the appropriate sign is retrieved from the movement database. To do this, the semantic database makes it possible to find the movement(s) labeled with the desired gloss. If several movements correspond to the gloss, a choice can be made using various criteria: length of the time segment, profile of the movement, etc. The synthesis of novel signs remains limited by the content of the *MoCap* database. For

utterance synthesis, the captured and possibly edited captured signs are concatenated and the transitions are synthesized by motion interpolation or motion blending. An additional criterion for the retrieval of signs can be added: distance from previous and/or following movements [52].

Despite different isolated sign synthesis techniques, *JASigning* and the avatars *SignCom/Sign3D* have the same drawback regarding utterance synthesis: the synthesis mostly uses the signs in their citation form and do not integrate complex mechanisms of SL such as proforms or size and shape specifiers.

5.4.3 *Paula-Azee*

Paula (see Figure 5.22) of DePaul University is an American Sign Language avatar based on a multi-track animation engine while *Azee* is a language modeling system, first used for French Sign Language, that aims to represent the syntax of SL utterances in a non-sequential way (see Section 5.3.1) [203]. *Paula* and *Azee* were developed separately but they both rely on a multi-track system which makes their connection possible.

Paula is a hybrid system mainly animated with hand-crafted keyframes but that can also take advantage of procedural approaches to add precision or realism. To build the set of hand-crafted animations of the isolated signs, the *PDTS* classification of Jonhson & Liddell (that defines signs as a sequence of postures and transformation segments) is used as a template to manually draw the keyframes of *Paula* [153], [154]. Procedural animation of the spine can be added to increase the naturalness of the animation.

For utterance generation, the hand-crafted animations of *Paula* can be modified procedurally thanks to IK solvers in order to synthesize proforms or spatial referencing mechanisms [189], [196]. Moreover, *Paula* uses a multi-track system that is very similar to the *Azee* sign score shown in Figure 5.19. A mapping between *Paula*'s utterance specification tool and *Azee*'s formalism was created allowing *Paula* to be driven by *Azee*'s linguistic modeling [189].

Figure 5.22 – *Paula* from the DePaul ASL Avatar Project (image extracted from [204]).

Year	Name of the avatar and/or the project	Input of the animation system (specification language)	Sign synthesis technique	Utterance synthesis technique	Targeted applications
1982	Shantz Avatar [151]	List of fingers angles	**Keyframe**: hand-crafted	/	Study of SL perception
1999-2003	*SignAnim* [181] & Tessa [180]	Gloss sequence	**Data-driven**: play-back	Concatenative	Subtitle-to-SSE translation & Face-to-face **speech-to-sign translation**
1999-2001	*GessyCA* [143], [148]	QualGest	**Procedural**: with IK-based controllers	Articulatory	Animation of gestures for human communication
2001-2004	*ViSiCAST* [168]	SigML	**Procedural**: with dynamic controllers	Concatenative	Broadcasting, Face-to-face transactions and Interactive Internet info
2005	*Vsigns* [161]	SWML	**Keyframe**: automatic using kinematics	Concatenative	SL **dictionnary** and/or SL editor for newscast animation
2006	*Gérard* [54]	Gloss + biomechanic features	**Data-driven** : augmented database	Concatenative	Expressive animation of SL
2007	*Elsi* [125]	Video footage + gloss sequence	**Keyframe**: hand-crafted	Concatenative	Train station pre-recorded information
2007-2008	Czech signed speech avatar [162]	HamNoSys + NMFs	**Keyframe**: automatic using kinematics	Concatenative	**Speech-to-sign translation**
2008	Greek SL avatar [163]	HamNoSys + NMFs	**Keyframe**: automatic using kinematics	Concatenative	**Text-to-sign translation**
2009	LSF Avatar [165]	Zebedee	**Keyframe**: automatic using kinematics	Concatenative	Sign Synthesis
2010	EMBR Avatar [145]	EMBRScript	**Keyframe**: automatic using kinematics	Concatenative	Animation of Embodied Conversational Agent

2010-2016	*JASigning* [198], [199]	SigML	**Procedural**: with dynamic controllers	Concatenative	Multiplatform SL avatar, **Text-to-Sign translation**
2011	*SignCom* [64]	*MoCap* & Gloss Sequence	**Data-driven**: augmented database	Concatenative	Interactive assisting Internet technologies, **translation**
2011	Italian SL Avatar[155]	Gloss + modifiers	**Keyframe**: hand-crafted	Concatenative + modifiers	**Text-to-sign translation**, Virtual interpreter
2012	LSE Synthesizer [184], [185]	HLSML	**Keyframe**: hand-crafted	Concatenative + modifiers	Multimedia applications, inflected sign synthesis
2015	Huenerfauth [13]	Huenerfauth	**Data-driven**: machine learning	Concatenative	Directional verbs study
2016	Yorganci [159]	*MoCap* & phonological notation	**Keyframe**: hand-crafted	Concatenative	**Educational** tool for deaf children
2016	*Sign3D* [69]	*MoCap* & Gloss sequence	**Data-driven**: augmented database	Concatenative	Interactive assisting Internet technologies, **Translation**
2016	Libras Avatar [142]	XML version of Movement/Hold	**Data-driven/Keyframe**: *MoCap* keyframes	/	**Translation** of written textbooks for children
2016-now	*Paula* [153], [196]	SLPA/Azee	**Keyframe**: hand-crafted and automatic using kinematics	Hybrid	**Translation**

Table 5.3: Non-exhaustive list of the existing sign language avatars.

5.5 Summary and Discussions

Previous work on sign language avatars do not often discriminate isolated sign synthesis from utterance synthesis. However, we noted that, for a given avatar, signs and utterances are rarely built in the same way. The construction and animation of utterances call for an extensive knowledge of the linguistic specificities of sign languages. As sign languages are multilinear languages that involve the movement of several body parts in parallel as opposed to the sequential restriction of the oral medium, a particular attention must be given to the construction and synchronization of full utterances.

Isolated signs can be created using hand-crafted or automatically computed keyframes, procedural animation or data-driven techniques.

Keyframe-based techniques (either hand-crafted or automatically generated) give a non-continuous definition of motion where each keyframe is a given posture of an avatar at a given time. As the number of keyframes is too small to define a smooth motion, interpolation between two consecutive keyframes must be performed. The resulting motion greatly depends on the definition of the keyframes and on the complexity of the interpolation. Hand-crafted animations require a tedious process for which the realism of the results depends on the skills and choices of the graphic designer. They are generally based on a visual representation of signs (e.g., video recordings or drawings). Automated keyframing animations are based on keyframes generated using isolated targets and forward and inverse kinematics algorithms.

Procedural techniques take advantage of the temporal control of systems (whether kinematic or dynamic), using cost functions to be minimized to achieve objectives (e.g., moving targets), in order to create continuous motion.

In the cases of automatic keyframing and procedural techniques, the targets are generated thanks to a sign representation based on a phonological view of the sign. Both the automatic keyframing and procedural techniques are appropriate for generating precise, configurable and flexible animation but produce robotic and stiff motions since gaze direction, facial expression and body movement are often not stated in sign specifications where only the relevant manual features are described. In addition, the temporal specification of events is explicit sometimes leading to uncanny motions. Random noise and signal processing methods are often used to improve the final animation.

Still, those synthesis methods lack the expressiveness of human motion whereas, in

data-driven techniques, the resulting animation has the authenticity of natural human motion without needing to add special treatments. *MoCap* is thus a great tool for linguists and computer animators to analyze and synthesize motion. It can be a way to find motion laws using statistics on the data or to observe some linguistic phenomena of interest. However, the calibration, capture and post-processing of the data are complex, tedious and time-consuming. The *MoCap* equipment can be invasive with the presence of markers or sensors potentially impacting the realization of the signs. Moreover, the use of data leads to the development of optimized motion retrieval techniques to prevent slowing down the animation process. In addition, new sign language utterances are hard to generate from a limited set of motions in the database and context-based variation in the captured motions is not easy to synthesize which is a problem considering the great iconicity and variability of signed languages. Machine learning methods are a promising way to create new content from a limited *MoCap* database even though, as data-driven techniques, they will still produce utterances biased by the input data.

Sign representations can be the basis for precise, flexible and fast generation of movements allowing for real-time animation. However, the exploitation of sign representations often requires the prior manual intervention of transcribers to perform the mapping from a specific gloss to the lower-level sign representation. The automation of the gloss-to-representation mapping is an open issue that greatly depends on the chosen sign representation. Indeed, some representations are configurable – the difference between [BIG BALLOON] and [SMALL BALLOON] in *Zebedee* will only be a parameter to change (the radius of the balloon) while it will be necessary to modify each symbol if these same signs are described in *HamNoSys*. The gloss-to-representation mapping for illustrative/iconic signs can be automated using keywords (like BIG or SMALL in the previous example) while the correspondence for signs in their citation form can only be done manually.

In the case of utterance animation, the most common synthesis technique consists in concatenating chunks of motion (corresponding to the isolated signs previously defined or to sub-lexical structures) in order to create the utterance. This kind of synthesis is based on a database built offline and often relies on a sequence of glosses to retrieve the relevant motions. The transitions between the motions are generated using signal processing functions like blending or interpolation, and modifiers of the original signs are often added to add prosodic cues or to take into account the context of the sign. Such concatenative synthesis is an inherently sequential technique that can be seen as

contradictory to sign languages philosophy; however, concatenating chunks of motions along the temporal axis is not the only option of concatenative synthesis. Little work has been dedicated to the concatenation of smaller motions on the different channels of sign language (hand configuration, motion, orientation, facial expression, etc.) [64]. This is one of the focuses of our thesis work (see Chapter 8).

Articulatory synthesis aims to build sentences on-the-fly based on a sign or gesture specification. The animation can be computed from biocontrol or inverse kinematics models.

The benefits and drawbacks of articulatory and concatenative approaches are oddly complementary. On the one side, the articulatory techniques allow for a real-time generation of novel utterances but are poorly accepted by the Deaf community due to their inexpressive and robotic motions. On the other side, concatenative approaches based on *MoCap* technologies or hand-crafted signs allow for authentic, human-like motion. However, the variety of utterances that can be synthesized is limited by the initial corpus. Moreover, the sequential aspect of concatenative approaches, enforced by the sequential representation taken as input, does not do justice to the richness of the language. Hybrid models, taking advantage of the strengths of both the concatenative and articulatory approaches could result in a generic and well accepted avatar. Hybrid synthesis is a promising technique for sign language animation and will certainly be one of the main concerns of future work.

Furthermore, new, non sequential ways of specifying utterances like *Azee* [188] or the *P/C Formalism* [187], or specifications taking sign inflections into consideration such as *ATLAS* [193] or *HLSML* [184], are being developed to overcome the limitations of the current representations.

Even though Non-Manual Features (NMFs) are mentioned and that some of the described techniques are used to animate the face and torso of the avatar, the focus of this survey is the animation of the avatar's arms and hands. By no means we want to lessen the importance of NMFs which are paramount in any sign language animation, a great part of the meaning being conveyed by them. A survey dedicated to the animation of facial expressions for sign language avatars was proposed by Kacorri [122].

Finally, as sign language avatars are mainly intended to be used by the Deaf, their approval by the community is necessary. A deaf person will be badly receptive to a robotic

motion the same way a hearing person will to a robotic, unnatural voice. One of the main issues of the human-looking avatars is the risk of falling into the uncanny valley, first introduced by Mori in [205]. To prevent this risk, some research teams on sign language avatars choose to give a cartoon-like appearance to their virtual signers (e.g., avatar of Sign360 [175] or Adamo *et al.* [197]). Surveys assessing the acceptance of sign language avatars by deaf people have been made by Kipp *et al.* [206], by Adamo-Villani *et al.* [207], and by Lu *et al.* [65], [66]. They show that non-manual features like facial expressions and natural body movements are of great importance to deaf users. Such surveys provide precious insights on the acceptance of the sign language avatars. Perceptual evaluation of signing avatar animations by deaf consumers should therefore be performed systematically to ensure a good response to the technology.

Part II

Contributions

Chapter 6

LSF-ANIMAL: A Motion Capture Corpus in French Sign Language

Contents

6.1 Corpus Definition 119
6.2 Acquisition of the Data 124
6.3 Data Post-processing 128
6.4 Perceptual Evaluation of the Corpus 131
6.5 Summary and Discussions 144

Je me sers des animaux pour instruire les hommes.

Fables
Jean de La Fontaine, 1668 - 1694

This chapter is based on an article published in the proceedings of the Language Resources and Evaluation Conference (LREC 2020) [208].

Data are the basis of our research work. They constitute the raw material from which the information that is used to analyze and synthesize movements will be extracted. It is therefore essential to collect motion data that includes the mechanisms we want to study and recreate on an avatar. *MoCap* data have the advantage of seizing the observed movements with great accuracy.

This chapter presents the *LSF-ANIMAL* corpus, a *Motion Capture* corpus in French Sign Language (LSF) suited to the animation of signing avatars. Most of the work presented in this thesis was performed on this corpus. The existing video and *Motion Capture*

data sets for different sign languages have been reviewed in a previous chapter (Chapter 4). Section 6.1 describes the objectives and the content of the *LSF-ANIMAL* corpus. Section 6.2 presents the acquisition process of the data and the detailed marker set used. Section 6.3 details the post-processing and annotation steps. Finally, Section 6.4 presents the design and results of a perceptual evaluation of the quality of the data set.

6.1 Corpus Definition

When designing a sign language corpus either to study or to synthesize sign language utterances, a trivial solution might be to cover all the possible cases of sign production by recording all the existing signs in all the possible contexts. Huge data sets composed of Wikipedia pages or Twitter posts for example can be easily retrieved from the Internet for oral languages. However, this is impossible for sign languages for three main reasons. First, the signed equivalent of sentences databases like Wikipedia does not exist on the Web, especially *MoCap* databases. The time and memory resources needed to capture or simply obtain one sign language utterance far exceeds the time needed to get one written or spoken sentence. Secondly, sign languages use both the $3D$ space and the temporal dimension which leads to the production of an infinite number of combinations of the different physical channels. And finally, sign languages are not limited to their standard signs (the signs described in dictionaries): many sign language mechanisms such as proforms are as important as standard signs and depend strongly on the context of the sentence. As it is impossible to capture the language in its entirety, corpora are defined with respect to the phenomenon being studied resulting in databases dedicated to specific applications.

Each of the corpora presented in Section 4.1 has been designed with a specific purpose in mind. No available corpus in LSF was precise and exhaustive enough for our purpose. It therefore seemed necessary to us to define and build our own corpus adapted to our needs. The design of this database was based on previous work done in the team on *Motion Capture* and corpus content selection [209], [210].

The construction of a database is a decisive step in the thesis work because the initial data will constrain the work that can then be carried out. It is therefore a question of designing a thoughtful and coherent corpus in relation to the thesis objectives. In the following section, the synthesis objectives and the content of the database are presented.

6.1.1 Objectives

The objective of our corpus is dual. On the one hand, it constitutes the material to be analyzed in order to highlight motion laws, invariants and LSF phenomena. In this case, the data can be considered as ground truth and is used to make observations and to evaluate our synthesis results. On the other hand, the data becomes the synthesis material. We aim to generate new, natural and realistic LSF utterances based on the observations of the ground truth, and editing of the captured motions. This analysis/synthesis complementarity is paramount for our research work and the corpus is designed to handle this duality.

More precisely, we wish to study and synthesize three phonological components [5]:

1. **Various Hand Configurations (HC) of LSF**. Keeping our synthesis goals in mind, we aim to study two phenomena: the transition from one configuration to another and the synchronization of the HC with respect to the other components such as hand orientation and placement. The corpus must then incorporate these HC in various linguistic constructions: in signs containing a change in the hand configuration – e.g., the sign [*SALON*] (living room) begins with an 'O' and ends with a 'C' configuration –, as well as in full utterances in which the chosen HC appear in a natural and contextualized way. Moreover, for synthesis purposes, the isolated HC must be captured to serve as a basis to our synthesis system.
2. **The placement of the two hands in the signing space**. The placement of the hands can designate both (i) the global area where the sign is produced which does not change during the sign production but which can change depending on grammatical inflections [1], [9], and (ii), at a lexical level, the discrete area or the specific coordinates where the hand is positioned at a precise time. Both are interesting for our study. We need to capture instances of the same sign placed at different locations of space and signs in which the hands are not static and whose position varies. The capture of full utterances will naturally provide various placement features.
3. **Hand movement**.Hand movements with different trajectories (straight line, circular, waves, etc.), and with zero or more repetitions must be incorporated into the database.

In addition, we wish to study **coarticulation** mechanisms in full utterances. In order to measure the impact of the sign $N-1$ and $N+1$ on the sign N, it is necessary to record natural LSF utterances with various combinations of the same signs.

Finally, we want the corpus to be used to test automatic annotation algorithms with different levels of granularity (e.g., gloss, hand configuration, placement). For this purpose, the presence of various types of data streams, such as isolated HC, isolated signs and full sentences, is beneficial.

We used those constraints to define the content of our corpus.

6.1.2 Content of the *LSF-ANIMAL* Corpus

To meet the requirements detailed in the previous section, the *LSF-ANIMAL* corpus contains five subsets (see Table 6.1).

The first subset constitutes a list of the most common **hand configurations of LSF**. 39 hand configurations have been chosen with care by comparing five sources of different nature: (i) a LSF teacher, (ii) the hand configurations annotated in the Sign3D Corpus [69], (iii) the International Visual Theatre book which is a reference for LSF grammar and vocabulary [9], (iv) the research book of [3] and (v) a textbook to learn LSF [211]. The configurations that were common to at least 4 of the 5 different sources were chosen. To these 39 configurations, we added the missing letters (e.g. the 'M' is only present in 2 sources but, being a letter, it is automatically put in the list of configurations). This part contains thus all the letters used for fingerspelling in LSF [1] which can be used to spell some words and names, and 22 isolated configurations. All those 41 configurations were executed with both hands [2].

The second subset is composed of **isolated signs**. Three types of signs have been chosen to address three types of needs:

— 11 signs containing a change of hand configuration within the sign (like the sign [*WEEK-END*] in LSF which begins with the 'W' and ends with the 'E' hand configurations). Those signs can be used as examples and ground truth to synthesize the passage from one configuration to another and to study the coarticulation of the hand configuration component within a sign. In French, the signs are [*OK*], [*COCA*]

1. Not each of the 26 letters of the fingerspelling alphabet is a hand configuration. The 'Q' letter, for example, possesses the same configuration as the 'G' letter (thumb and index fingers stuck together and in the up position while the other fingers are in a folded position). However, the alphabet alone represents 19 different configurations.

2. 7 other hand configurations were found during the annotation of the corpus for a total of 48 hand configurations labels (see Section 7.2)

(Coca-Cola), [*OR*] (gold), [*FIN*] (end), [*LSF*], [*DODO*] (nap), [*LA*] (to be present), [*50*], [*SALON*] (living-room), [*WEEK-END*], [*VAVA*] (indication of the future like "will").

— 9 question words (the LSF equivalent of *where?*, *when?*, *what to do?*, *how?*, *how old?*, *what?*, *why?*, *who?*, *how much/many?*). Those signs are crucial in LSF: in addition to their function as interrogative pronouns, they are used to explain the context of a situation in relative clauses. The sign *where?* can thus introduce the place of the action and the sign *why?* can mean "because" in an affirmative sentence.

— 47 names of animals (e.g., [*AUTRUCHE*] (ostrich), [*CANARD*] (duck), [*BALEINE*] (whale)) and 25 animal descriptors (e.g., [*MAMMIFERE*] (mammal), [*PLUME*] (feather), [*BLEU*] (blue)). Each animal name was performed twice to ensure the presence of two almost identical signs in the resulting data. Having a choice between different instances of the same motion segment is beneficial to add realism in a synthesis context. Animal names present a great range of contextualized hand configurations. Those configurations are therefore performed in different locations in the signing space. Animal names and descriptors can also have recreational applications and can be used, in serious games, to teach the signs corresponding to animals to a French hearing or deaf population.

The third subset consists of 26 **descriptions of animals** in four categories (6 dogs, 5 cats, 11 birds, and 4 mammals with horns). The color, type of skin (fur, plumage, etc.), food preferences and habitat are described for each animal. Then, the animal is identified with a name ([*MOUETTE*] (seagull), [*CHIEN NOIR*] (black dog), [*VACHE*] (cow), etc.). This task was inspired by a LSF lesson for beginners. It provides a natural and authentic flow of LSF utterances. Role shift in which the signer impersonates the character that he/she is talking about (see Section 3.2.2) are numerous in such descriptions as the signer will naturally imitate the animal he is describing. In addition, the resulting production contains various different hand configurations, placements and types of motion.

The fourth subset focuses on three **grammatical mechanisms** of LSF (and of SL in general): size specifiers, pointing gestures and proforms (see Section 3.2.2). The particularity of the first two mechanisms is that only one feature (resp. movement amplitude and hand placement) changes to modify the meaning of the sign/utterance. In the third mech-

anism, two features can be modified: the hand configuration and the hand movement [3]. This property is very interesting for the synthesis of new content by recombination of the phonological components. **1 - Size specifiers** consist in using the standard sign of an object with a different amplitude of movement to accurately represent the size of the object [20], [212]. A big bone, for example, will be signed by doing the sign [*OS*] (bone) with a larger amplitude than for a normal bone. We chose to capture some examples of size specifiers based on the text of a popular fairy tale, namely *Goldilocks and the Three Bears* in which a young girl finds herself interacting with various objects of three different sizes.

2 - Pointing gestures can be used to designate the subject(s) of an action or to associate virtual objects to $3D$ locations in the signing space. Those objects can then be referred to using a pointing gesture on the $3D$ location. To capture various pointing gestures, we designed a task in which the signer had to point with his index to various places on its body and in the signing space. Other types of pointing gestures exist, involving the gaze, the shoulder or torso movements but we limited this study to manual pointing gestures (i.e. index pointings [28]).

3 - Proforms consist in using a particular hand configuration to represent an entity, and a movement of the hand to show the movement performed by the entity. In our case, proforms have two interesting features: they take full advantage of hand configurations and show a wide range of movement trajectories that can be reused in different contexts. We chose to record various utterances in different situations involving vehicles and pedestrians proforms (moving forward, crossing each other, etc.).

Finally, the fifth part was required by our partnership with the Deaf community of our area (*Association des Sourds du Morbihan*). It consists of a **sequence of utterance listing the winners of a theater competition** in the Deaf community. Those sentences are composed of standard signs and no transfer. They have the advantage of being very repetitive and all possess the same global structure with a change in the vocabulary which is precious to study the coarticulation between signs. This part contains sentences in LSF meaning "The winner of the actress category is..." or "The three best theater plays are...", for example.

3. For the three mechanisms, the information conveyed by the face and gaze are also an important part of these structures but they are not the focus of this corpus.

#	Task name	Content	Purposes	Duration per signer
1	Hand configurations	Fingerspelling alphabet + 22 isolated configurations	Training set for automatic annotation of the hand configurations and synthesis of hand configurations.	Signer 1: 2 min Signer 2: 3 min 30
2	Isolated Signs	11 signs with a change in the hand configuration + 9 question words + 47 animal names and 25 descriptors	Analysis of hand configuration transitions, presence of words reusable in various contexts, recreational/educational purposes.	Signer 1: 9 min Signer 2: 5 min
3	Continuous signing	26 descriptions of animals in 4 categories (6 dogs, 5 cats, 11 birds, 4 animals with horns)	Study of transfers of person, of coarticulation and of the impact of the phonological components of LSF.	Signer 1: 20 min Signer 2: 9 min
4	Grammatical mechanisms	Size and shape specifiers + pointing gestures + proforms	Recombination of the phonological components for synthesis purposes.	Signer 2: 10 min
5	Theater competition	Presentation of the winners of a theater competition in different categories	Work for the Deaf community and study of phonological components in LSF utterances	Signer 1: 7 min

Table 6.1 – Content of the *LSF-ANIMAL* corpus.

In addition, we captured various **calibration tasks** as technical aspects like the adjustment of the *Motion Capture* devices and of the *Motion Capture* model used to simplify the post-processing steps require the recording of dedicated gestures. This part also contains some "*range of motion*" tasks : the maximal stretch of one finger with respect to the others to be used in normalization processes, the neck and facial ranges of motion to account for some non manual features. The "*T-pose*" also falls within this part: it eases the binding of the skeleton and the avatar model in the rigging operation, before the animation of the avatar.

6.2 Acquisition of the Data

The capture must follow a strict protocol to collect clean and usable data. Technical considerations, the signers' profile and the elicitation protocol are presented hereafter.

6.2.1 Technical Considerations

In addition to the definition of the corpus, it is necessary to prepare the capture room and to define a marker set in accordance with the task.

6.2.1.1 *Motion Capture* Room

The capture of French Sign Language utterances can be performed in a limited space as the signer does not move during the linguistic production but it also brings important technical constraints [213]: (i) the need to accurately capture gestures with small but meaningful variations (the finger motions particularly), (ii) the temporal dynamics (velocity, acceleration, jerk) must be preserved which requires a high sampling rate and (iii) the whole body is involved in sign language production: facial expressions, gaze, torso motions and manual characteristics must be captured simultaneously.

For the capture, we used a Qualisys environment composed of sixteen infrared cameras (eight OQUS 400 and eight OQUS 700) and one video camera. The two models of infrared cameras do not have the same technical features and it was decided to put the camera with the lowest spacial resolution (OQUS 400) near the signer. The cameras were placed in order to cover the whole signing space and to prevent the cameras from interfering with each other. To preserve the dynamics of the language, the capture was performed at a sampling rate of 200Hz.

6.2.1.2 Marker Set

The choice of the position and size of the optical markers attached to the signer's body are very important to capture the linguistic production with precision. A trade-off must be made between the quality of the capture and the intrusiveness of the equipment. Markers with a large radius will be visualized more easily by the infrared cameras than markers with a smaller radius but will impede the signer's motion and will thus impact the quality of the resulting data. Bigger markers were placed on body locations which are not prone to collide with other body parts. Smaller markers were therefore attached to the fingers and the face.

We used 123 optical markers in total for our capture (see Table 6.2). For the body (without considering the hand and facial markers), the locations described in [214] were chosen. It consists in putting two markers of large radius (12.7 mm) around each joint position (elbows, knees, wrists, ankles) and at other strategic places (sternum, back, feet, etc.) (see Figure 6.1).

Facial markers are paramount to capture the facial expressions which are meaningful in LSF utterances but also, and most of all considering our objectives, to capture the areas where the hand touches the face in some signs. To capture subtle deformations and

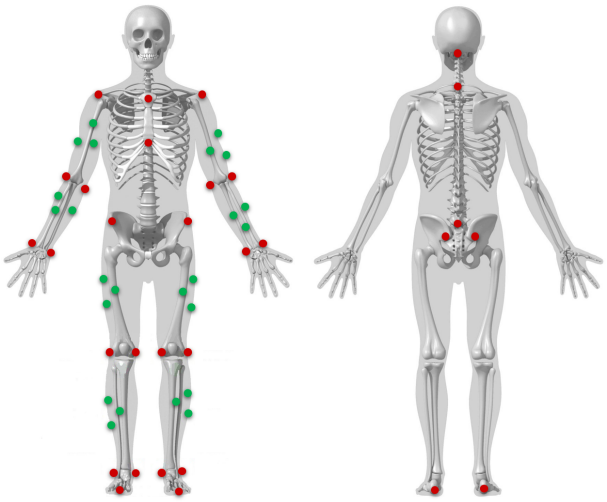

Figure 6.1 – Body marker set : the red markers are used for the definition of the segments and for the tracking whereas the green markers are only used for tracking.

to interfere as little as possible with the signer, those facial markers have a small diameter (4 mm). Given that the hands and their movements are at the center of our study, only 16 markers have been placed on the face; they are a subset of the facial marker set of [215] and form a coarse cartography of the face and its main elements (nose, forehead, mouth, cheek, chin) that serve to indicate the position of the hand with respect to the face.

A more thorough study was performed to determine the location of the hand markers. In order to accurately capture the complexity of the hand motion, it is necessary to use numerous small-sized markers. The performance of reduced hand marker sets (down to six markers on the hand) to produce natural motions were compared in [216]. The authors of the article mainly sought to obtain a realistic motion for simple tasks. However, in addition to realism, sign languages require the avatar motions to be identical or very similar to the source motion. Besides, the location of the markers on the hand is very important to subsequently reconstruct the hand skeleton from the data. The right part of Figure 6.2 shows a skeleton of the hand. A marker was placed on each of the MCP joints, the closer to the bone as possible, and two markers were put on each PIP joints. One marker was added on the extremity of the second and third phalanges. This way, every first and second phalanges are defined by isosceles triangles. It gives an indication of the finger width and direction and simplifies the recognition of phalanges in post-processing work (in particular during the labeling of unidentified trajectories of markers).

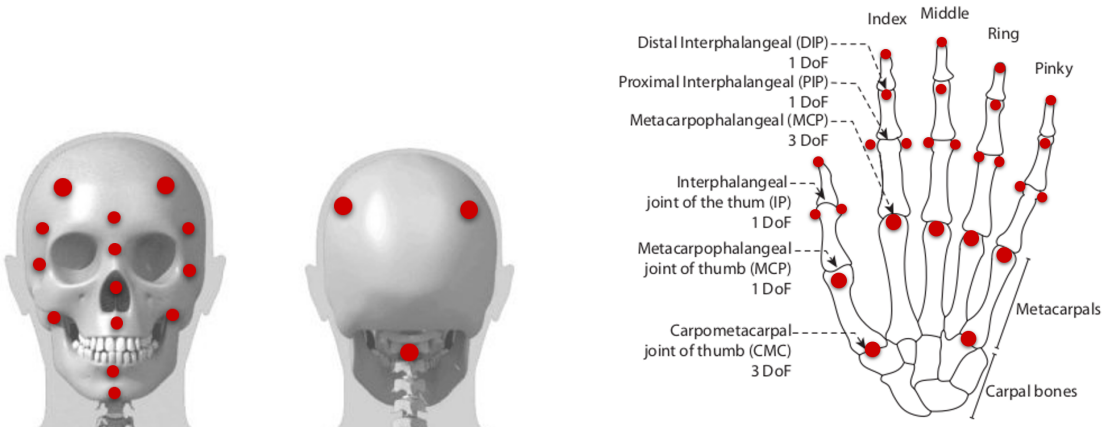

Figure 6.2 – Left: head marker set. Right: skeleton of the right hand and marker set.

Marker set	Nb of markers	Marker diameter
Head	16	12×4 mm 4×12.7 mm
Hands	26 ($\times 2$)	19×4 mm 7×6.5 mm
Body	55	12.7 mm
Total	123	50×4 mm 14×6.5 mm 59×12.7 mm

Table 6.2 – Our marker set: number and diameter of the markers per body part.

6.2.2 Signers and Elicitation

The five subsets of the corpus were captured on two deaf LSF instructors fluent in written French (called Signer 1 and Signer 2 in Table 6.1). Signer 1 was accompanied by an interpreter. The instructions for each task were displayed on a screen in front of the signers. Before each new task, the instructions were clarified in LSF by a member of the lab. The signers knew in advance the global content of the corpus but they discovered the precise tasks during the capture session.

A trade-off was found between precisely controlling the corpus content and giving the signers enough freedom to have the most natural and realistic sign language production. To take this into account, three of the tasks were precisely controlled with instructions written explicitly while the signers were given some leeway in the third part in which he/she has to describe various animals. For this task, we needed the signers to be able to sign in the manner they see fit. We decided to give the signers an image of the animal to be described next to some words underlining the important information that the signers

must provide in their description. No sentences were imposed. The signers could add as many details as they wanted. Two examples of elicitation slides are given in Figure 6.3.

The two capture sessions lasted 4h30 each. We obtained around 1h of raw data in total (≈ 30 min for each signer).

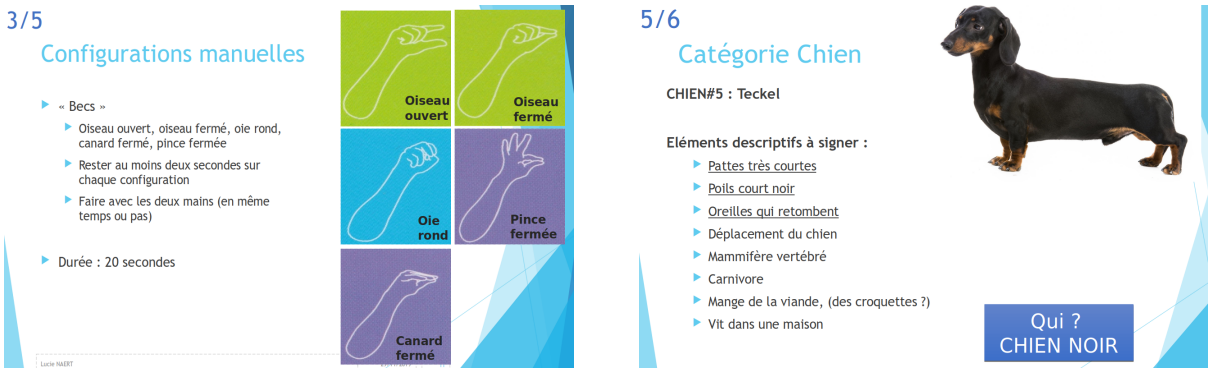

Figure 6.3 – Two projected slides. To the left, an example of a precisely controlled task: the capture of the 'beak' hand configurations. To the right, an example of a less controlled task: the description of the dachshund dog. The image of the animal to be described is shown and descriptive elements to be signed are proposed (the underlined elements are mandatory, the others are optional). The name of the animal is indicated in the blue box.

6.3 Data Post-processing

To obtain the positions and rotations of the joint centers from the positions of the markers, it is necessary to post-process the captured data. To this purpose, the markers are first identified in the capture software. Then, the positions of the markers that were occulted during the capture are reconstructed by interpolation in order to visualize the position of the 123 markers at any time. The transition from the position of the markers to the position and rotations of the joint centers to a motion file in a classical format (bvh, fbx) is performed using the MotionBuilder software [217]. The data thus processed can be directly used to animate a 3D model or can be annotated to perform linguistic analyses and/or motion synthesis operations.

6.3.1 Identification of the Markers

During capture, in real time, Qualisys capture software, QTM [218], tries to assign a marker name to each captured point. This is the automatic identification of markers.

However, some points and trajectories in space are not automatically identified and some identified markers are incorrect (sometimes due to confusions between two markers). A tedious work of correcting and identifying the trajectories manually is therefore necessary (see Figure 6.4). The points captured by the cameras that are not identified are individually tagged. "Ghost" markers (points that appear in the captured sequence but which do not exist in the capture) are deleted.

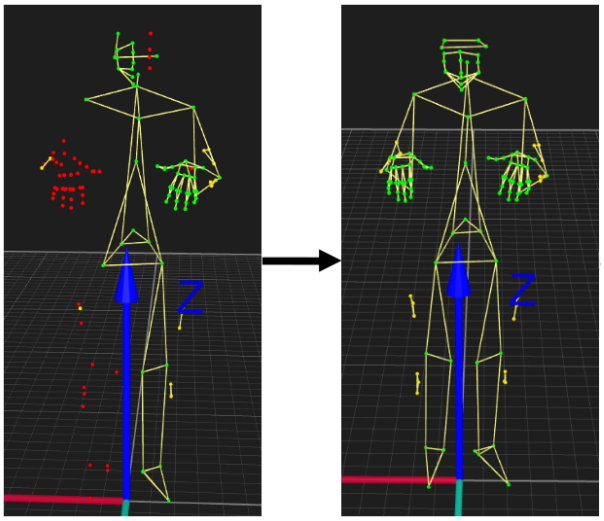

Figure 6.4 – Marker labeling on Qualisys. The red points are unidentified markers whereas the green ones are identified. The two images show the identification of the marker before (left) and after (right) the manual labeling step.

6.3.2 From Raw Data to Standard Motion Format

After having identified the trajectories of the markers captured by the system, the completeness of those trajectories must be checked, i.e. the position of each marker should be captured by the system at each frame. As occlusions often occur during the capture, the missing information must be reconstructed. The missing trajectories are interpolated thanks to the Laplacien operator and the constraints of distances between the markers following the work of [219].

After filling in the missing trajectories, the positions and rotations of the joint centers from the position of the markers remain to be deduced. Indeed, the markers are never positioned at the exact location of the joint because they are put on the epidermis. The MotionBuilder software in the Autodesk suite automates the calculation of the position

and rotations of joint centers to obtain motion files in standard formats (FBX and/or BVH) that can be used for analysis or to animate an avatar.

The BVH format is a format developed by Biovision, a company specialized in *MoCap* data processing. It is a format to represent movement, especially on human skeletons. These files are composed of a first part describing the skeleton, i.e here, a hierarchical set of joints, and a second part to describe the animation via the list of rotations of each joint. The FBX format, on the other hand, is the proprietary format developed by Autodesk. It has the advantage of running directly on Autodesk suite software but, unlike BVH, is not directly readable by human eyes. It works in the same way as the BVH format but can also contain the animated model itself and other animation information such as blendshapes coefficients that allow facial movements.

6.3.3 Playback on an Avatar

From motion files containing an animated skeleton, it is possible to replay the capture sequences on a mesh. The quality of the animation will depend on (i) the quality of the capture data, (ii) the creation of the skeleton that will deform the mesh and (iii) the *rigging* step in which each vertex of the mesh is mapped to one or more bones according to a precise weighting. In the case of replaying captured data, the skeleton corresponding to the data is often different from the skeleton of the model to be animated. In this case, a *retargeting* operation can be performed; the mapping between the bones of the first skeleton and those of the second is provided to the animation system. Retargeting is not recommended if you want to play back the captured data accurately. To avoid retargeting errors and to limit the number of artifacts (inter-penetrations of meshes in particular), it is preferable to animate a mesh as similar as possible to the subject whose movements have been captured. Our choices concerning the animation of avatars are detailed in Section 8.

6.3.4 Data Annotation

The data annotation process consists in associating to each frame of the captured data one or more labels describing the movement performed during this frame. Annotation consists of (i) dividing a continuous stream of movements into smaller segments and (ii) labeling these segments. These tagged segments will then be retrieved to be studied or to animate an avatar. This step can be extremely complex in the case of annotation of signed productions because phonetic, phonological and semantic rules must be taken into

consideration to define the relevant movement segments.

The content of the *LSF-ANIMAL* corpus was manually annotated at a gloss level after the capture, on the ELAN software [71] by visualizing the video stream. The capture data was also used when there was an uncertainty for the precise temporal determination of a segmentation tag. The gloss annotations were then automatically refined [4].

Given that the focus of our work is the study of the different phonological components of sign languages, 12 tracks, corresponding to the configuration of each hand, the placement of the hands, the movement type and orientation were created (more details can be found in Section 7.2).

Hand configurations were annotated automatically using machine learning methods (see Section 7.4) [220]. The placement tracks were also automatically annotated by computing distances between the hands and the body/facial markers (see Section 7.5).

Our corpus is therefore composed of two dependent data sets: the captured motions and the annotations. To manipulate motion segments, the annotation data set can be queried and the corresponding motions are retrieved from the *MoCap* data set [64]. The annotation step is detailed in the chapter 7.

6.4 Perceptual Evaluation of the Corpus

As the initial corpus serves as the core material for the analysis and synthesis of movements, the quality of the synthesis will depend on the quality of the corpus. It is thus necessary to assess whether the signs and motions of the corpus are accurate and realistic. We therefore evaluated the quality of the data present in the corpus using a perceptual evaluation on a subset of this corpus.

To this end, we formulated the following hypotheses :

H_1: The captured data is intelligible.

H_2: The captured data is accurate.

H_3: The captured motions are realistic.

H_4: No information is lost when post-processing the data.

4. See Section 7.3

6.4.1 Design of the Evaluation

To validate or reject our hypotheses, we created videos by varying independent parameters. We then randomly showed the videos to the participants and asked them to recognize the video content and grade the realism and accuracy on a 5-point Likert scale.

6.4.1.1 Evaluated Videos

The *LSF-ANIMAL* corpus is a compound of isolated signs, signs in context and utterances. The physical descriptions of various animals constitute a large part of the corpus. It is difficult and irrelevant to segment these descriptions into discrete signs because, since they are not standard, they would not have a meaning without their context. We therefore decided to evaluate the corpus by presenting two types of sequences: (i) isolated standard signs (animal names) and (ii) whole utterances corresponding to animal descriptions. Participants were asked to find the meaning and to evaluate the accuracy and realism of the sequences.

In addition, in order to be able to validate hypothesis H_4, it was necessary to show different types of data representation at different stages of the processing. Three representations were selected: (A) the points in space linked by segments given by the Qualisys software after identification of the trajectories, (B) the skeleton with the position of the joints calculated by MotionBuilder, and (C) the avatar controlled by the skeleton (see Figure 6.5).

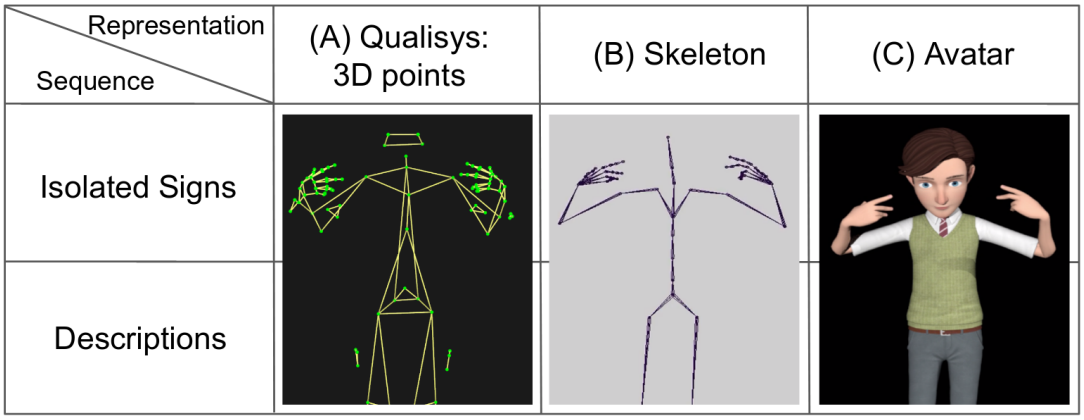

Figure 6.5 – The different combinations of sequences and types of representation.

We chose to evaluate 18 sequences : 9 isolated signs and 9 descriptions (see Table 6.3) multiplied by 3 types of representations (54 videos in total). To avoid overloading the

participants, we separated the evaluation into 3 sub-evaluations of 18 videos. Each sub-evaluation was composed of two parts (Part 1: isolated signs, Part 2: descriptions) and each part contained 9 videos of 9 different sequences. No sequence was evaluated more than once by the same participant, not even with two different representations but they all saw the 3 representations. The videos could be played as many times as the participants wanted. One additional video per part was used as a training session.

#	**Part 1: Isolated signs**	**Part 2: Descriptions**
1	Bird	Labradoodle (dog)
2	Eagle	Dachshund (dog)
3	Duck	Eagle
4	Goose	Rooster
5	Tiger	Grey cat
6	Zebra	Duckling
7	Horse	Persian cat
8	Frog	Red cat
9	Mouse	Black kitten
Training	*Cat*	*St Bernard (dog)*

Table 6.3 – The 18 signs and described animals (plus the 2 training sequences).

The isolated signs were chosen in order to test if small differences in the phonological elements of LSF were visible (see Table 6.4). For example, we chose the signs [*OISEAU*] (bird), [*OIE*] (goose) and [*CANARD*] (duck) which are identical in movement and placement but differ in the hand configuration. The three animals are signed with respect to the appearance of their beak, with the hand of the signer representing the beak. In each case, the hand of the signer is placed in front of the mouth and does an *open/close* movement as if the animal opens and closes its beak. For the bird, the beak is done with the thumb (representing the lower jaw) and index finger (the upper jaw). We named this configuration 'bird_beak'. The goose's beak is performed with the thumb, index and middle fingers (the 'goose_beak' configuration). The duck is done with the thumb for the lower jaw and every other fingers for the upper jaw (the 'duck_beak' configuration). The [*AIGLE*] (eagle) sign is a variant of those signs. An eagle is also represented by its beak but the movement of the hand in this case does not correspond to the *open/close* motion of the beak but is used to describe the appearance of the beak itself.

In a similar way, [*TIGRE*] (tiger) and [*ZEBRE*] (zebra) are identical except for the hand placement (the stripes are represented on the head for [*TIGRE*] and on the torso for [*ZEBRE*]). In the [*SOURIS*] (mouse) and [*GRENOUILLE*] (frog) signs, the non-dominant

hand and arm represent the ground on which the animal (the dominant hand) is moving. The motion of the two animals are different (the frog bounces while the mouse runs) and motion and hand configurations of the dominant hand show this difference. The motion of the two animals are different (the frog bounces while the mouse runs) as well as the hand configuration of the dominant hand that represents the animal shape. [*CHEVAL*] (horse) is unique. It does not belong to a group. It has been chosen to test the accuracy of the sign as it was executed with only one hand by our signer although we usually find it with two hands in dictionaries.

Group	**Signs**	**Element tested**
Beak	Bird, Duck, Goose, Eagle, *(turkey),(hen)*	Hand configuration (Eagle: movement)
Stripes	Tiger, Zebra	Hand placement
OnGround	Mouse, Frog, *(snail), (slug)*	Dominant hand Movement and hand configuration
IsCorrect	Horse	Accuracy of the sign

Table 6.4 – Name and composition of the groups for the isolated signs. The names in italics and between parenthesis were not present in the videos but were proposed as answers to the participants.

6.4.1.2 Questions

Three questions per video were asked to the participants: a question testing the intelligibility of the sign or description (H_1), a question on accuracy (H_2) and a question on the realism of the realization (H_3). All questions and their possible answers were signed in LSF and the resulting videos were subtitled in French.

At first, only the question testing intelligibility was visible. In the case of a sign, the question asked was: "*What sign was made?*" The answer was in the form of a drop-down list of animal names containing about fifty animals, with, at the end, two additional lines: "I did not recognize the animal" and "The animal I recognized is not present in the list" (chance level of 2%). In the case of a description, the question asked was: "*Which image best matches the description that has been made?*". Nine answers were proposed: eight images plus the answer "No image matches the description" (chance level of 11%). The suggestions were close enough so that the answer was not obvious (see Figure 6.6).

When the participant had validated his/her answer to this first question, the correct answer appeared (e.g., "It was the sign DOG") and the following questions were made visible. The second question concerned the accuracy and precision of the sign or description: "*Do you think that the sign/description was done correctly?*". The third question evalu-

ated the naturalness of the movement: "*Do you think that the sign/description in LSF is natural/realistic/spontaneous (does it seem to be the movement of a real person)?*". In both cases, possible responses were presented on a Likert scale ranging from 1 (most negative) to 5 (most positive) (see Figure 6.7).

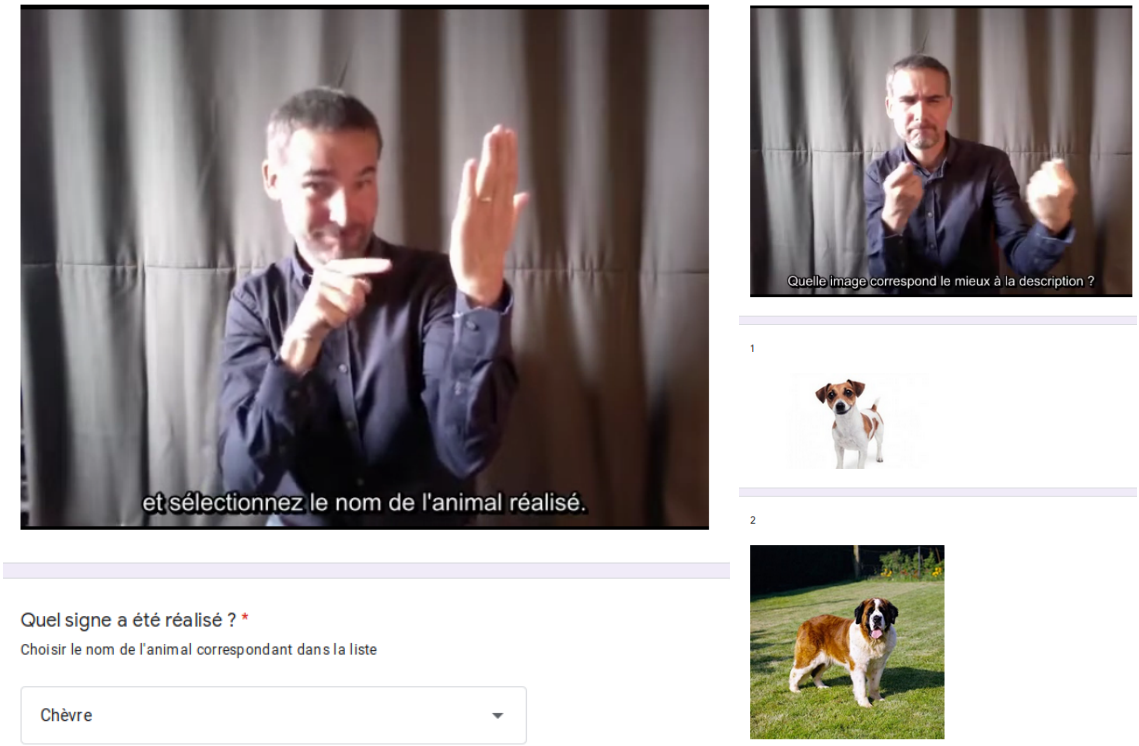

Figure 6.6 – First question on the intelligibility in the case of isolated signs (left) or descriptions (right).

6.4.2 Results

We released the questionnaire online and collected the results from 50 participants, 16 men and 34 women with an average age of 37.64 years old (+/- 14, 7 years, min = 18, max = 72). Among the participants, 31 were hearing people *("entendant")*, 3 were hearing-impaired *("malentendant")*, 2 had become deaf during his/her lifetime *("devenu sourd")* and 14 were deaf since birth *("sourd de naissance")*. In addition, the participants were asked to assess their level of French Sign Language (*no knowledge of LSF*: 9 participants,

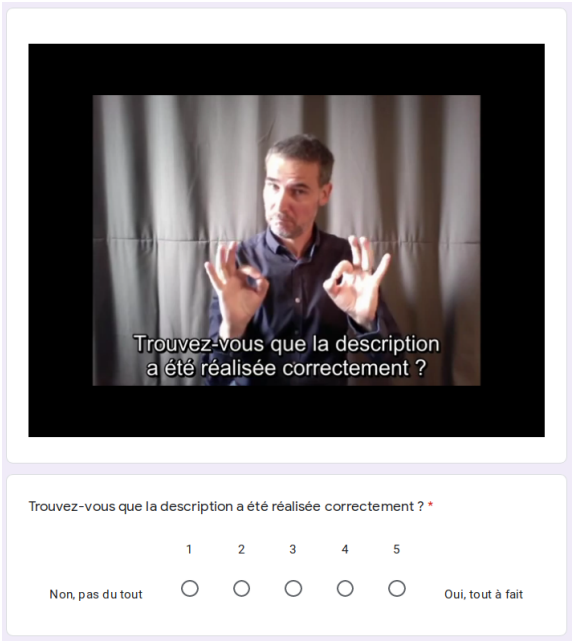

Figure 6.7 – Questions 2 and 3 on the accuracy and the realism of the sequence. Questions are in LSF and in written French.

beginner: 13, *quite good*: 6, *good*: 5, *very good*: 8 or *native*: 9) [5]. We thought interesting to keep the recognition results of the participants with *no knowledge of LSF* but discarded their evaluation of the accuracy of the signs as they were not legitimate to judge this aspect.

6.4.2.1 Recognition Rate

The recognition rate with respect to the level of French Sign Language of the participants [6] is shown on Figure 6.8. In a logical way, the *very good* and *native* signers achieve a better recognition rate for **isolated sign recognition** (blue columns on the figure) than participants with a lower level of LSF. The recognition rate of the 9 participants with *no knowledge of LSF* is above chance level (recognition rate of 24% for the isolated signs) which shows that some of the chosen signs are highly iconic. Still, their recognition rate of isolated signs is lower than the recognition rate of the participants with a higher level of LSF (significant difference with a *p-value* of $1.63e{-}3$ with the *beginners* and $1.78e^{-4}$ with

5. This assessment could be improved by referring to the qualifications listed in a standard language acquisition scale like the Common European Framework of Reference for Languages (CEFR).

6. The level of LSF used is the one specified by the participants in the questionnaire. Among the five *deaf since birth*, only three indicated a *native* level of LSF.

the *native* with the unilateral Mann-Whitney tests). This and the significant difference between the recognition rate of isolated signs of the *beginners* and the *natives* (*p-value* of $6.65e^{-5}$ with the unilateral Mann-Whitney test) show the non-triviality of the task.

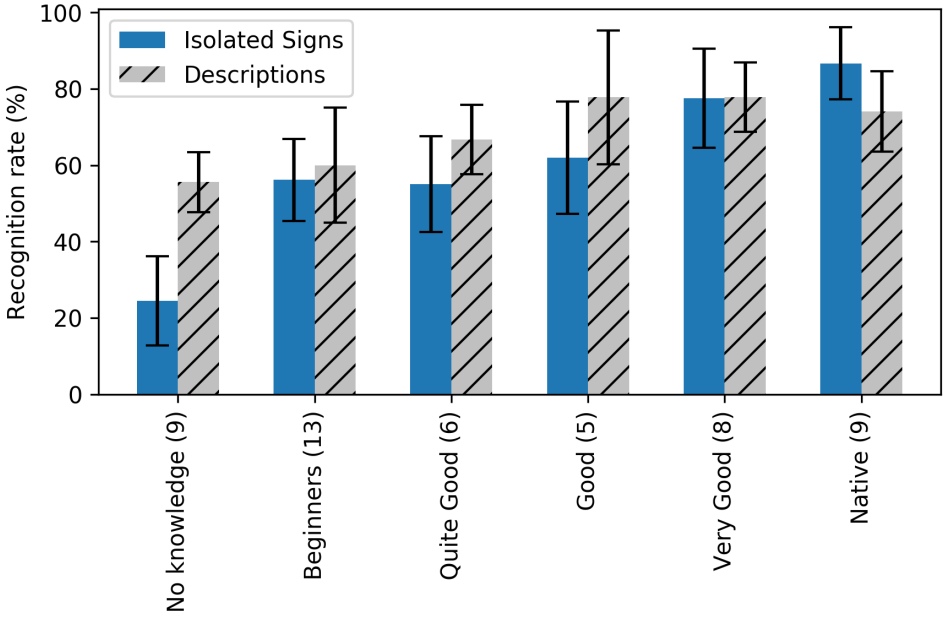

Figure 6.8 – Recognition rate of isolated signs (blue bars) and descriptions (hatched bars) with respect to the level of LSF of the participants (the number of people in each category is specified between parenthesis).

Generally, participants with a *good* level and below have a better recognition rate for **descriptions** (grey hatched columns on the figure) than for isolated signs [7]. The higher recognition rates for the descriptions with respect to the isolated signs for the participants less knowledgeable in LSF can be due to several reasons: (i) a random response on the description part is more likely to be correct as there are fewer possible answers in the description part than in the isolated signs part, (ii) unlike animal names which are isolated standard signs, the descriptions of animals are contextualized and iconic sequences: even people not knowledgeable in LSF can have an idea of the animal described just with the impersonation of the signer (e.g., the behavior of the signer when describing the dachshund's walk or the labradoodle's curly hairs), (iii) participants can learn signs as

7. This is particularly true for the participants with *no knowledge of LSF* who achieve a recognition rate of 55% for descriptions and of 24% for isolated signs (significant difference with a *p-value* of $2.9e^{-3}$ with the unilateral Mann-Whitney test).

they watch the videos: the vocabulary to describe the type of fur, for example, is repeated on different descriptions while, in the case of signs, learning was useless because no two signs were identical. In the remainder of the chapter, for greater clarity, the results of the 9 participants with *no knowledge of LSF* have been discarded, leaving the results of 41 participants.

Isolated signs and descriptions were correctly identified with an average recognition rate of 76.068% by the 22 *good*, *very good* and *native* LSF signers. Figure 6.9 shows their recognition rate per sequence. We can see that 9 out of 18 sequences have a recognition rate higher than 80% even though we intentionally chose similar signs (e.g., *bird*, *duck*, *eagle* and *goose*).

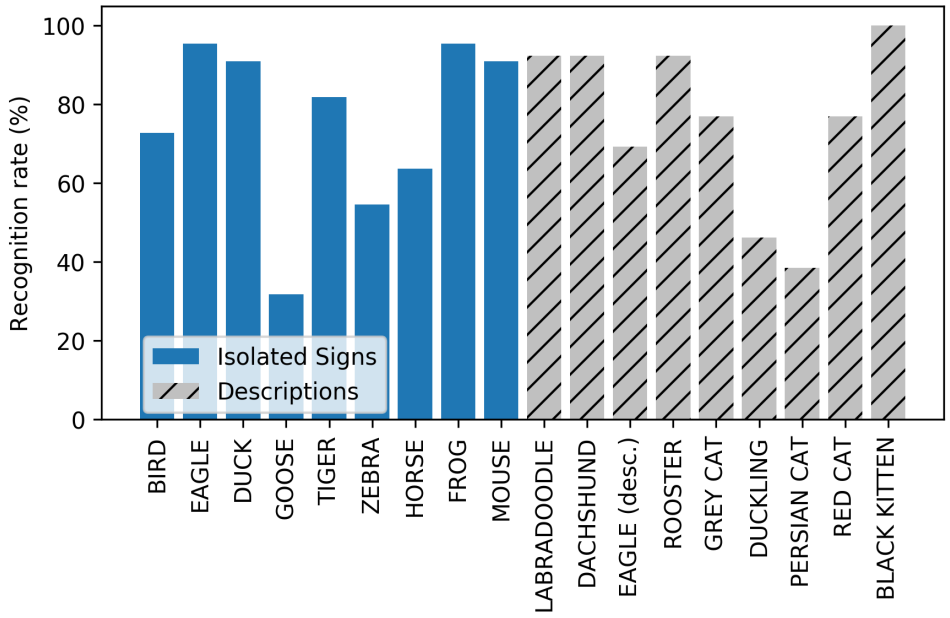

Figure 6.9 – Recognition rate per sequence for the *good*, *very good* and *native* LSF signers.

Regarding the **isolated signs**, [*OIE*] (goose) has the lowest results. On the confusion matrices of Figure 6.10, one can see that 23% of the *good*, *very good* and *native* LSF signers mistook *goose* for *turkey* which shows the same hand configuration at a slightly different location (on the mouth for the *goose* and the nose for the *turkey*). The participants who replied *turkey* had the avatar representation, the only representation with the head visible: the placement may be slightly off on the avatar. This confusion was not made by the other participants who mistook *goose* for *bird* or *hen* whose signs are done at the same placement. The confusion matrices of Figure 6.10 also show the confusions inside the

groups defined in Table 6.4. The *Beak* group, testing the ability to discriminate between hand configurations, contains some confusions that could indicate small inaccuracies in the finger animations. However, there are few confusions between the two members of the *Stripes* group: the hand placement characteristic is well distinguished. Still, the *zebra* sign was not well recognized by the participants with a *beginner* or *quite good* level of LSF (out of the 19 *beginner* and *quite good* participants, 12 answered "I did not recognize the animal"). In practice, *zebra* can be done in two different ways: with the sign for the stripes on the torso preceded by the sign for horse or just with the stripes on the torso. We chose the second possibility so that it could be compared with *tiger*. As a consequence, its meaning was harder to guess for participants with a lower level of LSF. For the *OnGround* group, there was no confusion between *frog* and *mouse*. The hand movement was perfectly distinguished. Yet, there were some confusions between *mouse*, *snail*, *slug* and *snake* which are four signs that present the way of moving of the corresponding animals (with respect to a ground presented by the non-dominant hand for the first three of them). Those confusions almost disappear for the participants who have a *good* and above level of LSF. Finally, the sign *horse* was not often recognized. Indeed, *horse* is usually done with both hands but, in our database, it is done with only one hand which can explain the poor results.

21 participants with knowledge of LSF answered the second part of the questionnaire about **the descriptions**, including 13 *good*, *very good* and *native* participants. Figure 6.11 gives the details of the confusions for the description part. In this part, *persian cat* and *duckling* were not associated with the correct picture in a majority of cases. For both of them, the picture that was chosen instead represented an animal with a color specified in the description (an orange cat instead of an orange-eyed persian cat and a brown duck instead of a brown duckling). Figure 6.12 shows the pictures presented to the participants for those two cases while Figure 6.13 shows the pictures presented to the participants for two cases of well-recognized descriptions.

However, participants can make mistakes and it is therefore important to analyze the answers to the following questions concerning the accuracy and the realism of the signs.

6.4.2.2 Accuracy and Realism

We considered that only the *good*, *very good* and *native* LSF signers were legitimate to answer the accuracy and realism questions. So we exclusively took into account their

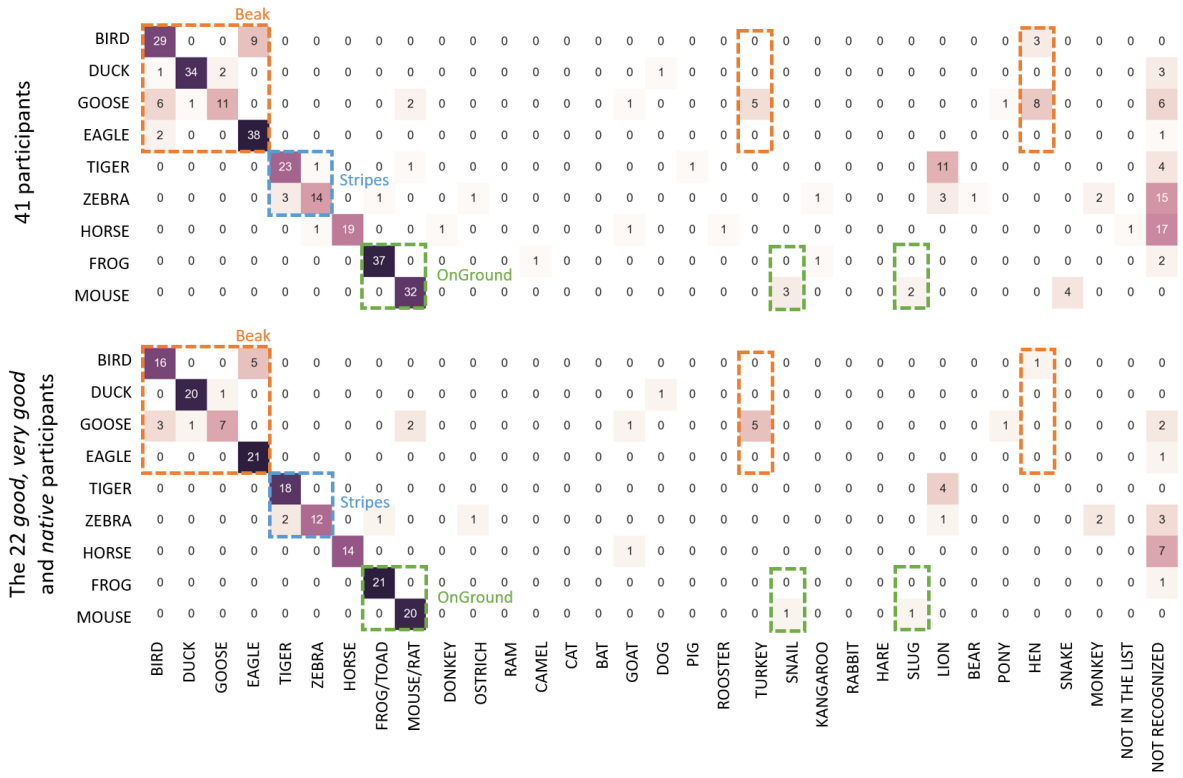

Figure 6.10 – Confusion matrices for the isolated signs (top: the answers of the 41 participants with knowledge of LSF, bottom: the answers of the 22 *good*, *very good* and *native* participants). The categories defined in Table 6.4 are boxed in orange (Beak), blue (Stripes) and green (OnGround).

answers in this section. Table 6.5 shows the mean accuracy and realism scores while Figure 6.14 details the answers of the participants.

	Isolated Signs	**Descriptions**
Accuracy	3.545/5 (0.703)	3.701/5 (0.584)
Realism	3.667/5 (0.398)	3.957/5 (0.425)

Table 6.5 – Accuracy and realism average scores of the *good*, *very good* and *native* participants (22 participants for isolated signs and 13 for descriptions). The standard deviation is specified inside parentheses with respect to the scores per sequence.

With more than 3.5 out of 5, we considered the realism of the movement to be acceptable. The reconstruction of the movement thus provides realistic human motions.

As for the accuracy of the signs, some signs, such as *horse*, were not recognized, not because of a problem in our processing but due to the original movement. As its form is not the standard form of the *horse* sign, it has been discarded from the database. Apart

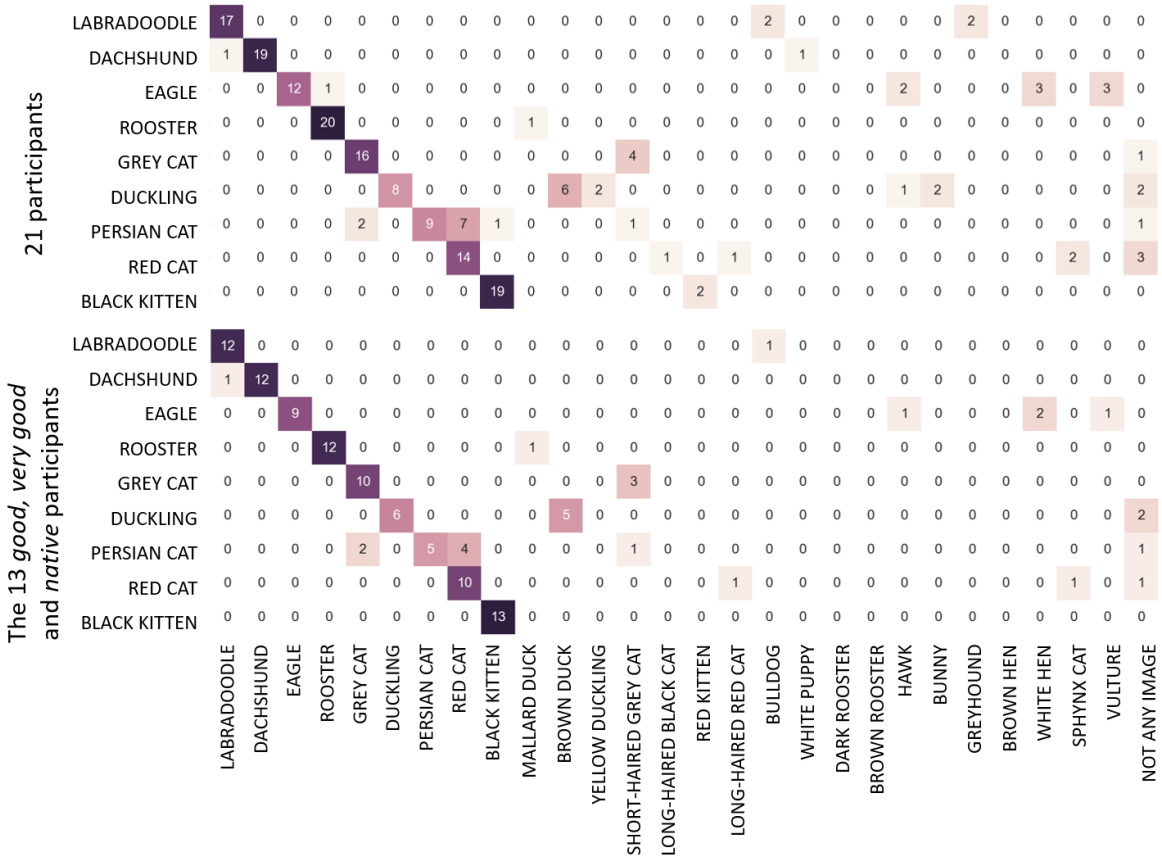

Figure 6.11 – Confusion matrices for the descriptions (top: the answers of the 21 participants with knowledge of LSF having answered the second part, bottom: the answers of the 13 *good*, *very good* and *native* participants having answered the second part).

Figure 6.12 – Two examples of descriptions with a large number of confusions: *duckling* (top) and *persian cat* (bottom). The right answer is in a green box and the number of participants that choose each image is detailed (the number of participants who answered "No image matches the description" is visible in the confusion matrices of Figure 6.11).

Figure 6.13 – Two examples of descriptions with few confusions: *rooster* (top) and *dachshund* (bottom). The right answer is in a green box and the number of participants that choose each image is detailed.

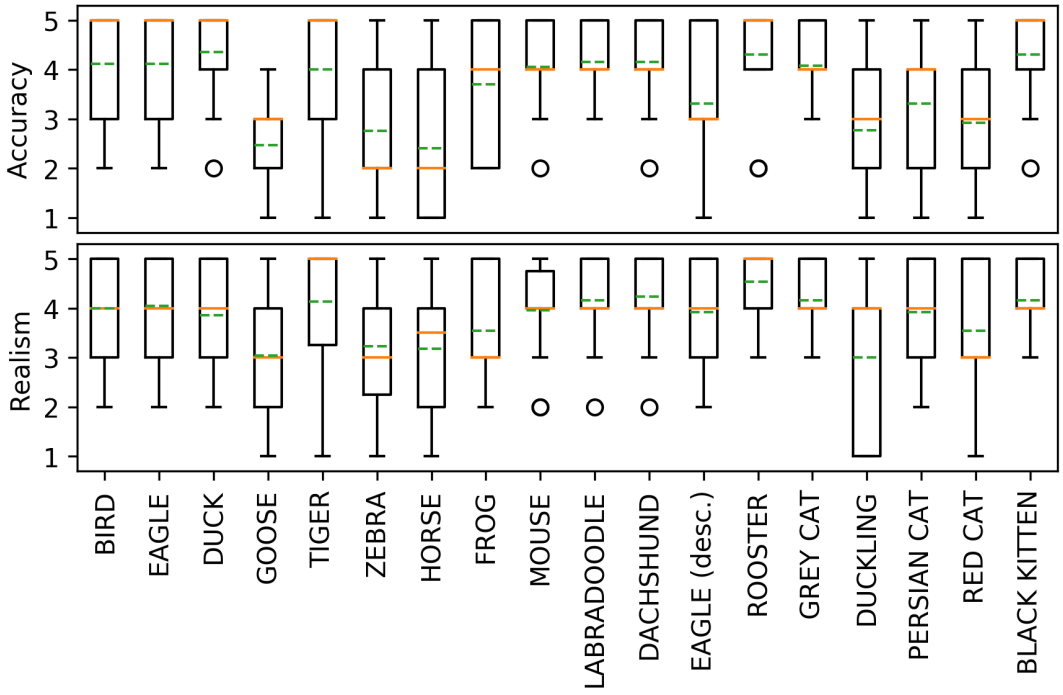

Figure 6.14 – Accuracy and realism score per sequence (the median is the orange line, the mean is the dotted green line, the whiskers go to 1.5 multiplied by the interquartile range). For example, *bird*, *eagle* and *duck* have an accuracy of 5 according to half of the participants.

from those signs, the medians of the results in accuracy are high (equal or above 4) for a majority of sequences (12 out of 18 sequences).

Except for some sequences that should be removed from the corpus, we consider that

H_2 and H_3 are verified which means that the data can be used for synthesis work.

6.4.2.3 Impact of the Type of Representation

To verify the H_4 hypothesis, the accuracy and realism scores were grouped by type of representation (Qualisys, Skeleton or Avatar, see Figure 6.5) for participants with an LSF level greater than or equal to *good*. Each participant rated between 9 and 18 sequences depending on whether or not he/she did the second part (among the 22 participants with a level greater than or equal to *good*, 13 responded to the two parts). We therefore gathered 105 ($22 \times 3 + 13 \times 3$) realism and accuracy ratings per representation (see Figure 6.15).

As the data do not follow a normal distribution, we used the Kruskal-Wallis test for non-parametric data to determine if the type of representation had an impact on the ratings. Whether for accuracy or realism, the results of the statistical test do not allow us to rule out the H_4 hypothesis (for accuracy: $p\text{-}value = 0.10$ and, for realism: $p\text{-}value = 0.65$). We performed unilateral Mann-Whitney tests for the accuracy results and obtained a result that was close to being significant between the skeleton and the avatar representation ($p\text{-}value = 0.022$).

Ideally, we could have benefited from a higher number of results from LSF experts but, as it stands, we can consider that the type of representation had no significant impact on the ratings. The quality of the data was preserved in terms of realism from the raw *MoCap* data (Qualisys representation) to the animated skinned avatar. For the accuracy, there might be a small loss between the skeleton and the avatar representations but the current results do not allow us to reject the H_4 hypothesis.

6.4.2.4 Comments

At the end of the questionnaire, we allowed participants to express their feelings in a free text space. Out of the 41 participants, 9 commented on the lack of facial expressions, 6 expressed their preference for the skinned avatar, justifying it by the presence of placement information of the hands with respect to the face and by the fact that the avatar had a human appearance, 3 said they preferred the skeleton for the precision of the gestures, 3 others preferred the Qualisys representation for the same reason and 3 said they were enthusiastic about the precision and fluidity of the gestures. One participant expressed his surprise in understanding the two avatars without heads.

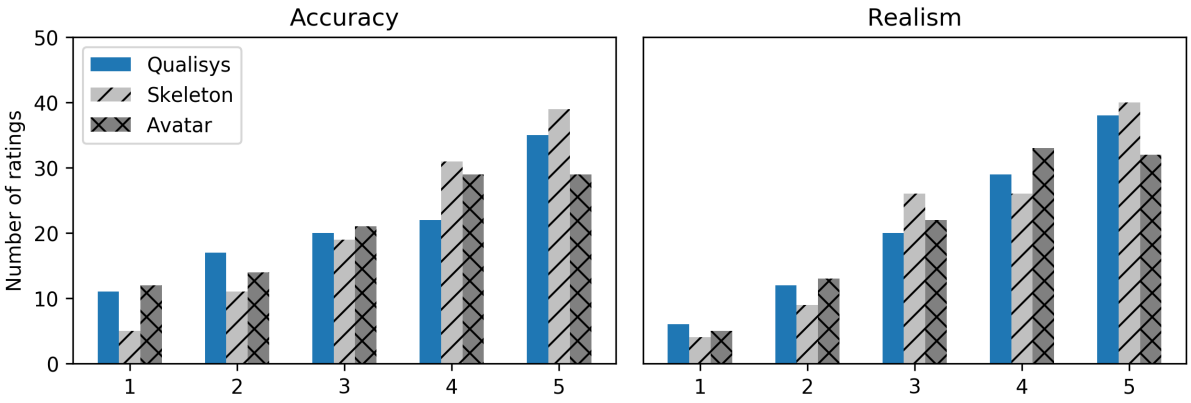

Figure 6.15 – Number of ratings for the accuracy and realism per type of representation for the *good*, *very good* and *native* LSF signers.

6.5 Summary and Discussions

We presented the *LSF-ANIMAL* corpus, a new *MoCap* corpus of French Sign Language for sign language analysis and synthesis applications. The captured data has been post-processed so that it can directly be used to animate virtual signers.

The corpus is composed of five subsets, each of them containing various signs and grammatical mechanisms to meet different synthesis objectives. Some manual phonological components including hand configuration, placement, movement were captured and annotated. Therefore, our corpus, that includes semantically meaningful data, can be used to create new utterances with concatenative synthesis but can also be enriched by combining the different motion segments present in the data set and/or by editing the motion signal in order to create new content not limited to animal names and descriptions. And, given the exhaustive annotation scheme, this corpus can be used for other applications including LSF analysis.

For the capture of our motions, we designed a hand marker set suited to the subtle motions of the fingers when performing sign language signs and utterances.

The captured data was evaluated through one perceptual study involving 50 participants with different LSF levels. We found that the signs were recognized far above chance level and that the participants, except for some specific signs, mainly found the motions of the avatar to be precise, natural and believable when performing signs and utterances. We also found that the post-processing of the *MoCap* data did not impact the quality of the data in a significant way.

Most of the work presented in the rest of this thesis was performed on this corpus.

Chapter 7

Automatic Annotation of Continuous Sign Language MoCap Data

Contents

7.1 Objectives and Challenges 146
7.2 Annotation Scheme of the *LSF-ANIMAL* corpus 149
7.3 Automatic Refinement of the Segmentation of the Gloss Tracks 153
7.4 Automatic Annotation of the Hand Configurations 159
7.5 Automatic Annotation of the Hand Placement 182
7.6 Summary and Discussions 189

> It's supposed to be automatic, but actually you have to push this button.
>
> *Stand on Zanzibar*
> John Brunner, 1968

This chapter has been the subject of three different publications: [52], [220], [221].

To be able to use the captured data of the *LSF-ANIMAL* corpus described in Chapter 6 for analysis or synthesis work, the data needs to be carefully annotated. Indeed, the annotation adds a semantic layer to the raw motions contained in the *Motion Capture* database. This semantic layer can be queried according to linguistic constraints in addition

to the already queriable kinematic or signal constraints. However, the annotation of these data at different levels can be tedious if done manually. In this chapter, we propose to describe and evaluate our automatic annotation techniques to annotate the glosses, hand configurations and hand placement of continuous signing. The remainder of this chapter is organized as follows: Section 7.1 presents the objectives and challenges of the automatic annotation of our captured data. Section 7.2 describes the chosen annotation scheme. The automatic refinement of the gloss tracks of the annotation scheme is detailed in Section 7.3. Section 7.4 and Section 7.5 present the techniques and the results of the automatic annotation of the hand configuration and hand placement tracks respectively. Finally, Section 7.6 sums up and discusses the different results.

7.1 Objectives and Challenges

As mentioned in Section 4.2, the annotation is a three-step process with, first, the definition of an annotation scheme listing the different tracks of the annotation, then, the segmentation of the continuous data stream and, finally, the labeling of the segmented data.

The annotation of sign language data done manually is a tedious and time-consuming task that is subject to imprecisions and errors. As the quality of the subsequent analysis and synthesis of the data is dependent on its annotation, we aim to automatize the annotation process to reduce the annotation time and increase the precision of the resulting annotation.

Moreover, in order to test the generalization properties of our annotation techniques, we wish to annotate two different LSF data sets :

— The *Sign3D* corpus [222] which contains eight sequences of motion. Each of these sequences is composed of one to five French Sign Language utterances. The utterances give information about the opening hours and entrance fees of various town places (swimming pool, museum, etc.), as well as the description of various events (exhibitions, theater play, etc.). The capture was performed on one signer using a *Vicon MoCap* system [223] and an eye-tracking device to follow gaze direction. Facial expressions, body and finger motions were simultaneously recorded during approximately 9 minutes at 100 fps (around 54000 frames in total).

— The *LSF-ANIMAL* corpus whose content and capture are detailed in Chapter 6. It contains approximately one hour of data with isolated hand configurations, iso-

lated signs, descriptions of animals and grammatical mechanisms recorded on two different signers. It was captured with a *Qualisys MoCap* system [224]. Coarse facial expressions and precise body and finger motions were recorded simultaneously at 200 fps.

Our objective is therefore the implementation of generic annotation techniques for *MoCap* SL data to reduce the annotation time and inaccuracies of the manual process.

7.1.1 Terminology

The following is a list of terminology used in this section:

Segment	Element defined by its **temporal limits** (the time tag or frame number of its beginning and of its end).
Segmentation	Action of finding the segments in a continuous stream of data.
Label	Value of a segment
Labeling	Action of assigning a value to a segment
Annotation	The segments and their labels or the action of segmenting and labeling.
Annotation track	The annotations of a category (glosses, right hand configurations). On an annotation tool like ELAN [71], a track is represented by a horizontal line in which segments and labels can be determined, called a *tier* in ELAN. Many annotation tracks can be defined and annotated in parallel.
Vocabulary	Set of possible values for a label. It can be limited (closed vocabulary) or not limited (open vocabulary).
Label format	The formalization rules to write the labels (upper case, use of brackets, etc.).

7.1.2 Challenges

The annotation of human motion in general is a complex problem which was detailed in Section 4.2. However, some challenges are specific to our field of application: the annotation of SL *MoCap* data.

Multi-tracks annotation: sign languages take advantage of the capabilities of the whole body (face, hands, torso, gaze...) to convey a message. Depending on the final application, the annotation can be performed at different granularities providing information on syntactic, lexical or phonological levels. The annotation of SL data therefore requires the definition of a precise and often complex annotation scheme with multiple tracks taking values in parallel. Section 7.2 describes our annotation scheme which focuses on the gloss and phonological levels. The challenges specific to each annotation track are detailed in the section dedicated to the corresponding track (Section 7.3 for the gloss track, Section 7.4 for the hand configuration track and Section 7.5 for the hand placement track).

Segmentation of continuous signing: the two data sets that we decided to annotate contain full utterances and, therefore, continuous signing. Continuous signing is problematic for the segmentation at a gloss level for three main reasons. First, the beginning and end of a sign are harder to place precisely as the signs quickly follow one another. Secondly, the signs are inflected by grammatical and coarticulation mechanisms and are not always in their form of citation which complicates recognition tasks. Finally, the transitions between signs, sometimes called "movement epenthesis" [107], [113], are shorter and contain more variations than in a sequence of isolated signs, making them harder to determine. However, these transitions are very important in the case of continuous signing because they constitute natural articulations between signs that could be studied in order to be resynthesized to build correct utterances while transitions in sequences of isolated signs do not contain interesting attributes for synthesis. Finally, the segmentation done manually must be rigorous, particularly for synthesis purposes, as the segments with the same label should be commutable.

Temporal variations: one sign performed by the same person at two different times can differ because of various reasons: the level of fatigue of the signer or the context of the sign may be different. The same is true for the phonological components: the same hand configuration can be done slightly differently between two different takes. These temporal variations must be managed by annotation algorithms.

Inter-personal variations: the data that we choose to annotate was recorded on three different signers (one woman for *Sign3D* and two men for *LSF-ANIMAL*). The different morphologies of the signers – their size, the size of their hands or their fingers, the

maximum spacing between their fingers, etc. – have an impact on their signing. Therefore, it is interesting and challenging to apply annotation algorithms on multiple data sets in order to test the robustness of the algorithms against inter-personal variations.

7.2 Annotation Scheme of the *LSF-ANIMAL* corpus

The annotation scheme gives the structure of the annotation by indicating the different annotation tracks, the vocabulary used to label each track and the format of the labels. The annotation scheme depends on the final application: annotations, in our case, are used to index the data for further processing by an analysis/synthesis tool in order to perform specific synthesis tasks. For the synthesis of signs by recombination of the phonological components with different values, we need an annotation at a phonological level. For the synthesis of utterances with inflected signs, we require a gloss-level annotation. Our annotation scheme must therefore possess multiple tracks. Before presenting those tracks, their vocabulary and format, we briefly justify the choice of our annotation tool.

7.2.1 Annotation Tool

The annotation task requires a tool to manually annotate the data and to visualize those annotations. Ideally, this tool should be able to view video recordings of the data (videos obtained from the RGB camera during capture), to manage different annotation tracks in parallel, to determine the beginning and end of time segments on the annotation tracks, and to label each of these segments according to a vocabulary that can be chosen in advance. We selected the ELAN tool [71] which meets all the requirements and is regularly used for sign language annotation.

7.2.2 Annotation Scheme

For the annotation of the *LSF-ANIMAL* database, we chose to focus on the manual phonological components. Thus, the channels of the hand configuration, hand placement, hand movement and, to a lesser extent, hand orientation are present while the non-manual aspects (e.g., facial expression, torso movement, gaze orientation) are left out. Gloss tracks were added to the phonological tracks to allow indexation at a higher level. There is no notion of hierarchy or dependency between these tracks so that they can be annotated completely independently. The 18 chosen annotation tracks are detailed hereafter.

7.2.2.1 Gloss Annotation

Glosses are annotated in French on several tracks. Glossing consists in associating one or more words of an oral language to a sign to describe it.

It is important that all the variants of the same sign always have the same gloss with an identical spelling and identical case. Table 7.1 lists the formatting rules defined with this purpose in mind.

Category	Rule	Example
Brackets	The glosses are put between square brackets	[DOG]
Case	To differentiate the glosses from a translation, we chose to write them with uppercase letters	[HOUSE]
Accent and spacing	No accent must be written on the letters and the space between words must be replaced by an underscore	[*ELEPHANT_GRIS*] ([ELEPHANT_GREY]) instead of [*ÉLÉPHANT GRIS*] or [*IL_Y_A*] ([THERE_IS]) instead of [*IL Y A*]
Singular/plural	When there is a doubt on the plural/singular form of a sign, the singular form must be preferred	DOG instead of DOGS
Adjective ending	In French, the adjective ending changes with the corresponding noun. For glossing, the adjective must be in its masculine/singular form	[*LONG*] instead of [*LONGUE*] (the feminine form of "long")
Adjective	For descriptions, some signs correspond to both the noun and an adjective (for example, a unique sign can be performed to signify "a crushed nose"). Those signs are annotated with, first, the noun and then the corresponding adjective separated by an underscore	[*NEZ_ECRASE*] (NOSE_CRUSHED)
Verb Conjugation	The verbs are in the infinitive form	[*MANGER*] in French or [TO_EAT] in English
Interrogative signs	If the sign corresponds to a question word, the gloss incorporates the question mark	[*QUI?*] ([WHO?]), [*QUAND?*] ([WHEN?])

Table 7.1 – The formatting rules for the gloss tracks.

Four tracks were defined for the glosses:

— **Two handed gloss** (*Glose_2M*): annotation of glosses for signs involving both hands (ex: [*MAISON*] (house) in LSF is done with both hands performing a symmetrical motion).

— **Right hand gloss** (*Glose_MD*): annotation of glosses for signs made exclusively with the right hand (e.g., for the sign [*ROUGE*] (red), the dominant hand is placed at the mouth level). Almost all the signs made with one hand are made with the dominant hand which is often the right hand for right-handed persons. The right hand is the dominant hand of both our signers.

— **Left hand gloss** (*Glose_MG*): annotation of the glosses for signs made with the left hand (e.g., [*SOL*] (ground) in the sign [*GRENOUILLE*] (frog) or [*BOL*] (bowl) in the sign [*CROQUETTE*] (pet food) in LSF). The left hand is often the non-dominant hand which can be used to preserve or place the context. It is rare that a sign is made exclusively with the non-dominant hand.

— **Complementary gloss** (*Glose_complementaire*): higher level gloss. For example, the complementary gloss [*VERTEBRE*] (vertebrate) will encompass the two handed glosses [*COLONNE_VERTEBRALE*] (spine) and [*LA*] (there). In the same way, [*BEBE*] (baby) and [*CHIEN*] (dog) will give [*CHIOT*] (puppy) as a complementary gloss.

7.2.2.2 Annotation of the Phonological Components

Each of the following tracks is defined separately for the right hand and for the left hand. Table 7.2 lists the 12 annotation tracks corresponding to the phonological components (6 for each hand).

The tracks for the distance, height and radial placement correspond to the hand placement phonological component and are annotated automatically (see details in Section 7.5).

As stated in Section 6.1.2, the closed vocabulary for the hand configurations was defined by comparing 5 different sources: (i) the 32 manual configurations annotated in the *Sign3D* corpus [69], (ii) the 45 configurations presented by an LSF teacher, (iii) the 63 configurations of an LSF textbook [211], (iv) Cuxac's 41 configurations [3] and (v) the 62 configurations of International Visual Theatre book which is a reference for LSF grammar and vocabulary [9]. The configurations that were common to at least 4 of the 5 different sources were chosen. To these 39 configurations, we added the missing letters, as well as the configurations discovered during the annotation (e.g., *S_pouce_interieur*) to obtain 48 configurations in total.

French Track Names	**Definition**	**Vocabulary**
Configuration_MD *Configuration_MG*	Configuration of the hands. The hand configuration segments correspond to the time span during which the hand configuration is stable even if the rest of the body moves.	Closed, 48 labels: {*3, 3_plie, 4, 4_plie, 5, 5_plie, A, B, C, C_large, D, E, F, G, H, I, K, L, M, O, R, S, S_pouce_interieur, T, U, V, W, X, Y, angle_droit, angle_droit_index, ballon, bec_canard_ferme, bec_oie_rond, bec_oiseau_ferme, bec_oiseau_ouvert, clef, index, lit, main_plate, majeur, moufle, oui, personne, pince_fermee, pince_ouverte, u_pouce, wc*}
Mouvement_MD *Mouvement_MG*	Hands trajectory.	To be defined
Orientation_MD *Orientation_MG*	Wrist orientation.	To be defined
Emplacement_Distance_MD *Emplacement_Distance_MG*	Distance between the body of the signer and the hand with respect to the size of the signer's arm.	Closed, 4 labels: {*Touch*, *Close*, *Normal*, *Far*}
Emplacement_Hauteur_MD *Emplacement_Hauteur_MG*	Height of the hand of the signer in the signing space.	Closed, 6 labels: {*BellowBelt*, *Abdomen*, *Chest*, *Neck*, *Head*, *AboveHead*}
Emplacement_Radial_MD *Emplacement_Radial_MG*	The radial orientation of the hand with respect to the body of the signer.	Closed, 5 labels: {*Front*, *Left*, *Right*, *BehindLeft*, *BehindRight*}

Table 7.2 – The 12 annotation tracks of the phonological components

7.2.2.3 Other Annotation Tracks

Two other annotation tracks were added to the annotation scheme.

The first track, called *Phrase*, corresponds to the translation of some expressions (e.g., "floppy ears", "white hair") or utterances (e.g., "who am I?"), or to the description of movement without meaning (in this case, it will be marked in square brackets, e.g., "[T pose]"). The *Phrase* track is annotated using lowercase, as opposed to the uppercase gloss tracks.

The second track is called *Symetrie_Mains*. It describes the symmetry characteristics of the two hands with respect to movement, hand configuration and orientation. If symmetry is indicated, it is not necessary to annotate the two phonological characteristics of the two tracks (right and left hands), although, in practice, we annotate them anyway to visualize the difference between intention (symmetry) and the production (slight shift from one hand to the other, advance/delay of one hand...). Symmetry can be planar with

respect to a horizontal (e.g., [*CROCODILE*]) or vertical (e.g., [*CHAT*] (cat)) plane. We also defined a relationship called "Alternance" between the two hands. In this case, the two hands make the same movement with the same configuration but with a temporal shift of one hand with respect to the other (e.g., [*THEATRE*]). Those three examples are shown on Figure 7.1. The *Symetrie_Mains* track can thus be annotated with a closed vocabulary of 3 labels: {*PV_Plan_Vertical*, *PH_Plan_Horizontal*, *Alternance*}

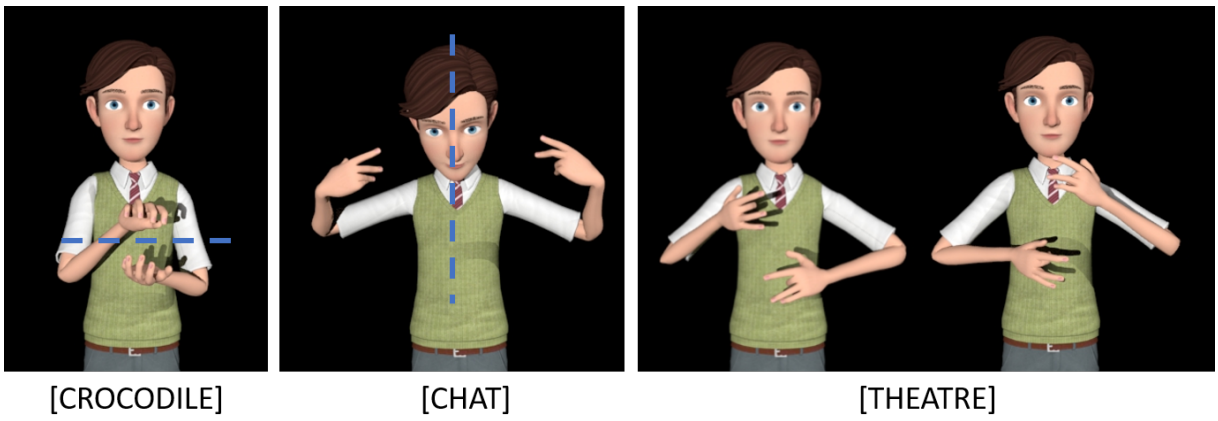

Figure 7.1 – From left to right: [*CROCODILE*] with a horizontal symmetry axis, [*CHAT*] (cat) with a vertical symmetry axis and [*THEATRE*] with a temporal shift of the two hands.

Among the 18 presented tracks of the annotation scheme, the manual annotation of 3 tracks has been refined using automatic means, 2 tracks have been annotated automatically based on a supervised machine learning algorithm and 6 have been automatically annotated using the morphological characteristics of the signers. The other tracks have been annotated manually. Appendix A sums up the 18 annotation tracks and their format.

7.3 Automatic Refinement of the Segmentation of the Gloss Tracks

In this section, we propose an automatic method to refine the segmentation done manually by human annotators at a gloss level. Three annotation tracks are impacted by this refinement: **Glose_MD**, **Glose_MG** and **Glose_2M**.

7.3.1 Motivations

In sign languages, the signer alternates between signs and transitions. When annotating recordings of sign language utterances at a gloss level, the first step consists in manually isolating signs from inter-sign movements which constitutes the transitions from one sign to the next. During the *Motion Capture* session, in addition to the infrared cameras, an RGB video camera records the performance of the actor. The video produced by the RGB camera is more easily understandable by human annotators and it is therefore taken as reference for the manual segmentation. As a consequence, manual segmentation results in imprecisions since *MoCap* data is sampled at a frequency of 100 to 200Hz but the RGB video used as reference for the annotators is sampled at 24Hz. Video annotation is thus 4 to 8 times less accurate than what *MoCap* data would ideally allow. Another drawback of manual segmentation is that it is a subjective task that greatly depends on the annotator's criteria and on the quality of the data to be processed. It is subject to variations between annotators and even between different annotations made by the same annotator. Even with clear and precise instructions, it is often difficult to point out with certainty the beginning and end of a sign especially since the effects of coarticulation and inflection amplify the smoothness of the signing.

The quality of the data-driven synthesis depends on the data and on its annotation. We therefore propose to refine the manual segmentation of the glosses using automatic means.

7.3.2 Method

To develop an automatic refinement of the manual segmentation, we first analyzed the kinematic properties of the *MoCap* data present in the *Sign3D* and *LSF-ANIMAL* databases. Then, we used those observations to implement our refinement method.

7.3.2.1 Preliminary Study of the Kinematic Features of LSF Motions

In order to study a possible correlation between the kinematic properties of the hand motions and the sign/transition segmentation, utterances of LSF composed of a few signs separated by transitions were examined. Those sequences, considered as the ground truth, are raw motions directly extracted from the *LSF-ANIMAL* database. We have computed several kinematic features on those motions for various joints and observed that the norm of the velocity of both wrists had interesting properties.

Figure 7.2 shows the norm of the velocity of both hands (left wrist and right wrist) for one sequence which can be translated as "It has short red hairs. It walks like a cat.". The black vertical lines mark the beginning and end of a gloss according to the manual segmentation. The vertical blue and orange lines show the local minima of the two curves, respectively depicting the norm of the velocity of the left hand and of the right hand. The values were processed using a lowpass filter to prevent the algorithm from detecting all the incidental minima due to noise in the data.

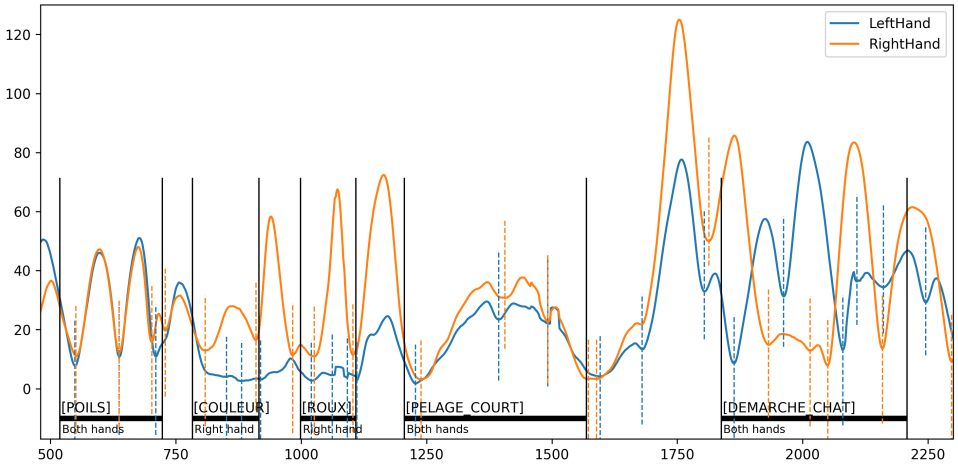

Figure 7.2 – Norm of the velocity of the left (blue line) and right (orange line) wrists with respect to the frame number for the sequence of LSF glosses [HAIR] [COLOR] [RED] [SHORT_HAIR] [CAT_WALK]. The hand(s) that perform the sign is/are indicated under the gloss tag. The vertical lines show the edge of the transition defined by the annotator (black lines), the local minima of the left hand (blue lines) and the local minima of the right hand (orange lines). We can see that the minima are not reached at the same frame for both hands.

According to the manual segmentation, the transitions seem to be delimited by two local minima in the norm of the velocity even though, due to the manual aspect of the task, the tags are not positioned exactly on the minima. Our hypothesis is thus that the correct segmentation of the signs should be on the local minima of the curves.

Furthermore, when doing the manual segmentation, the whole body is considered. So, the starting and ending time of a sign are considered to be the same for all of the skeleton joints. However, that is not always true in practice. On Figure 7.2, we can note that there is an offset between the minima for each hand: assuming that minima delimit the beginning/end of a sign and therefore the end/beginning of a transition, the transitions of the left hand do not occur at the same time as the transitions of the right hand.

Therefore, we decided to refine the manual segmentation of *Motion Capture* data of French Sign Language by:

(i) detecting the minima in the norm of the velocity of the wrists, and

(ii) assuming that the joints of the two hands are partially autonomous.

7.3.2.2 Automatic Detection of the Local Minima in the Velocity

We are looking to automatically detect minima in the wrist velocity curve. This detection is done in several steps which are listed below and illustrated on Figure 7.3.

1. Computation of the norm of the velocity for both wrists. The velocity is obtained by deriving the successive positions of the wrists according to an orthonormal coordinate system with the x, y and z axes. Then, the Euclidean norm of the velocity for each wrist is computed as:

$$normVelocity = \sqrt{{velocity_x}^2 + {velocity_y}^2 + {velocity_z}^2} \tag{7.1}$$

2. Filtering to smooth the norm of the velocity in order to prevent the algorithm from detecting all the incidental minima due to noise in the data. The filtering was done by computing the average of the values contained in a sliding window of size n:

$$smoothVel(i) = \frac{1}{n} \sum_{k=i-\frac{n-1}{2}}^{i+\frac{n-1}{2}} normVelocity(k) \tag{7.2}$$

3. Computation of the derivative of the filtered norm:

$$derivVelocity(i) = \frac{smoothVel(i+1) - smoothVel(i-1)}{2} \tag{7.3}$$

4. Rough detection of the local minima by storing the indexes corresponding to frames where the derivative of the velocity changes from a negative to a positive value.

5. Refinement of the minima by selecting the indexes with the lowest value in the norm of the velocity that are close (fewer than 3 frames) to the indexes of the rough detection. Indeed, the rough detection gives approximate results because of the computation of the derivatives that put a 1 frame offset and because of the possible 1 frame offset due to the detection method.

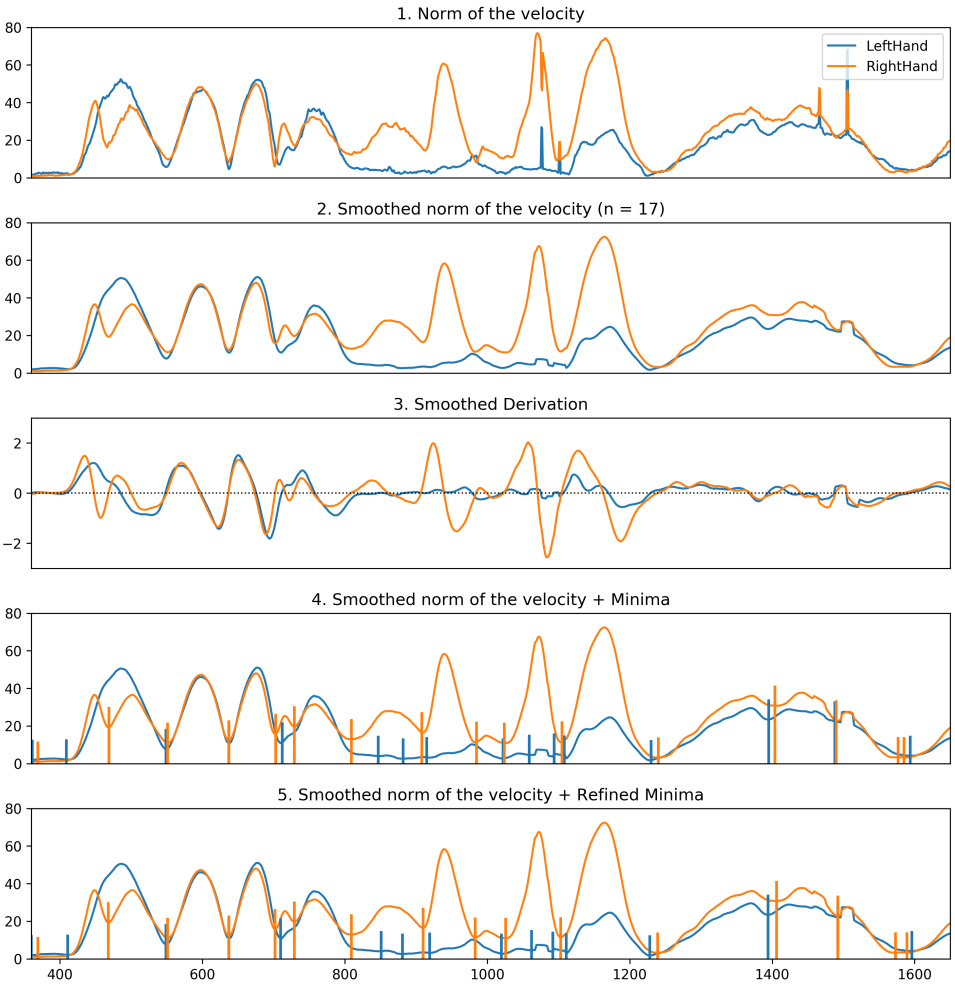

Figure 7.3 – Illustrations of the steps 1 to 5 of the automatic detection of the minima for a motion sequence.

7.3.2.3 Refinement of the Segmentation

The detection of the minima in the norm of the velocity of the wrists is a fully automated process. However, it presents mixed results when used for LSF segmentation because of false positives when a sign is too noisy or complex (like the sign [*POIL*](hairs) in LSF which is done on the arm and then on the torso or the sign [*APRES-MIDI*] (afternoon) which is a contraction of [*APRES*] (after) and [*MIDI*] (noon) in LSF).

We have thus used a combination of the manual annotation and the local minima computation. The manual segmentation tag is replaced by the closest automatically computed minimum in a specific range. If there is no minimum sufficiently close to a manual segmentation tag, it is kept with no modification (see Figure 7.4: the minimum for the

right hand ③ is too far away from the manual tag ④).

To this end, we used a threshold of approximately 4 times the timestep between two frames of the RGB camera. Therefore, a computed minimum distant of more than 4 frames of the RGB camera (i.e. more than $4 \times 1/24 = 167$ ms) from a manual segmentation tag is discarded. In practice, the threshold will be a multiple of the timestep between two consecutive frames of the *MoCap* cameras (e.g., at 200Hz, it will be a multiple of 0.005s).

As the *Motion Capture* data was segmented by people knowledgeable in LSF, we assumed that this relatively small threshold would result in a correct segmentation for a majority of cases. One benefit of this 4-frames threshold is that the minima of complex signs is removed from the segmentation (see Figure 7.4, ①). This segmentation can be considered as an automated refinement of the manual segmentation that has the advantage of providing a different segmentation for each hand (see Figure 7.4, ②).

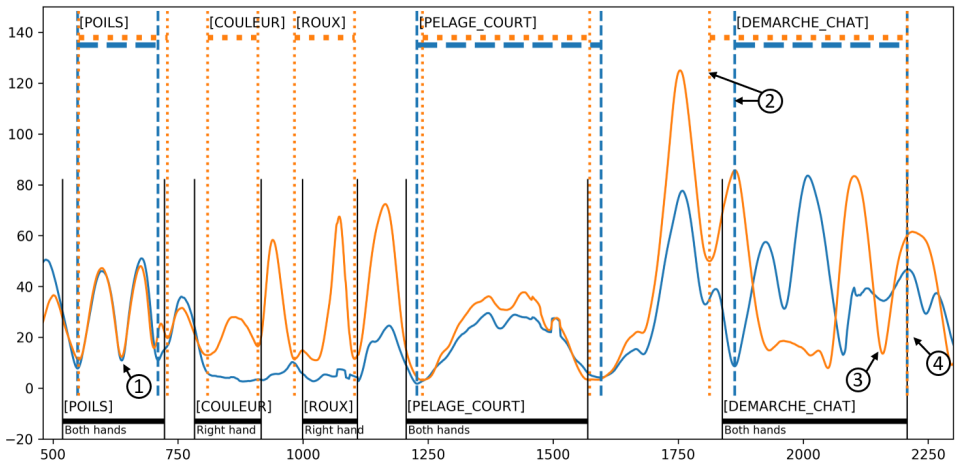

Figure 7.4 – Norm of the velocity of the left (blue line) and right (orange line) hands with respect to the frame number. The vertical lines show the edge of the signs defined by the annotator (black lines) and refined by the segmentation procedure (blue lines for the left hand and orange lines for the right hand). ① : Minimum of a complex sign. ② : Different segmentations for each hand. ③ : Minimum which is too far away from the closest manual tag ④ .

For the refinement of one-handed signs whose manual annotation is done on the **Glose_MD** or the **Glose_MG** tracks, we only detect the minima of the involved hand (right hand if it is a right hand sign, left hand otherwise). The refinement is directly applied on the source tracks (**Glose_MD** or **Glose_MG**). For two-handed signs (manual annotation on the **Glose_2M** track), the minima of the right hand and of the left hand are detected and the refinement is added to the **Glose_MD** and the **Glose_MG** tracks.

The **Glose_2M** is left unchanged unless otherwise specified, in which case the minima for the dominant hand are chosen to refine the track.

7.3.3 Application

This automatic refinement of the segmentation at a gloss level was applied on the *Sign3D* and *LSF-ANIMAL* databases. We used a threshold of $170ms$ (34 frames at $200Hz$ for the *LSF-ANIMAL* database and 12 frames at $100Hz$ for the *Sign3D* corpus). Figure 7.5 shows the result of the automatic segmentation of the gloss tracks with different segmentation thresholds. For this example, the augmentation of the threshold above $170ms$ does not impact the segmentation. The threshold of $170ms$ seems therefore to give the best results as it includes every minima.

Segmentation is a subjective task that is hard to evaluate. As we lacked a ground truth to compute comparative metrics between the manual and automatic segmentation, we did not evaluate quantitatively our method. However, we believe that our segmentation is less biased, more accurate and more homogeneous than the manual segmentation as we use precise quantitative features to segment.

7.4 Automatic Annotation of the Hand Configurations

After having refined the annotation of the gloss tracks, we propose in this section an automatic method to annotate the hand configurations from *MoCap* data in order to alleviate the manual annotation work done by specialists. The two annotation tracks of the hand configurations are impacted: **Configuration_MD** and **Configuration_MG**.

7.4.1 Motivations and Challenges

Hand configurations correspond to the different shapes of the hand done during the execution of signs or utterances. The hand configuration is a relatively stable parameter and many signs contain only one configuration (like [*CHEMINEE*] (chimney) in LSF, see Figure 7.6, left). The hand configuration is independent of the placement or orientation of the hand (see Figure 7.6, right).

The manual annotation of hand configurations is fastidious and can be erroneous

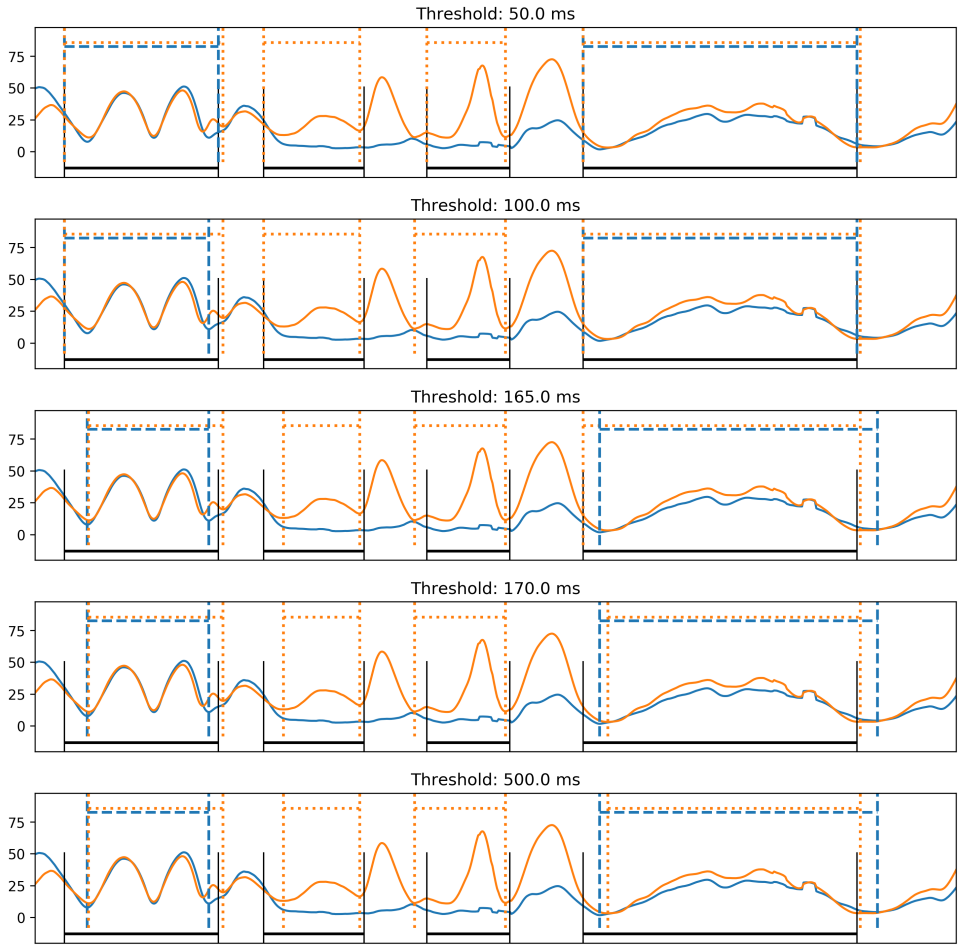

Figure 7.5 – Norm of the velocity of the left (blue line) and right (orange line) hands with respect to the frame number. The vertical lines show the edge of the signs defined by the annotator (black lines) and refined by the segmentation procedure (blue lines for the left hand and orange lines for the right hand). Different segmentation thresholds are tested. For this particular example, the augmentation of the threshold above 170ms does not impact the segmentation.

due to the unique point of view given by RGB cameras used for the annotation and to the different segmentation criteria which are sometimes based more on intuition than on objective values. We decided to develop an automatic technique for hand configuration annotation in order to reduce the annotation duration and to obtain temporal segments with boundaries determined in a less subjective way. Moreover, the automatic annotation of hand configurations, depending on the algorithms applied, can be used to determine equivalence or similarities between the configurations that can be reused in synthesis for an LSF teaching app for example.

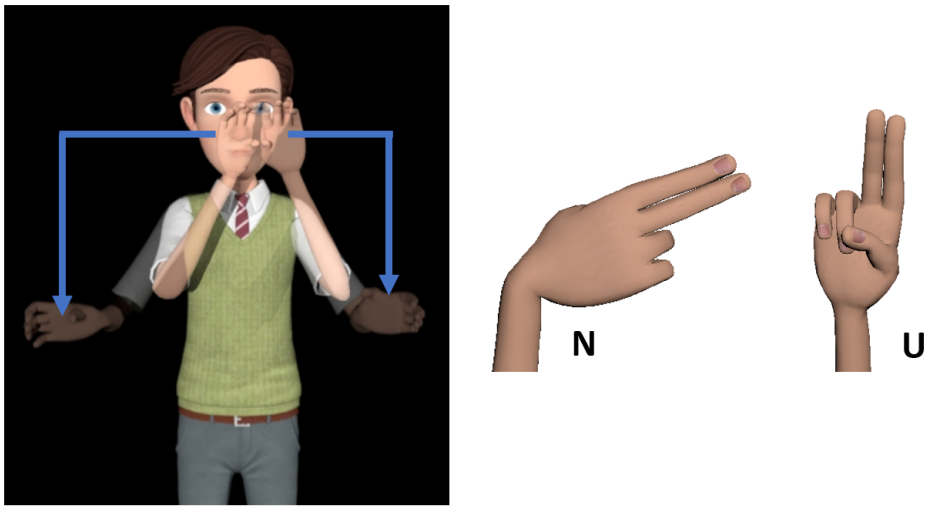

Figure 7.6 – Left: the sign [*CHEMINEE*] (*chimney*) in LSF. The hand configurations stay the same. Right: the letters 'U' and 'N' of the fingerspelling alphabet are composed of the same hand configuration despite a difference in hand orientation.

The automatic annotation of hand configurations is not a simple task as it involves automating both the segmentation and labeling steps. These two steps involve different processes and are therefore managed independently in the method developed here. The labeling step, in particular, corresponds to a classification task with 32 (*Sign3D*) or 48 (*LSF-ANIMAL*) classes, which represents a large number of classes given the small amount of data present. These numerous configurations can, in addition, be realized at any location in space and in any direction. Moreover, the two corpora that we use to test our method contain the data of three different signers. This makes the annotation task more complex while allowing us to measure the robustness of our technique. While the *LSF-ANIMAL* database has been conceived with the study of the hand configurations in mind, the *Sign3D* database was developed independently from those considerations. Both corpora present hand configurations in the context of utterances. Coarticulation effects alter the execution of some hand configurations and thus further complicate the recognition step. Finally, in sign languages, the spelling of some words of the oral language using the fingerspelling alphabet happens regularly. Determining the time segments in the case of spelling is particularly complex because the manual configurations taken successively follow one another very quickly.

Manual annotation is used as reference and training data for our automatic anno-

tation. It is thus necessary to have a thorough and precise annotation. The *Sign3D* and the *LSF-ANIMAL* databases have been annotated using the ELAN software [71].

The *Sign3D* corpus has been fully annotated on several tracks including, but not restricted to, gloss, hand placement, hand orientation, mouthing, facial expressions and hand configurations. To reduce the error rate and to have a more consistent annotation, two annotators knowledgeable in French Sign Language validated each others' work.

The hand configuration tracks of the *LSF-ANIMAL* corpus was annotated by one annotator but only on motion sequences with isolated hand configurations. The annotation method presented in this section was used to complete the annotation of the hand configuration tracks.

7.4.2 Method

The automatic annotation method for the hand configurations consists of three steps. First, we apply a segmentation step using the variations in the distances between the joints of each hand to separate the motion data into segments of two types, *hand configuration* or *transition*. The second step focuses on recognizing, using a supervised classification algorithm, the hand configurations performed at each frame of the *hand configuration* segments determined in the previous phase. Lastly, each *hand configuration* segment is annotated according to the predominant class in the segment (see overview on Figure 7.7).

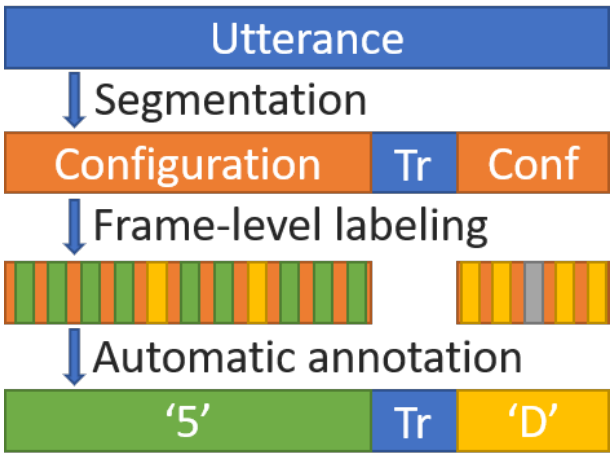

Figure 7.7 – Overview of the automatic annotation of the hand configurations.

The presence of a precise and complete manual annotation of the *Sign3D* corpus makes it possible to compute quantitative metrics measuring the difference between the manual

(ground truth) and the automatic annotation. We therefore used the *Sign3D* corpus to determine the descriptor set, algorithms and parameters of the different steps of the annotation method. We then applied the method thus parameterized on the partially annotated *LSF-ANIMAL* database to obtain the annotation of the hand configuration tracks.

In the remainder of this section, we first present the choice of the descriptors in Section 7.4.2.1. Then, we detail the labeling step at the frame-level and introduce the recognition models in Section 7.4.2.2. The next section, Section 7.4.2.3 presents two segmentation techniques, one of them relying on the findings of the previous section. Finally, Section 7.4.3 gives the results of the whole annotation process on *Sign3D* and *LSF-ANIMAL*.

7.4.2.1 Choice of the Descriptors

The processed *MoCap* data can be written as a vector of the $3D$ positions of the body joints at each frame. However, it does not constitute the best way of studying the hand configurations. Indeed, hand configurations are not directly defined by the absolute position of the hand and fingers in space (two identical hand configurations can be performed at different locations) but by the relative positions of the fingers with respect to each others. Using the initial data, we have thus computed descriptors that only depend on the analyzed feature; the nature of the hand configurations must not depend on the placement or orientation of the hands nor should it be sensitive to the morphological differences between signers.

While the positions and orientations of the joints vary according to the chosen reference frame, the Euclidean distances between two articulations and angles formed at each joint are invariant to the reference frame. The Euclidean distances between each joint of each hand were therefore chosen to describe the hand configurations. The equation (7.4) gives the calculation of the Euclidean distance between two joints $i(x_i,y_i,z_i)$ and $j(x_j,y_j,z_j)$.

$$d(ij) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{7.4}$$

In our model, each hand has 21 joints (four per finger and one for the wrist) resulting in a total of 210 possible combinations (our hand model is shown on Figure 7.8). However, some distances are more relevant than others. For example, the segments between two consecutive joints (e.g., the second and third joints of the middle finger) are physiologically similar to bones. Their distances only undergo small variations (due to noise in the data

and due to the fact that the rotations are not made around a single point, mainly because of the cartilage: the center of rotation is a centroid curve because of the sliding of one bone on the other [225]) and are not relevant to discriminate hand configurations.

The equation (7.4) gives the calculation of the angles $\angle AOB$ formed at each finger joint (A: previous articulation, B: next articulation, O: current articulation).

$$\angle AOB = cos^{-1} \frac{\overrightarrow{OA} \cdot \overrightarrow{OB}}{||\overrightarrow{OA}||||\overrightarrow{OB}||} \tag{7.5}$$

a) Descriptor Sets Eight sets of distances and/or angles have been tested in order to find the optimal feature set and to compare their impact on the annotation process.

1. *All distances*: the Euclidean distances between each joints of the hand model (210 for our hand model).
2. *All distances without bones*: the distances between each joints except the constant distances corresponding to bones (190 distances).
3. *29 distances*: A subset of the 29 most discriminating features (see Figure 7.8). It consists of the distances between:
   (a) the wrist and the extremities of the fingers (5 distances) to evaluate the bending of the fingers on the palm,
   (b) the extremity of one finger with its neighbors (5 distances) to measure the gap between the fingers,
   (c) the extremity of each finger and its corresponding knuckle (5 distances) to evaluate the bending of the fingers with respect to the knuckles, and
   (d) the extremity of the thumb and the joints of the other fingers (14 distances) to specify the behavior of the thumb.
4. *15 distances*: the distances (a), (b) and (c) of Figure 7.8.
5. *10 distances*: the distances (a) and (b) of Figure 7.8.
6. *5 distances*: the distances(a) of Figure 7.8.
7. *15 angles*: the angles at each joint with respect to the plane of each finger.
8. *29 distances and 15 angles*: the 29 distances of the feature set #3 and the 15 angles of the feature set #7.

The distance descriptor sets (#1 to #6) were tested in both the labeling and segmentation steps. The descriptor sets involving angles (#7 and #8) were only tested in the labeling step as they were not suited to our segmentation method which relies on the variation of the distances (see Section 7.4.2.3).

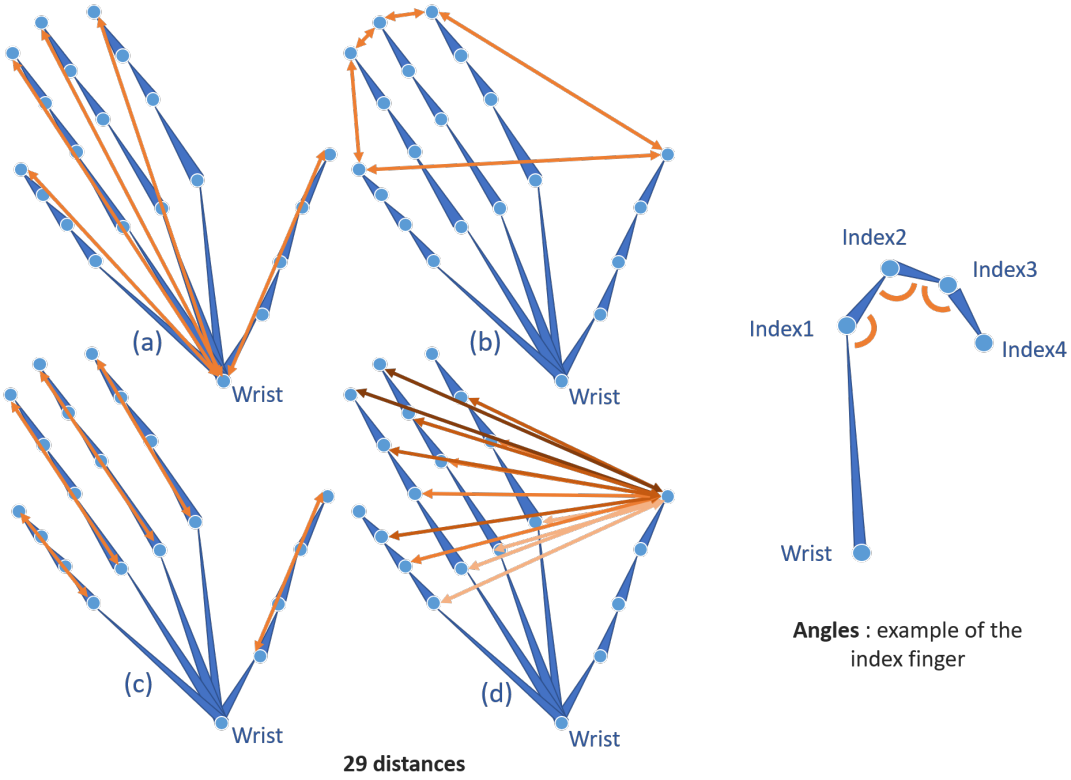

Figure 7.8 – Left: the descriptors corresponding to the 29 distances (descriptor set #3). Right: three angles of the angle descriptors used in the descriptor sets #7 and #8.

b) Normalization In order to have more generic results, to give each distance the same weight and to be less dependent on the signer's morphology, it is necessary to normalize our features.

In the remainder of this chapter, we test two types of normalization: first, the **max distance normalization** was performed by dividing each of the distances by its maximal value in the corpus for the same type of distance [1]. For example, each *wrist-indexTip* distance for a signer A was divided by the maximum value for the *wrist-indexTip* distance for signer A. All the distances have therefore a value between 0 and 1. Those distances

1. Those maximum values were obtained in range of motion sequences in which we ask the signer to spread his/her fingers as wide as possible

were then used to segment and label the hand configurations. This normalization thus describes each distance as a ratio of the maximum finger spacing and reduces the impact of the signer's morphology on the annotation process.

Second, the **standardization** of a distance x was performed by subtracting the mean value μ of the values taken by this distance in the entire data set and dividing the result by the standard deviation σ (see Equation 7.6).

For the angles, only the standardization was used.

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{7.6}$$

7.4.2.2 Frame-Level Labeling

The frame-level labeling consists in identifying and naming the hand configurations performed on each selected frame of the motion data. It is a multi-class classification problem in which the classes are the labels from the closed vocabulary of the hand configuration tracks. For the *Sign3D* corpus, 32 classes were defined corresponding to 32 different hand configurations (see fig. 7.9). For the *LSF-ANIMAL* corpus, 48 classes were identified.

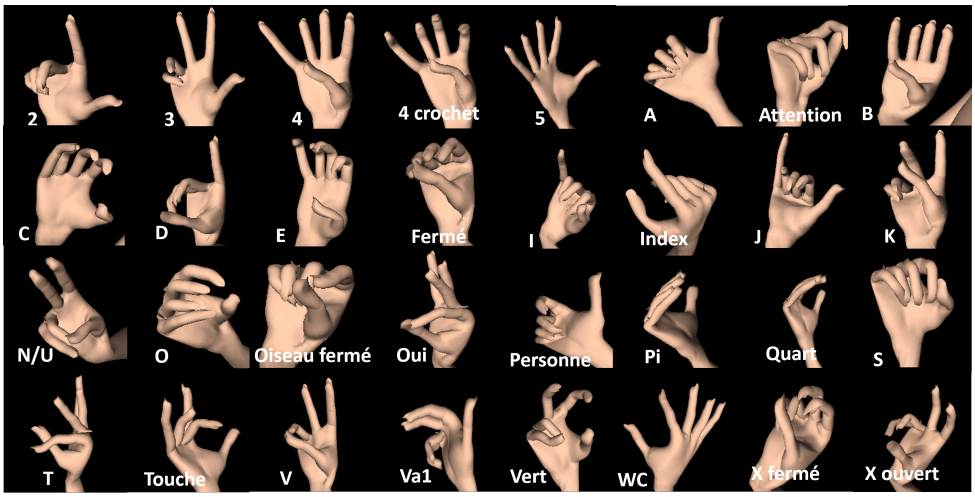

Figure 7.9 – The 32 hand configurations of the *Sign3D* corpus.

a) Methodology In order to select the descriptor set, the algorithm and the hyper parameters the most suited to the labeling of each frame, we tested different combinations

of the parameters with a 5-fold cross validation on a training set from the *Sign3D* corpus. The descriptor-algorithm-hyper parameters combinations with the best results were applied on a test set (still from the *Sign3D* corpus) to choose the final sets of parameters which were applied on the *LSF-ANIMAL* corpus.

For the labeling step, the tested parameters are :

— the descriptor sets: all the descriptor sets (#1 to #8) were tested.

— the classification algorithms: we tested 6 Machine Learning (ML) models: the Support Vector Machine (SVM), k Nearest Neighbors (kNN), Random Forest, Neural Network, Gaussian Naive Bayes and Logistic Regression algorithms.

— the hyper parameters of the algorithms: in ML, the hyper parameters are the parameters that are set before the training phase such as the number of neighbors for kNN or the type of kernel for SVM.

b) Training and Test Sets For the labeling step, our initial data is the position of the hand joints in a XYZ reference frame. However, only the frames **manually annotated as part of a hand configuration** were kept to test and select the different classification algorithms. The descriptors (distances or angles) were calculated on each of those 29415 frames of the *Sign3D* corpus. The training set consists of 83% of the data with 24365 examples. The remaining 17% constitute the test set.

Two frames belonging to the same hand configuration segment are inherently more similar than two frames belonging to different segments with the same label. To remove any bias, the test set consists of the hand configurations of 5 of the 25 utterances captured (see Figure 7.10). Therefore, the training set and the test set do not possess almost identical frames. We chose those utterances so that all of the hand configurations present in the test set are also present in the training set and the test set covers all of the 32 hand configurations of the *Sign3D* corpus. However, the hand configurations are not equally distributed in the corpus. Figure 7.11 presents the number of examples per configuration in the training and test sets.

The equations 7.7 and 7.8 correspond to the matrix X of training examples and to the vector y of labels associated with each example. N is the total number of training frames (24365) while D is the number of chosen descriptors (distances or angles). This number

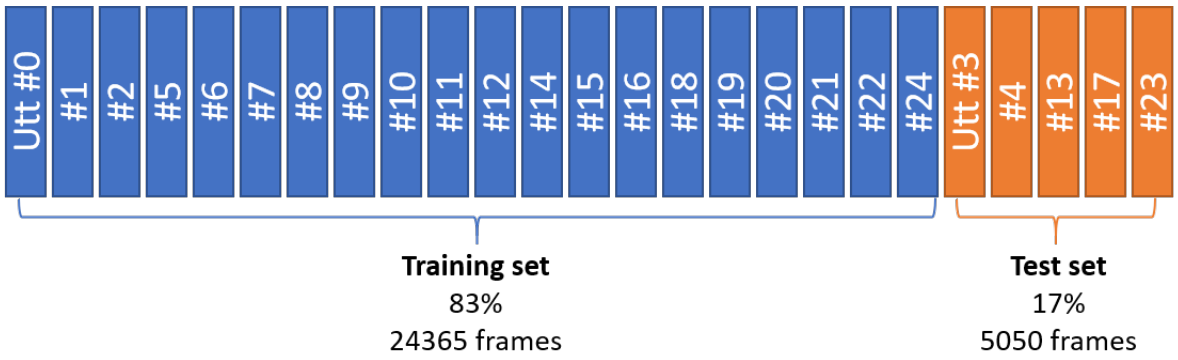

Figure 7.10 – Distribution of the training and test sets.

can vary from 5 (descriptor set #6) to 210 (descriptor set #1)

$$X = \begin{pmatrix} desc_1^{(1)} & desc_2^{(1)} & \dots & desc_D^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ desc_1^{(N)} & desc_2^{(N)} & \dots & desc_D^{(N)} \end{pmatrix} \tag{7.7}$$

$$y = \begin{pmatrix} LabelConfig^{(1)} \\ \vdots \\ LabelConfig^{(N)} \end{pmatrix} \tag{7.8}$$

c) Selection of the Algorithms Hyper Parameters To select the optimal hyper parameters for each machine learning algorithm tested, we performed a grid search with a stratified 5-fold cross validation on the **training set** using python functions and the scikit-learn library [2].

During a k-fold cross validation, the training set is divided into k subsets of equal size. One of these subsets is the validation set while the $k-1$ others are used to train the model. The process is repeated k times so that each subset is used as the validation set once and the results of the classification on each validation set are averaged. In the case of stratified k-fold cross-validation, the content of each subset is representative of the whole training set; the percentage of samples for each class is preserved. In our case, k is equal to 5 so that the size of the validation set is equal to 20% of the whole training set.

Table 7.3 lists the tested hyper parameters and the final choice for each algorithm (except for the Gaussian Naive Bayes algorithm which did not depend on any parameter). The choice was made with respect to the average accuracy score on the 5 values given by

2. `https://scikit-learn.org/`

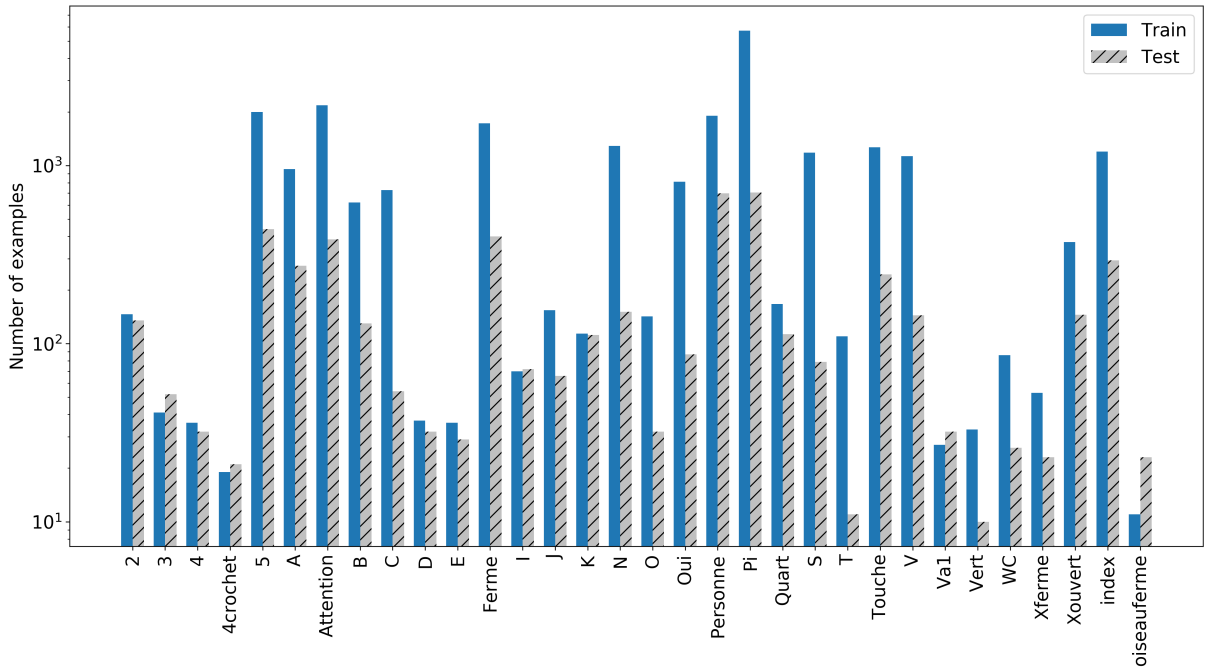

Figure 7.11 – Number of examples per configuration. As this number can vary from 34 ('oiseau fermé') to 6432 ('Pi') examples per configurations (training + test), the y-axis is scaled logarithmically.

the 5-fold cross validation. For N the number of samples in the tested set, y the vector of labels derived from the manual annotation (ground truth), $\hat{y}$ the vector of predicted values, we can define the accuracy score as:

$$accuracy(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N-1} 1(\hat{y}_i = y_i) \tag{7.9}$$

Where $1(\hat{y}_i = y_i)$ is equal to 1 when the label y_i for the sample i is equal to the predicted label $\hat{y}_i$.

For the SVM algorithm, we selected the radial basis function (rbf) kernel and the linear kernel because their accuracy scores were very close.

d) Selection of the Model and Descriptor Set The selection of the ML model and descriptor set to be used for the labeling of the hand configurations was performed in two steps. A first pruning was done with a stratified 5-fold cross validation that allowed to determine the ML models and a descriptor set that have the average best F1 score on the training set. We did not choose the very best combination on the validation set because

ML model	**Hyper parameters**	**Values tested**	**Best**
KNN	Number of neighbors	1, 3, 5, 7, 9, 11, 13	3
SVM	Kernel type	`linear`, `rbf` [a], `polynomial`	`linear` and `rbf`
	C [b]	0.1, 1, 10, 100, 1000, 10 000	0.1 (linear) and 1000 (rbf)
	γ [c] (with rbf kernel)	$1e^{-3}$, $1e^{-4}$	$1e^{-4}$ (rbf)
Random Forests	number of trees	20, 50, 100,200	100
	Number of features considered	5, 10, $\sqrt{Nbfeatures}$, all features	$\sqrt{Nbfeatures}$
	Maximum tree depth	3, 5, 7, 9, 11, No limit	No limit
Neural Network	Hidden layer size	(200), (100), (50), (100,50,25), (50,25,10)	(100)
Logistic Regression	Solver	`newton-cg` [d], `lbfgs` [e], `sag` [f], `saga` [g]	`newton-cg`

Table 7.3 – Results of a grid search on the ML algorithm hyper parameters: the chosen parameters are indicated in the last column.

a. Radial basis function.

b. Regularization parameter. The parameter C trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly (`https://scikit-learn.org/`).

c. Kernel coefficient. γ defines how much influence a single training example has. The larger γ is, the closer other examples must be to be affected (`https://scikit-learn.org/`).

d. Newton-conjugate gradient.

e. Limited-memory Broyden–Fletcher–Goldfarb–Shanno.

f. Implementation of the Stochastic Average Gradient [226].

g. Implementation of the algorithm proposed in [227].

the results of the cross validation are prone to over-fit as some frames from the same hand configuration segment can be found in the validation and training sets. However, the test and the training sets do not contain frames coming from the same segments. So, to reduce the impact of over-fitting, the ML models and descriptors with the best results were applied on the test set to select the very best model/descriptor combination.

For each class l, the precision and recall are defined as:

$$P_l = \frac{TP_l}{TP_l + FP_l} \tag{7.10}$$

$$R_l = \frac{TP_l}{TP_l + FN_l} \tag{7.11}$$

with TP_l: number of True Positive for the class l (i.e. number of samples labeled as l by both the ML model and the manual annotation), FP_l: number of False Positive for the class l (i.e. number of samples labeled as l by the ML model and not by the manual annotation), FN_l: number of False Negative for the class l (i.e. number of samples labeled as l by the manual annotation and not by the ML model).

For L the number of classes (i.e. the number of different hand configurations: 32), the precision and recall of the multiclass problem are defined as:

$$precision = \frac{1}{|L|} \sum_{l \in L} P_l \tag{7.12}$$

$$recall = \frac{1}{|L|} \sum_{l \in L} R_l \tag{7.13}$$

The F1-score is a weighted average of the precision and recall. It is defined as:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{7.14}$$

Figure 7.12 (top) shows the F1-score and training duration of the 7 ML models on the 8 descriptor sets on the training set. The ML models with the best F1-score are kNN, Random Forests, SVM with a rbf kernel, SVM with a linear kernel, and the logistic regression algorithm. We eliminated the logistic regression algorithm because of its high training duration. The descriptor sets with the best results are the *190 distances*, the *29 distances* and the *29 distances and 15 angles* descriptor sets.

Figure 7.12 (bottom) and Table 7.4 show the results on the test set. Based on this study, we have selected the **SVM model with a linear kernel** and the ***190 distances* descriptor set** to label the hand configuration frames which are determined thanks to the technique described in the next section.

ML model	Accuracy	Precision	Recall	F1-score	Training duration
Linear SVM	**93.84**	93.99	**86.93**	**90.32**	**3.71**
KNN	92.65	89.93	86.18	88.02	NA
Rbf SVM	93.60	90.88	86.82	88.80	4.39
Random Forest	89.62	**94.16**	82.20	87.77	29.79

Table 7.4 – Results of the ML models and the *190 distances* descriptor set applied on the test set.

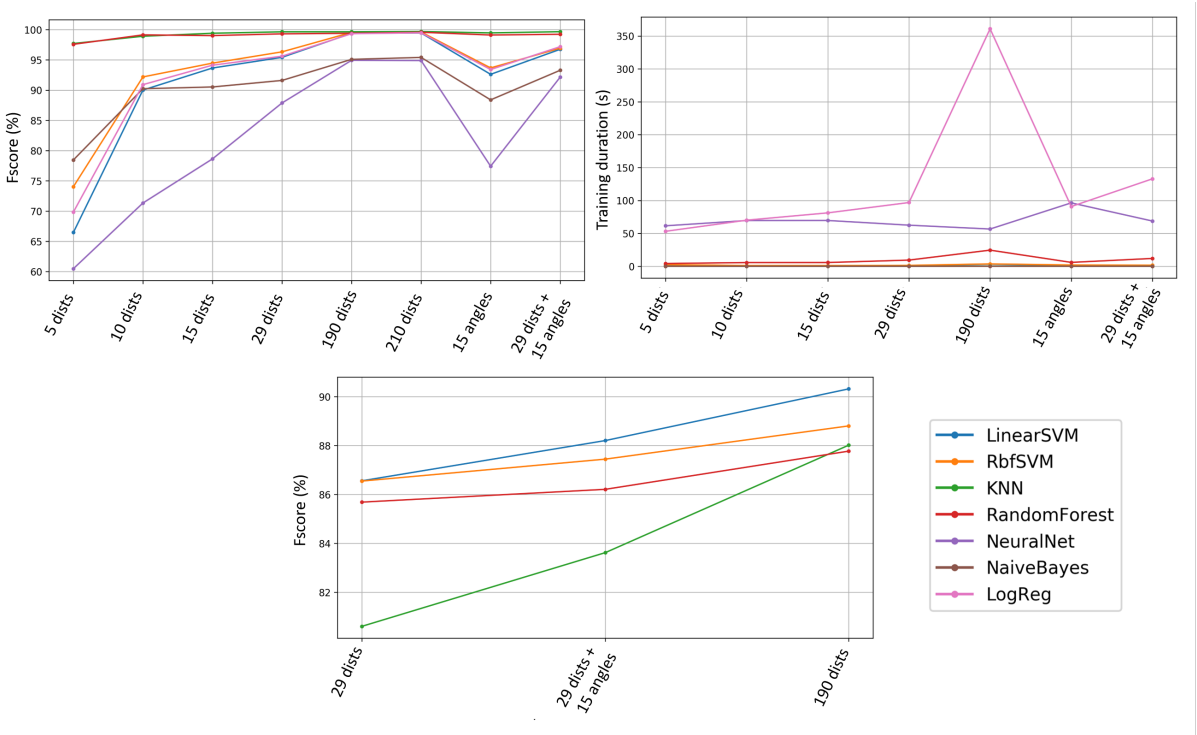

Figure 7.12 – Top: F1-score (left) and training duration (right) for each combination ML model/descriptors on the training set. Bottom: F1-score of the algorithms with the highest results applied on the test set. The best score is reached for the *190 distances* descriptor set and the linear SVM model.

Some configurations are more sensitive to confusion than others (see Figure 7.13). For example, the 'D' and 'index' configurations are always mistaken (in the two configurations, the index finger is raised and the others are folded). These configurations are very similar. The annotators themselves have sometimes difficulties differentiating between them. Errors in the manual annotation of the training set have been found which may explain the recognition errors. The 'WC' and '5' configurations are also confused because there are not enough 'WC' samples in the training set and the two configurations are similar (see Figure 7.9).

7.4.2.3 Automatic Segmentation

The segmentation step therefore consists in separating the continuous signing sequences in two types of segments : *hand configuration* or *transition*. The two segmentation techniques under consideration are a) distance variation and b) segmentation by recognition and are described hereafter.

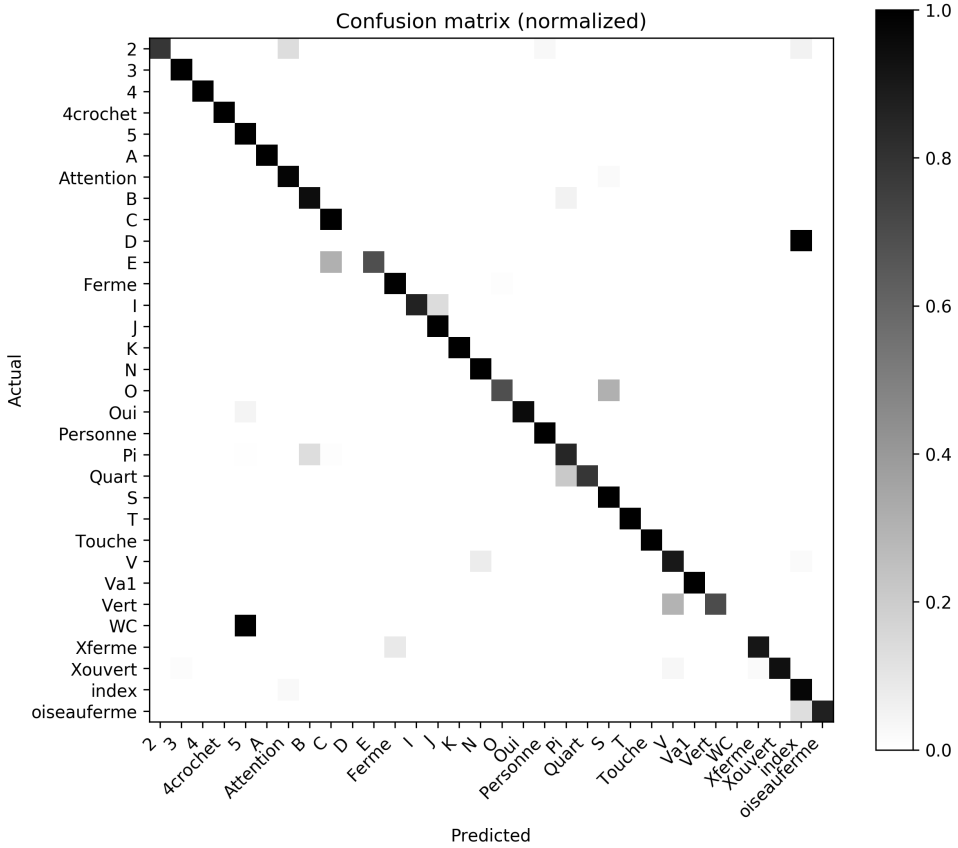

Figure 7.13 – Confusion matrix for the linear SVM model on the 190 *distances*. The two main confusions are between 'D' and 'index', and 'WC' and '5'.

a) Segmentation Technique #1: Distance Variation To perform the segmentation, we assume that the variation of the distances is discriminating, i.e. the variation will be small during stable configurations and high during transitions. For each frame f, and for each selected distance SD, the variation of the distances $d(i,j)$ between two joints (i and j) is computed between the frame $f-1$ and f. Those variations are summed (see Equation 7.15 for the right hand RH).

$$VarDist_{f,RH} = \sum_{i \in RH} \sum_{\substack{j \in RH \\ d(i,j) \in SD}} |d(i,j)_f - d(i,j)_{f-1}| \tag{7.15}$$

The segmentation relies on the use of a threshold. If *VarDist* is above this threshold, a *transition* segment will be detected. If *VarDist* is below the threshold, the segment will be recognized as a *hand configuration* segment. To select the value of the threshold and to evaluate our segmentation, we used the *Simple Matching Coefficient* (SMC) metric [228].

It measures the similarity between two sets of values (here, the manual and the automatic segmentations). The SMC is the ratio of the number of overlapping frames between the two segmentations on the total number of frames. Equation 7.16 shows the calculation of this coefficient. p is the number of frames considered as *hand configurations* in both automatic and manual segmentation, n is the number of frames considered as *transitions* in both segmentations, and t is the total number of frames.

$$SMC = \frac{p+n}{t} \tag{7.16}$$

In the corpus, the distributions of the two types of segments *transition* and *hand configuration* follow the same order of magnitude (*transition*: 31%, *hand configuration*: 69%). With the SMC, we consider that the two types of segments have the same weight. This is not the case for other metrics [3] that we do not consider appropriate here.

We compared the results of the automatic segmentation by varying three parameters:

— The descriptor set; the sets #2, #3, #4, #5, #6 were tested. The set with all the distances was not chosen because adding the invariant distances (the "bones") would add noise and could only badly impact the results. The descriptor sets involving angles were not deemed appropriate as we wanted to measure the impact of the variation of the distances.

— The normalization type; the distances were used *without any normalization*, with the *max distance* normalization and with the *standardization*.

— The segmentation threshold; different values of threshold have been tested depending on the descriptor set and the type of normalization.

Figure 7.14 shows the variation of the SMC on the whole *Sign3D* corpus with respect to the descriptor set, normalization type and threshold. Except for the range of the threshold which depends on the descriptors and normalization, the curve profiles are quite similar. Two SMC are computed: the "avg SMC/utt" is the average of the SMC of each of the 25 utterances while the "total SMC" is the SMC of the whole corpus. As a consequence, the

3. Like the Jaccard index of equation 7.17 which only measures the number of frames considered as *hand configurations* in both the automatic and manual segmentation and not the *transition* frames.

$$J = \frac{p}{t-n} \tag{7.17}$$

"avg SMC/utt" gives the same weight to each utterance whereas the "total SMC" gives a weight proportional to the utterance length.

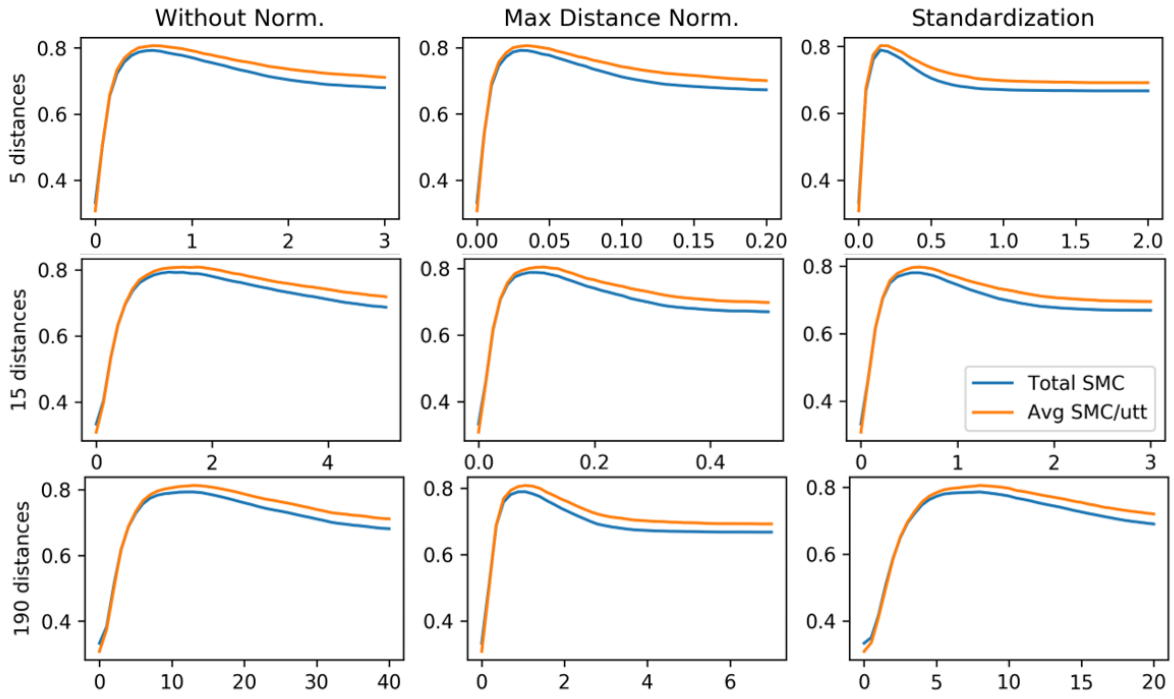

Figure 7.14 – SMC scores with respect to the descriptor sets (from top to bottom), normalizations (from left to right), and thresholds (abscissas of the curves). The profiles of the curves are similar while the abscissas vary as we do not sum the same number of distances and as the distances can be normalized.

Table 7.5 details the maximum SMC results and the corresponding threshold depending on the descriptor sets and normalization. We can see that the results are quite similar with an 80% match between the manual and automatic segmentation. The best results are obtained for the descriptor set with 190 distances (set #2) and no normalization. However, the generalization of the segmentation method requires the normalization of the data. We chose to select the parameters that obtained the second best results: the 190 distance descriptor set with the *max distance* normalization and a threshold equal to 1.100. This threshold is dependent on the sampling rate (i.e. 100Hz for the *Sign3D* corpus) as it corresponds to the distance covered between two consecutive frames.

Figure 7.15 makes it possible to qualitatively compare the automatic and the manual segmentation of two utterances.

Descriptors set	Without Norm.	Max Distance Norm.	Standardization
5 distances	0.807 (0.590)	0.806 (0.035)	0.803 (0.155)
10 distances	0.811 (1.250)	0.807 (0.086)	**0.805** (0.405)
15 distances	0.808 (1.700)	0.804 (0.108)	0.798 (0.633)
29 distances	0.808 (2.950)	0.805 (0.200)	0.800 (0.975)
190 distances	**0.813** (13.000)	**0.809** (1.100)	**0.805** (8.000)

Table 7.5 – Average of the SMC per utterance (avg SMC/utt) and, between parentheses, the corresponding threshold.

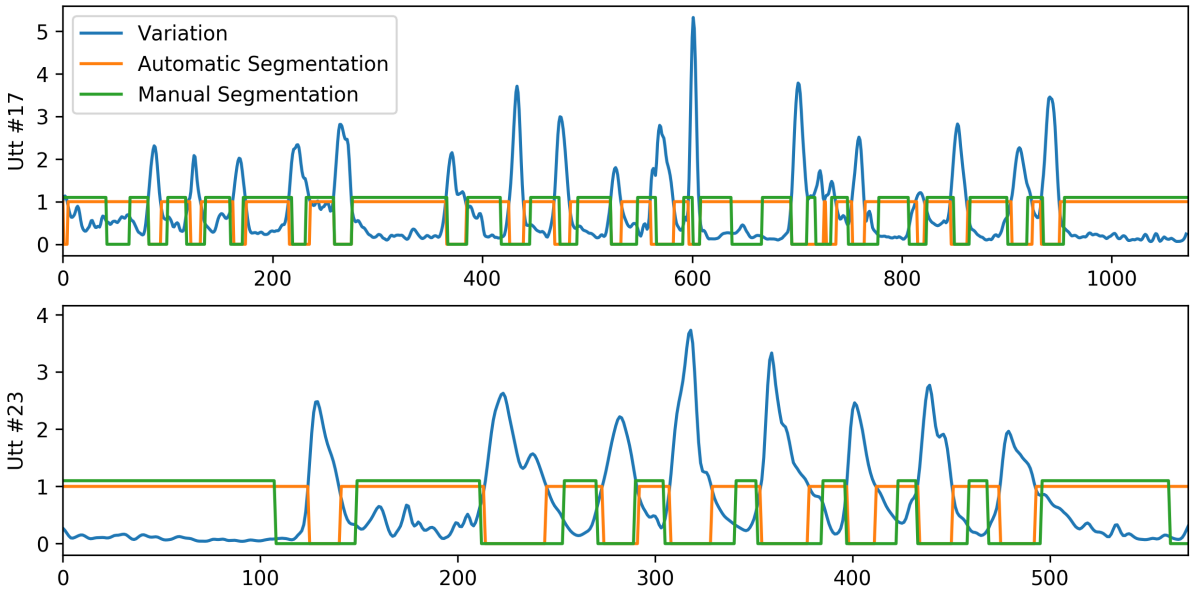

Figure 7.15 – Automatic and manual segmentation of the hand configuration for two utterances with the 190 distances descriptor set, the max distance normalization and a threshold of 1.1. The blue line represents *VarDist* for the right hand (*Sign3D* corpus).

b) Segmentation Technique #2: Segmentation by Recognition The distance variation segmentation method uses a threshold that measures the quantity of motion between two consecutive frames. As such, it is dependent on the frame rate but also on the velocity of the motion performed by the signer.

A second segmentation technique was considered to compensate for the limitations of the first technique. The goal of the segmentation is still to separate the stream of signing data into *hand configurations* and *transition* segments but instead of relying on a threshold measuring the variation in the descriptors value, the segmentation threshold measures the probability of each frame to be a hand configuration.

In this case, the results of the labeling step is exploited and a linear SVM classifier

is trained on a manually annotated portion of the data. Then, each frame of the tested utterance is given to the classifier that outputs the probability of the frame to belong to one of the 32 configurations. If this probability exceeds a threshold, the frame is considered to be part of a *hand configuration* segment.

Figure 7.16 shows the variation of the SMC with the segmentation by recognition method. The maximum is reached for a threshold of 70% with an SMC of 0.754. This segmentation method gives lower SMC results for the *Sign3D* data but is more flexible than the segmentation technique #1. However, this segmentation method constitutes a preliminary approach and further inquiries would have to be made, for example, by testing different ML models for recognition.

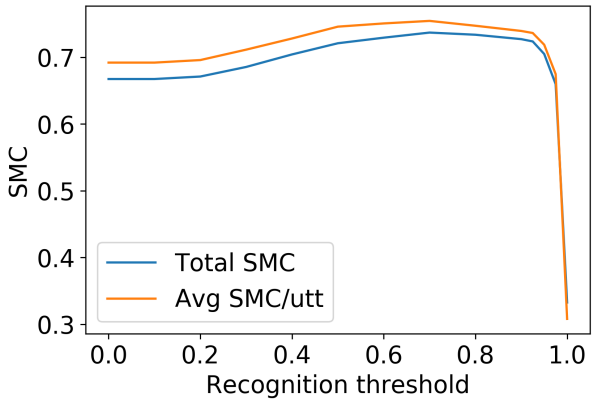

Figure 7.16 – Variation of the SMC with the segmentation by recognition method on the right hand (*Sign3D* corpus). The maximum is reached for a threshold of 70% with an SMC of 0.754.

7.4.3 Results of the Annotation of the Hand Configuration Tracks

The hand configurations of a continuous sequence of movements can be annotated by performing:

1. the segmentation to separate the phases of stable hand configurations from the transition phases, then,
2. the frame-level labeling exclusively of the segments of type *hand configuration* with the selected ML model and descriptor set, and, finally,
3. by determining the predominant class in each *hand configuration* segment (i.e. the segment is identified with the label with the highest number of occurrences).

7.4.3.1 Results on *Sign3D*

Figure 7.17 shows the annotation of the five utterances of French Sign Language belonging to the test set and Table 7.6 gives a quantitative evaluation of the results. The annotation was evaluated by considering the transitions as an additional class. Each frame of the annotated sequence is compared with the label of the manual annotation (or the absence of label in the case of a transition). The descriptor set used for both the segmentation and the labeling is the *190 distances* set which obtained the best results for the two tasks. The segmentation technique #1 was used with a segmentation threshold set to 1.1.

The machine learning model used here is SVM with a linear kernel. While the results can differ from one utterance to another, the recognized labels and segments are mainly consistent with the manual annotation. The results given by the metrics (i.e. accuracy, recall and precision) are computed with respect to the manual annotation and are therefore limited by the overlap of the automatic segmentation with the manual segmentation (SMC score in Table 7.6). There are few errors in terms of recognition of hand configurations (accuracy of 90%).

On Figure 7.17, we can note that the utterance #23 contains a fingerspelling part to spell the name "ANTOINE". Those letters, performed rapidly thanks to the fingerspelling alphabet, were correctly segmented and identified by our method.

Utt. id	Utt. translation	Accuracy	Precision	Recall	F1-score	SMC
3	The museum behind the tourist information office opens at 8.20 am and closes at 7 pm.	64.3	57.7	62.4	60.0	73.5
4	Admission fee is 12.5€	87.2	85.3	91.8	88.4	87.2
13	The festival normally takes place in the amphitheatre but if the weather is bad then it will be moved to the museum.	75.2	78.6	79.4	79.0	78.6
17	The festival normally takes place in the amphitheatre but if the weather is bad then it will be moved to the museum. (done with one hand)	71.9	65.2	70.9	67.9	80.9
23	The child's name is ANTOINE. His parents must be looking for him at the reception.	75.5	57.3	74.2	64.7	78.8
Avg		**74.8**	**68.8**	**75.7**	**72.0**	**79.8**

Table 7.6 – Quantitative results of the hand configuration annotation for each sequence of the test set.

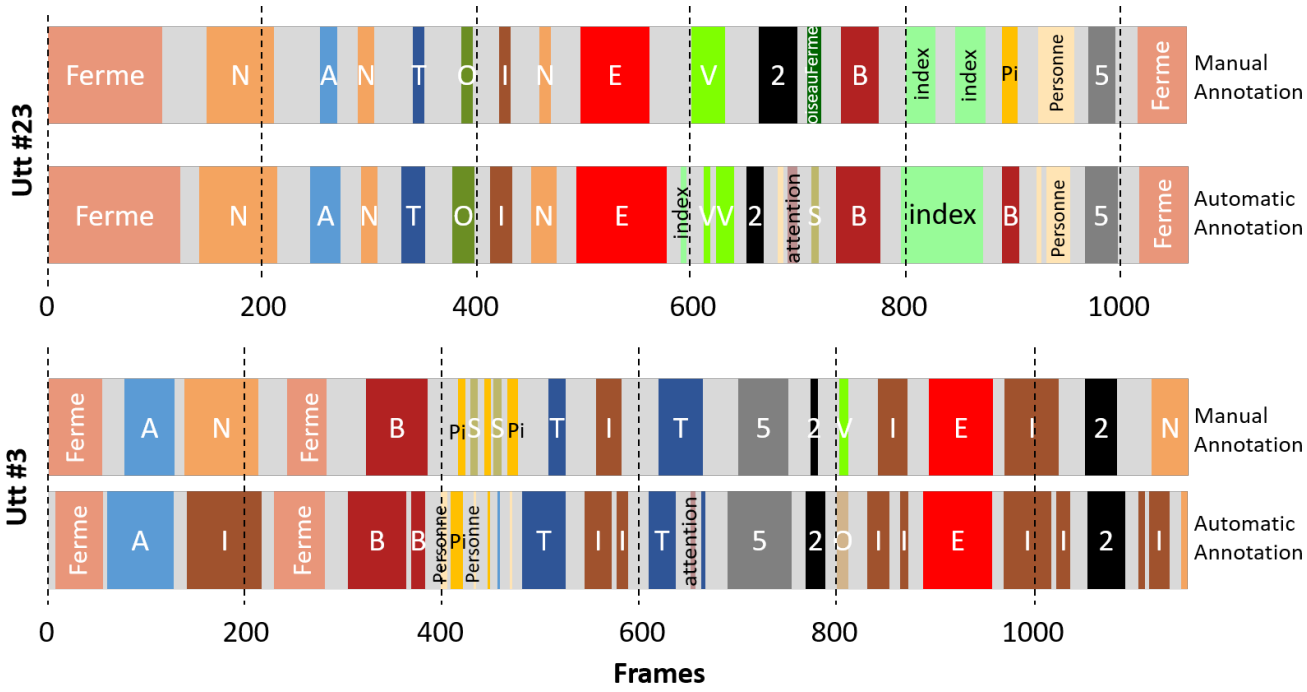

Figure 7.17 – Qualitative results of the hand configuration annotation for two sequences of the test set (*Sign3D* corpus).

7.4.3.2 Application on *LSF-ANIMAL*

The method to automatically annotate the hand configuration of a corpus was applied to the *LSF-ANIMAL* data set in order to rapidly provide an annotation that can later be refined by a human annotator.

The hand configurations of a small part of the *LSF-ANIMAL* corpus had been manually annotated beforehand to be used as training sequences and to obtain an estimation of the performance of the annotation method. The motion sequences containing the isolated hand configurations (part 1 of section 6.1.2) and 3 descriptions of animals (among the 26 descriptions of part 3 in section 6.1.2) were manually annotated. We used our method to provide the annotation of the rest of the corpus. Only the methods which obtained the highest results for *Sign3D* were applied on the *LSF-ANIMAL* data set (i.e. for the segmentation, the "distance variation" technique and, for the labeling, the SVM with a linear kernel) [4].

4. The automatic annotation of the *LSF-ANIMAL* data set was made to rapidly provide an annotation of the hand configuration tracks. As a consequence, few manual annotations were made beforehand. Due to the absence of ground truth for the whole *LSF-ANIMAL* data set, fewer numerical results were obtained for the *LSF-ANIMAL* data set than for the *Sign3D* data set.

a) Segmentation We performed the "distance variation" segmentation on *LSF-ANIMAL* and obtained an averaged SMC of 62.8% on the motion sequences that we had manually annotated. This relatively low SMC is due to various factors: (i) the signer of the *LSF-ANIMAL* data set made longer pauses between two signs or utterances and those pauses were detected as static *hand configurations* while those segments were not annotated as *hand configuration* by the manual annotator as they did not correspond to actual sign language hand configurations, (ii) the threshold of the segmentation method is dependent on the signing velocity of the signer. As the signer of *LSF-ANIMAL* and of *Sign3D* are distinct, their signing velocity can be slightly different.

To remove the errors due to the first factor, we added *hand configuration* segments called '*repos*' (rest) corresponding to the hand configurations of the signer in the moment of pause between two signs or utterances. We thus reached an average SMC of 79.8%. An example of segmentation result is visible on Figure 7.18

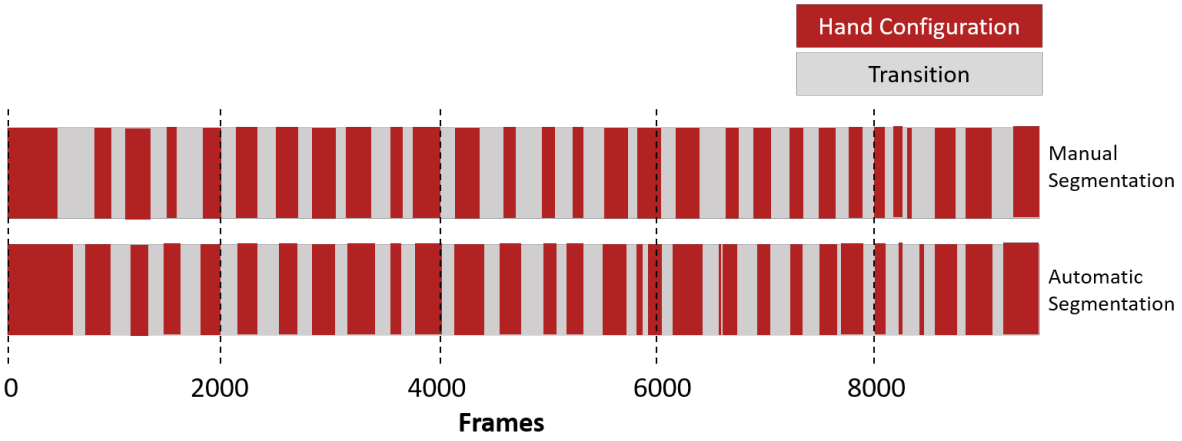

Figure 7.18 – Results of the segmentation on a sequence of *LSF-ANIMAL*. The red segments are the *hand configuration* segments and the gray are the *transition* segments.

b) Frame-level labeling For the labeling, we trained the linear SVM model on the motion sequences containing the isolated hand configurations and on one of the sequences describing a cat ("CHAT#2"). The evaluation was performed on the two remaining description sequences ("CHAT#1" and "CHIEN#1"). We obtained an accuracy of 72.95%, a precision of 71.22%, a recall of 72.95% and a F1-score of 71.28%. We can note a drastic decrease of the performance (F1-score of 90.32% for *Sign3D*) that can be explained by the greater number of classes in the *LSF-ANIMAL* database (48 different hand configurations instead of the 32 classes of the *Sign3D* corpus) and the fact that the main part of the

training data included isolated hand configurations deprived of the modifications caused by the addition of a context.

c) Annotation The annotation was done by determining the predominant class in each hand configuration segment. We obtained an average accuracy of 74.93%, precision of 69.64%, recall of 66.21% and F1-score of 67.02% on the 2 descriptions of animals that were not in the training set (see visual result on Figure 7.19).

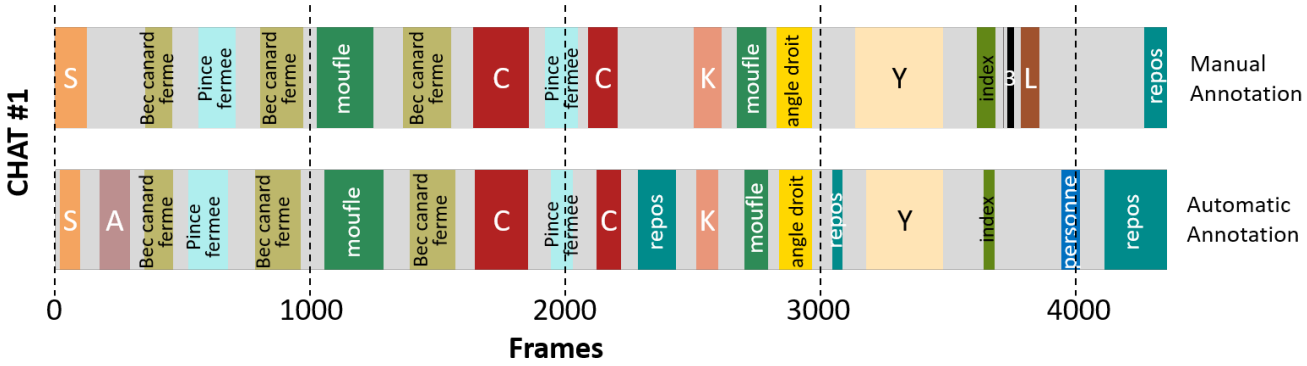

Figure 7.19 – Results of the annotation on an utterance (a cat description) of *LSF-ANIMAL*.

7.4.3.3 Limitations of the Method and Perspectives

The hand configuration labeling of *LSF-ANIMAL* is less effective than the labeling of *Sign3D* because of the number and nature of the hand configurations, of the fact that the signers are different between the two corpora and of the lack of training data containing hand configuration in different contexts for *LSF-ANIMAL*. Indeed, we only manually annotated the sequences of isolated hand configurations in which the hand configurations are relatively invariant.

At this time, the genericity of the annotation method is limited. Even though the data has been normalized, the results could certainly be improved by doing further work on the normalization of the hand data in order to be less dependent on the morphology of the signer. It could be interesting to perform inter-subject recognition tasks by training our classifier on the data of the *Sign3D* signer before applying it to the *LSF-ANIMAL* data set. To do such experiments, it would be necessary to homogenize the two data sets in terms of number and nature of hand configurations which would require to adjust the

manual annotation of both data sets. We did not find the resources for such a task in the context of this thesis but it constitutes an interesting perspective of this work.

Furthermore, the segmentation based on the variation of the inter-finger distances relies on a threshold that measures the quantity of motion between two consecutive frames. It is dependent on the frame rate but also on the velocity of the motion performed by the signer. A lower SMC can reveal a difference in the signing speed between the *Sign3D* and the *LSF-ANIMAL* corpus and further work to select an adapted threshold should be performed.

However, with an accuracy of 75% in the *LSF-ANIMAL* corpus, the method constitutes a useful tool to assist a manual annotator in his/her work. Indeed, we used the raw results of this automatic annotation for our synthesis work.

Finally, we performed the annotation by determining, on each *hand configuration* segment, the predominant class. To this end, we selected the class with the higher number of occurrences in the segment in question. Other solutions for the determination of the segment label could be investigated to replace the majority vote (e.g., the class present in the highest number of adjacent frames could be chosen).

7.5 Automatic Annotation of the Hand Placement

Finally, in this section, we propose to automatize the annotation of the hand placement component from *Motion Capture* data. In our annotation scheme, the hand placement corresponds to six tracks (see Section 7.2): **Emplacement_Distance_MD**, **Emplacement_Distance_MG**, **Emplacement_Hauteur_MD**, **Emplacement_Hauteur_MG**, **Emplacement_Radial_MD** and **Emplacement_Radial_MG**.

7.5.1 Motivations and Challenges

The hand placement, like the hand configuration, is a phonological component of sign languages that can impact the meaning of a sign. Some mechanisms like pointings or the spatialization of signs in which the location of the sign changes according to its context, heavily depend on hand placement. For analysis and synthesis purposes, it is interesting to be able to query motions with respect to the hand placement. Like for the other tracks,

the manual annotation of the hand placement is made more difficult due to the unique viewpoint of the RGB-camera used for the annotation and its lack of depth information. Moreover, unlike hand configuration that can be precisely defined by the position of the fingers, the quantitative values of hand placement (i.e. the discrete areas) are not visible on a video recording which can sometimes make the manual annotation of the hand placement subjective.

When annotating the *hand placement*, the challenges come from both the definitions of "placement" and of "hand". Should the placement of the hand be defined by precise numerical values or by global areas of space? Can we assume that the placement of the tip of the index or the wrist joint is the placement of the hand?

With the exception of certain gestures requiring a precise contact with the body or the location of one object with respect to another, signs are made, at least cognitively speaking, in areas of space: a difference of a centimeter in the location where a sign is made will not affect the meaning of the sign. We have therefore chosen to define global areas corresponding to a discretization of the signing space. The precise definition of this discretization is presented in Section 7.5.2 and the computation of the hand placement for the *LSF-ANIMAL* corpus is described in Section 7.5.3 [5].

7.5.2 Discretization of the Signing Space

The strategies for the discretization of the signing space depend on the purpose of the annotation. We based our division of space on existing work on space discretization.

7.5.2.1 Existing Work

Different work in linguistics and computer animation aim to discretize the continuous space around the signer. For example, Lenseigne *et al.* [229] propose to use a semantic model of the signing space to formalize the spatiotemporal structures of LSF. The different zones of their model are deduced from the analysis of different video recordings of signed utterances. Their model, using a prediction approach, aims at the recognition of full utterances. However, the signing space areas described in their model are not fixed but depend on the utterances and on the part of the body (tip of the finger, hand or

5. Moreover, depending on the granularity of discretization of the signing space, some signs can be performed in more than one hand location in space with a placement at the beginning of the sign different from the one at the end.

wrist) considered. Such a model is not convenient for the automatic annotation of sign language content. In [11] (Figure 7.20, left), Millet defines 6 zones in the signing space that possess a syntactic role. Zone ① represents the neutral placement whereas the zones ② represent the areas in which the hand places a third person as a subject or object of an action (for example, in the signed sentence "A says to B", A will be located in one of the ② areas while B will be in the other and the verb [TO SAY] will be performed from A to B). The definition of Millet associates a syntactic role to each of the hand placements. While it is essential for linguistic purposes, it does not possess the objectivity needed for the annotation as it does not correspond to an annotation scheme for hand placement but to the syntactic rules deduced from the observation of an already annotated corpus.

On the contrary, Lefebvre-Albaret *et al.* [230] (Figure 7.20, center) directly propose an annotation scheme for the *Sign3D* corpus. Twelve identical cubic zones are defined with three letters corresponding to the values taken by three axes: lateral, height and depth. The "GBL" zone of Figure 7.20 is the "Left - Bottom - Far" zone *(GBL = Gauche - Bas - Loin)* from the point of view of the signer. While the signing space is often represented as a sphere around the signer with a radius equivalent to the size of the signer's arm, the total volume covered by this representation is a rectangular cuboid: the reaching capability of the signer is not considered. Moreover, the volume covered does not take the head level into consideration nor locations slightly behind the signer (e.g., the [SPINE] sign can be done in LSF by pointing one's back). Finally, signs in the neutral space (① of Figure 7.20, left) are situated between two to four zones which is neither convenient for annotation, nor for motion querying.

A low-level space specification has been proposed in [148] to model the hand location for the application of sign synthesis. In this proposition, the hand placement is described by combining (i) a direction (e.g., 'up', 'down', 'forward', 'right') or a combination of two or three directions (e.g., 'down-right', 'right-forward-up') and (ii) a distance ('proximal', 'medial', 'distal' and 'extended'). This representation, centered around the signer, is adapted to the specification of SL content for sign synthesis. However, the use of directions instead of areas complicates the annotation task as the signing space is not covered exhaustively.

The identification of signing areas must be adapted to both the linguistic characteristics of the motions (e.g., the neutral space must preferably be defined by only one area)

and the annotation constraints (e.g., the areas should be precisely defined and should cover the total volume of the signing space). For example, the 140 areas of the signing space defined by Kipp *et al.* [231] (Figure 7.20, right) are centered around the signer thanks to a radial orientation component. This representation was intended for the annotation of co-verbal gestures and not sign languages but it is a precise and human-centered representation that can be reused for SL purposes. However, the work of annotation and querying is complicated by the hand-dependency of radial orientation component (e.g., the "inward" location of the right hand will be the "out" location for the left hand and vice versa).

As we did not find a perfectly adequate solution in the existing work, we defined our own discretization of the signing space for the annotation of the hand placement component.

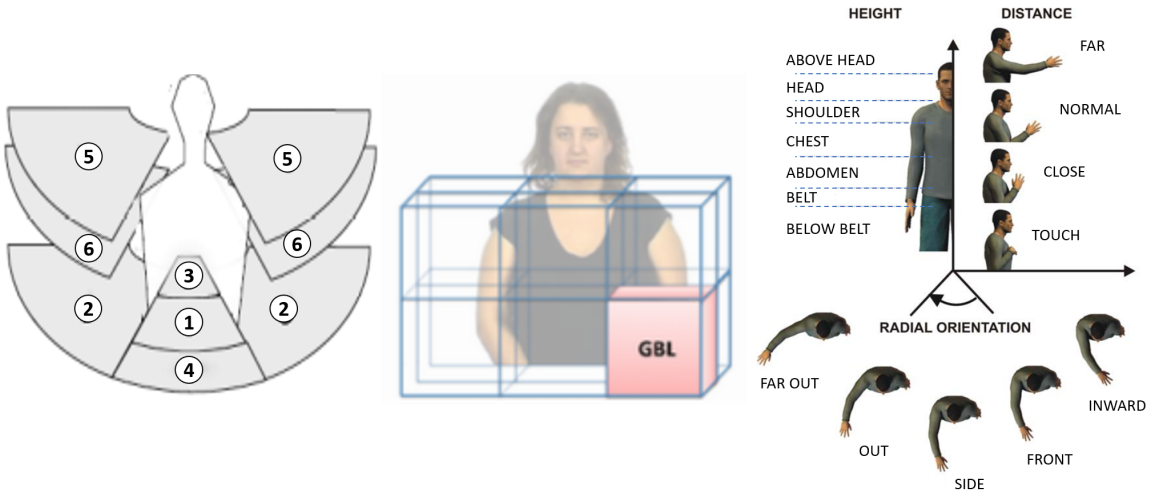

Figure 7.20 – Different representations of the signing space. Left, a syntactic representation [11]. Center, a geometric representation for manual annotation [230]. Right, a human centered representation [231].

7.5.2.2 Definition of our Discretization

Three elements need to be established in order to have a complete definition of the discretization of the signing space: the hand definition, the center of reference and axes of the annotation system, and the precise division of space. We felt that the representation proposed by Kipp *et al.* [231] was the most suitable for our needs in the annotation of the hand placement component. We therefore chose to use this proposal as a basis for our

own discretization.

Hand definition: Compared to the size of the human body, the hand is not an object that can be reduced to a point in space due to its length, width and thickness that are not negligible. It is therefore necessary to define a precise location on the hand whose position will be considered as the hand placement. The joint at the **base of the middle finger**, because of its central position in the hand, was chosen.

Center of reference and axes of the annotation system: The placement of the hand is dependent on a signing space which is specific to each signer and centered on the signer. Moreover, the notions of distances are relative to the signer's height, morphology and size and, in particular, to the size of his/her arms. Our placement areas are therefore defined according to the position of the signer and relatively to the size of the signer's arms. Each hand is placed into a coordinate system with three axes. Therefore, like in Kipp *et al.* [231], a location of the hand will be encoded with three parameters: the height with respect to the ground, the distance between the hand and the body and the radial orientation of the hand (see Figure 7.21).

Division of space: The granularity of the division will impact the size of the areas of the signing space. These areas must be linguistically significant while being precisely and quantitatively defined. For each axis (height, distance and radial orientation), we chose labels and boundaries that we thought suited to the description of hand placement in the context of sign language. Unlike the representation of Kipp *et al.* [231], the radial orientation is defined in an absolute way, not depending on the hand, which simplifies the annotation and querying work.

The areas of the division of the signing space are delimited by precise criteria on the position of the joints of the reconstructed skeleton of the signer. This reconstructed skeleton moves in a 3 dimensional Cartesian coordinate system defined by the X, Y and Z axis shown in Figure 7.22 (left).

H is the joint representing the hand for the computation of the hand placement. In practice, it is the joint at the base of the middle finger. The precise position of H in the Cartesian coordinate system of the signer is represented by (x_H, y_H, z_H). For the computation of the hand placement, we used (x'_H, y_H, z'_H) where $x'_H = x_H - x_{Hips}$ and

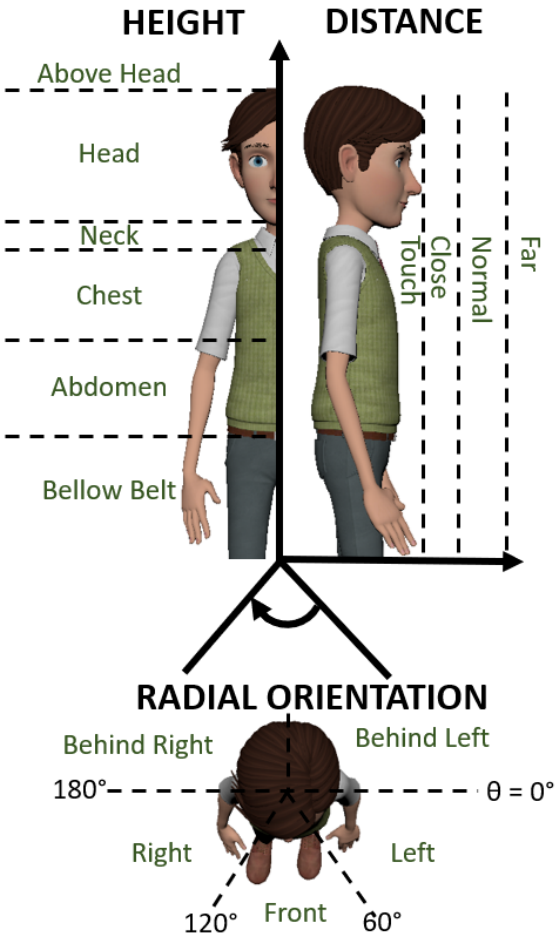

Figure 7.21 – Placement of the hands with respect to three dimensions: height, distance and radial directions. Figure inspired from [231].

$z'_H = z_H - z_{Hips}$ to center the lateral axes X and Z around the *Hips* joint of the signer (i.e. the root joint). y_H is used as such to determine the *Height* dimension of hand placement.

D represents the distance between H and the Hips-Neck axis at each frame. To compute the *Distance* dimension of the hand placement, D is compared with $SIZE\_ARM$ which represents the size of the arm (right and left are distinguished). Equation 7.18 shows the computation of D, $\times$ representing the cross product and $\vec{s}$ is the spine vector going from Hips to Neck_base while $\vec{SH}$ is the Hips to H vector. The constant $SIZE\_ARM$ is the sum of the size of each bone from the shoulder to H.

$$D = \frac{||\vec{s} \times \vec{SH}||}{||\vec{s}||} \tag{7.18}$$

Dimension	**Labels**	**Segmentation Criteria**
Height	Above head	$y_H > y_{Head\_end}$
	Head	$y_H \in]y_{Neck\_end}; y_{Head\_end}]$
	Neck	$y_H \in]y_{Neck\_base}; y_{Neck\_end}]$
	Chest	$y_H \in]y_{Sternum}; y_{Neck\_base}]$
	Abdomen	$y_H \in]y_{Hips}; y_{Sternum}]$
	Bellow Belt	$y_H \leq y_{Hips}$
Distance	Touch	$D < 1/4 * SIZE\_ARM$
	Close	$D \in [1/4 * SIZE\_ARM; 5/12 * SIZE\_ARM[$
	Normal	$D \in [5/12 * SIZE\_ARM; 3/4 * SIZE\_ARM[$
	Far	$D \geq 3/4 * SIZE\_ARM$
Radial Orientation	Behind Left	$x'_H > 0$ and $z'_H \leq 0$
	Left	$\theta \in]0°; 60°]$
	Front	$\theta \in]60°; 120°]$
	Right	$\theta \in]120°; 180°[$
	Behind Right	$x'_H \leq 0$ and $z'_H \leq 0$

Table 7.7 – The labels and segmentation criteria for the hand placement.

Finally, θ represents the angle (in degrees) between the coronal plane (see Figure 7.22, right) and the Hips to Hand vector to determine the *Radial orientation* dimension of the hand placement:

$$\theta = acos(\frac{x'_H}{\sqrt{{x'_H}^2 + {z'_H}^2}}) \tag{7.19}$$

Each of the chosen label for each dimension is listed in Table 7.7. The segmentation criteria for each label are also given on Table 7.7 depending on the previously defined position of H, D and θ. To determine the *Height* component, the position of the hand is compared to the position of different joints of the body such as the sternum or the base of the neck. The *Distance* component is specified with respect to the size of the arm. The "Touch" area does not correspond to a real contact between the body and the hand as we only used the skeletal data for the annotation which do not contain the signer's volumetric information. We therefore empirically fixed a limit at $1/4 * SIZE\_ARM$ by reviewing examples on the data set. As for the boundaries of the *Radial Orientation* dimension, they are computed in a straightforward manner to have three zones of equal size in front of the signer and two zones behind.

7.5.3 Application on the *LSF-ANIMAL* corpus

The computation of the hand placement segments and annotation labels was performed on each frame of each motion sequence of the *LSF-ANIMAL* database. Our definition

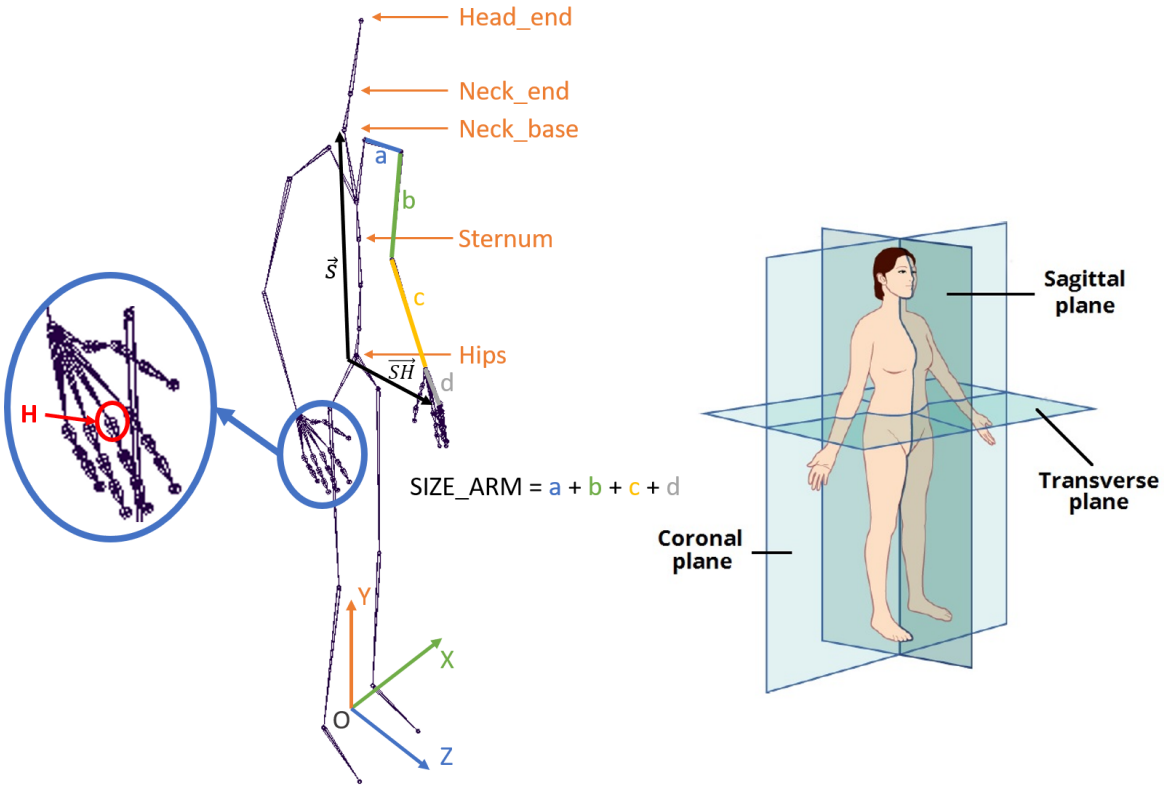

Figure 7.22 – Left: the coordinate system and the points of reference for the computation of the hand placement component. Right: The three anatomical planes of the human body (figure from `http://www.liberaldictionary.com/sagittal/`).

of hand placement covers the entire signing space. As a consequence, motion sequences are annotated continuously: there is no transitory segment between two hand placement segments. Moreover, instead of defining one track of hand placement for each hand which would force us to concatenate the results of the three dimensions to obtain small segments labeled with tags such as "Head-Normal-Left", we decided to keep one track per dimension and per hand which justifies the six tracks of the hand placement component. Figure 7.23 shows an example of the annotation of the hand placement tracks.

7.6 Summary and Discussions

The annotation of a data set is an important post-processing step when doing data analysis or data-driven synthesis. Indeed, it adds a semantic layer to the raw motions

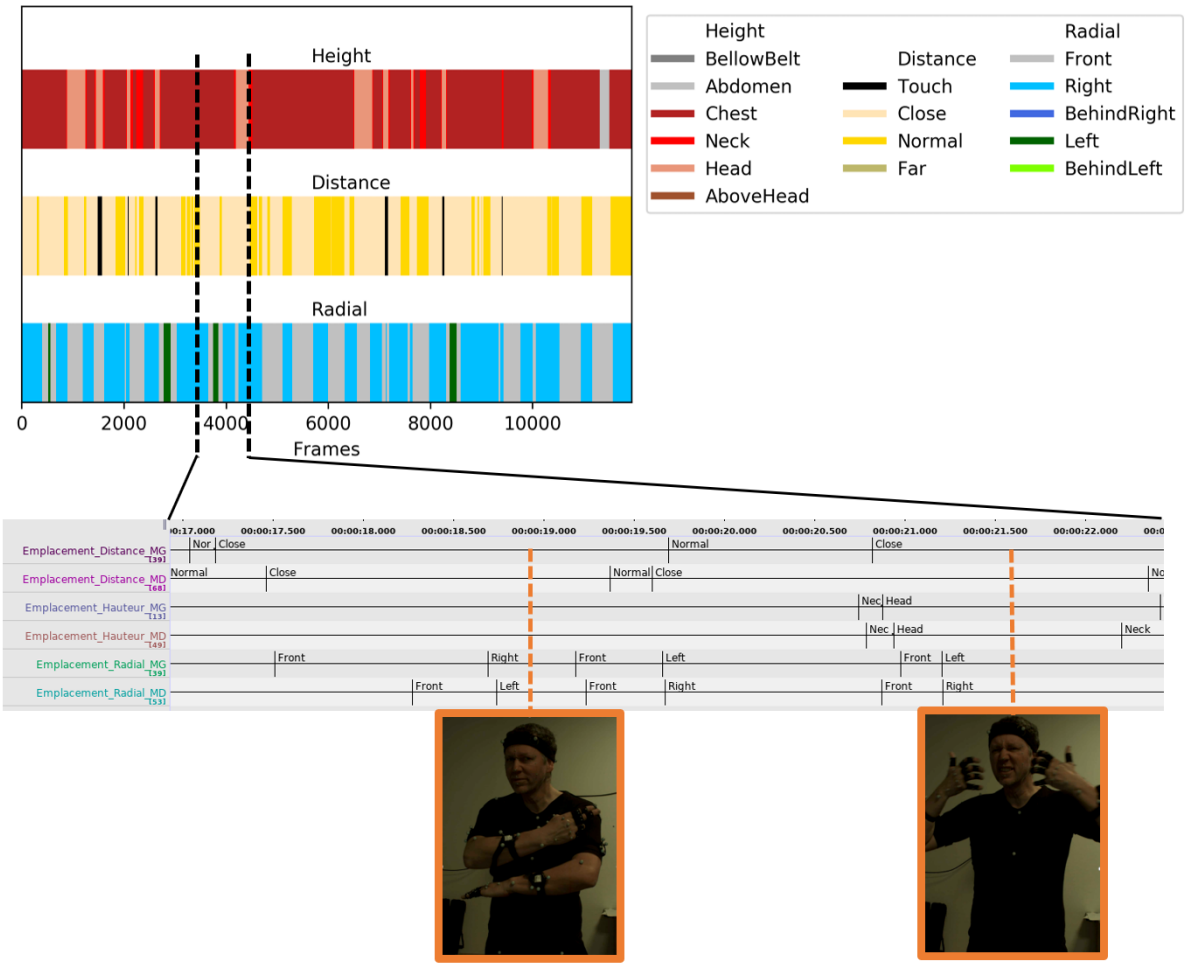

Figure 7.23 – Annotation of the hand placement tracks of one motion sequence. Top: the annotation of the whole sequence for the right hand on the three placement tracks. Bottom: a focus on a few frames of the sequence on ELAN with two illustrated examples.

contained in the data set which can then be queried according to linguistic constraints. Following the parametrical approach of SL, we defined an 18-tracks annotation scheme with a dedicated track for each of the hand components (configuration, placement, movement and orientation) and four additional tracks for the gloss level.

To limit the inaccuracies and errors of the manual annotation and to reduce the annotation time, we proposed to automatize or clarify the segmentation and labeling of some tracks.

We first presented a refinement of the segmentation of the **gloss tracks** done by analyzing the kinematic properties of sign language motions and, more specifically, by

detecting local minima in the norm of the velocity for each hand. The manual annotations done by deaf experts were then refined by selecting the nearest corresponding minima for each manually defined segment. This segmentation is based on the observation that each hand has a partially autonomous behavior and resulted in a segmentation at a gloss level for each hand even for two-handed signs. The use of precise quantitative features makes our segmentation less biased, more accurate and more homogeneous than the manual segmentation.

In a second section, we presented an automatic technique to segment and label the **hand configurations** based on 190 normalized distance features of the hand. The segmentation is based on a measure of the quantity of variation in the finger positions and on the detection of a threshold crossing of this measure. Unlike the PCA segmentation described in [85], this segmentation method has the advantage of not only determining the boundaries of segments but also of giving the nature (*hand configuration* and *transition*) of the segment. The labeling step is based on machine learning methods trained on a small, manually annotated, subset of the data to assign a tag to each *hand configuration* segment. This method was trained and tested on the *Sign3D* corpus, and applied to reduce the manual annotation time on the *LSF-ANIMAL* database. As we use machine learning models for labeling, the results could be significantly improved by increasing the size of the data set. 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 data sets.

Finally, we presented a computation of the **hand placement** with respect to three spatial dimensions: *Height*, *Distance* and *Radial orientation*. The annotation of hand placement was done according to our discretization of the signing space which is relative to the signer's position and reaching capabilities. The computation was performed on the *LSF-ANIMAL* corpus to allow the query and synthesis of motions with respect to their location in space.

The question of the evaluation of the results is complex. The reduction of the annotation time is a simple feature to compute provided that the development time of the automatization is not considered. However, the quality of the automatic annotation compared to the manual annotation is harder to determine. Indeed, how can we measure the quality of our annotation if the reference is flawed ? As our synthesis is based on the

annotation, a perceptual evaluation of the synthesis results could validate the annotation process.

Chapter 8

Motion Synthesis and Editing for the Generation of New Sign Language Content

Contents

8.1 Overview 194
8.2 Building New Signs with Phonological Recombination 199
8.3 Utterance Synthesis 231
8.4 Summary and Discussions 244

> His limbs were in proportion, and I had selected his features as beautiful. Beautiful! Great God! His yellow skin scarcely covered the work of muscles and arteries beneath; his hair was of a lustrous black, and flowing; his teeth of a pearly whiteness; but these luxuriances only formed a more horrid contrast with his watery eyes, that seemed almost of the same colour as the dun-white sockets in which they were set, his shrivelled complexion and straight black lips.
>
> *Frankenstein; or, The Modern Prometheus*
> Mary Shelley, 1818

This chapter is an extended version of an article submitted to Machine Translation's Special Issue on Sign Language Translation and Avatar Technology in 2020.

In this chapter, we propose to use the raw data and knowledge extracted from the annotated French Sign Language (LSF) database described in chapter 6 to synthesize new LSF content. First, in Section 8.1, we present our objectives and the constraints specific to the synthesis of sign language as well as the adopted approach. Then, the following sections deal with the synthesis of SL content on two different levels: at a sign level in Section 8.2 and at an utterance level in Section 8.3

8.1 Overview

8.1.1 Objectives

In the previous chapters, we presented the capture and annotation of an LSF database. The resulting corpus contains approximately one hour of motion data which is hardly representative of the entire language. In this chapter, we thus propose different techniques to synthesize specific SL mechanisms absent from our database in order to enrich it with new content.

As the new content is part of an actual language, it must meet strict requirements in terms of accuracy and realism. To satisfy both requirements, we use our initial data set in two ways:

1. As analysis material: the study of the kinematic profile of specific instances of the selected LSF mechanisms makes it possible to understand, model, and reproduce the involved motion phenomenon. It is then possible to select and parameterize a synthesis technique that will allow us to be as close as possible to the ground truth.
2. As raw synthesis material: the raw data can be cropped, transformed and recombined to compose new signs having the kinematic and dynamic properties of the original motions.

In both cases, we heavily rely on the decomposition of signs into distinct phonological components with a special interest on three of the manual components: hand placement, hand configuration and hand movement.

The techniques presented hereafter make it possible to synthesize new content at a sign level with the generation of signs in their citation form, deprived of any contextual information, and at an utterance level with the implementation of inflection and articulation mechanisms.

8.1.2 Approach

In order to select and tune our synthesis techniques, we follow an analysis/synthesis approach which can be broken down into several stages (see Figure 8.1). First, a linguistically relevant mechanism is selected provided that some instances of the aforementioned mechanism are present in the database. Those instances are extracted from the database and are considered as ground truth. The observation of the ground truth provides a first overview of the specific phonological components and, at an animation level, the particular joints involved in the studied mechanism (the correspondence between the phonological components and the joints of the avatar's skeleton is described hereafter). The time series constituted by the translation and rotation of those joints along time are then studied in order to choose the synthesis techniques that best fit the results. This synthesis technique is used to reproduce the instances present in the ground truth or other instances of the same phenomenon. Finally, the synthesis technique is validated by perceptually evaluating the synthesized phenomenon.

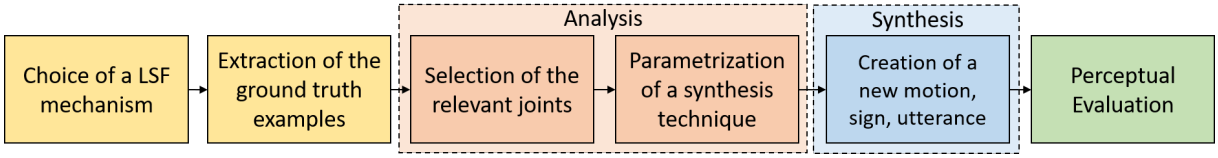

Figure 8.1 – Overview of the analysis/synthesis approach for the synthesis of LSF mechanisms.

8.1.3 Our Synthesis System

Our synthesis platform is the result of different works carried out over the years by numerous researchers and engineers. As a result, an animation software platform has been developed, allowing data-based control of a signing avatar, with a multi-channel approach [64], [222]. Except for Subsection 8.1.3.3 which corresponds to a result of the thesis work, the current section (Section 8.1.3) describes the global functioning and the query system of the platform as they existed before the beginning of this thesis. The following sections (Section 8.2 and Section 8.3) correspond to contributions.

8.1.3.1 Skeletal Animation of an Avatar

In this thesis, we use the skeletal animation method described in Section 5.1: an avatar is a human-shaped $3D$ mesh that is driven by the motions of a skeleton which is a

hierarchical structure composed of rigid bones and joints.

The state of the skeleton at a given frame is called a skeleton's pose or posture. As bones are rigid segments that cannot grow or shrink, the specification of the absolute position and orientation of the root joint (positioned at the hips level) and the relative orientations of the other joints of the hierarchy are sufficient to fully describe a skeleton's pose. From this specification, it is possible to calculate the orientation of each of the joints in an absolute way (with coordinates in a world reference frame, generally chosen in the environment or close to the avatar's center of gravity) or in a relative way (each joint being positioned relative to its parent joint). Thus, the relative or absolute position of the joints can be deduced from the successive relative orientations. These different ways of specifying a posture are used in our data-driven synthesis system.

A skeleton's motion is defined by a sequence of skeleton's poses and a frame rate which indicates the number of poses per seconds. Therefore, a motion M is a sequence of k poses; its duration equals to k/f where f is the frame rate in frames per seconds (fps or Hz). In our system, the skeleton resulting from the *MoCap* and the skeleton driving the avatar's mesh are two distinct skeletons. The first one is adapted to the morphology of the two signers in our database [1] and the second one is adapted to the avatar and is rigged to the mesh, i.e. each vertex of the mesh is linked with a weight to one or more joints in order to move along with this joint. A retargeting operation, carried out by Motion Builder [217], allows the avatar to be animated using *MoCap* data. As the avatar's mesh does not precisely match the signers' morphology, there may be imprecisions in the location of a sign or problems of mesh interpenetration, especially between the hand and the head, which is much larger than average due to the "cartoon" design of our avatar.

This chapter focuses on the synthesis of motions at the skeletal level: the problems of retargeting are not within the scope of this chapter and we consider, in the remainder of this chapter, that the transition from skeletal movement to $3D$ avatar movement is implicit.

8.1.3.2 Heterogeneous Database and Query System

The synthesis system is based on a dual database containing both a motion data set indexed by the motion signal that has been recorded during the *MoCap* sessions and a semantic data set indexed by the annotation labels and annotation tracks [172].

The system can be queried in a number of different ways, either by requesting a

1. Incidentally, our two signers have similar morphologies.

movement directly from the motion database or by querying the semantic database (see Figure 8.2).

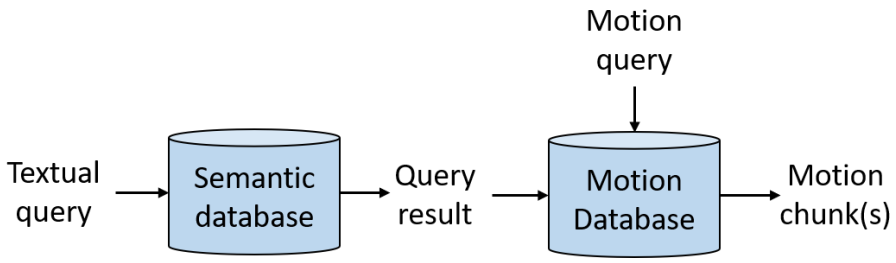

Figure 8.2 – The motion database can either be queried directly or through the semantic database.

In the case of the motion query, it is possible to extract a motion by specifying the start and end frames. The motion can then be extracted for each joint of the whole body or only for a specific set of joints.

In the second case, using the semantic database, it is possible to make a query by label value and/or by track (e.g., hand placement, gloss, hand configuration). The results of the textual query are then sent back to the motion database and the relevant movements are retrieved. Track-based queries are bound to a set of joints (see Section 8.1.3.3). The movements resulting from these requests are therefore the movements of these joints or, if specified by the operator, the movements of the whole body. Textual queries can give several results: for example a query on the [*CHIEN*] (dog) sign gives 5 results because there are 5 instances of [*CHIEN*] in *LSF-ANIMAL*. In this case, it is possible to retrieve all instances, a specific instance per index, a random instance or the instance that minimizes the distance to another motion (see Section 8.3.1.4).

8.1.3.3 Mapping Between the Phonological Components and the Avatar's Joints

The use of a captured database and the objective of animating an avatar forces the definition of a mapping between each phonological component (and, consequently, the corresponding annotation track) and a set of joints of the avatar's skeleton. Indeed, if we want to extract the values taken by a component (e.g., the hand configuration) from the database to replay it on an avatar, it is necessary to define a correspondence between the component in question and the joints relevant to this component. It is obvious, for example, that the configuration of the left hand will not impact the movement of the chest

or left shoulder. We call *channel* the set of joints corresponding to a component. Thus, the right hand configuration channel is the set of joints involved in the hand configuration phonological component for the right hand.

We tried to define a mapping that is both straightforward and has as little overlap as possible between the sets of joints corresponding to different components to have an independent control of each set of joints (see Table 8.1 and Figure 8.3) [2]. Thus, we get:

— **Hand configurations** are performed by all the finger joints. A configuration corresponds to the shape taken by the hand: the articulation of the wrist, which only participates in the movement and orientation of the hand, is not concerned.

— **Hand movements** correspond, in this thesis, to the trajectory of the wrists during the execution of a sign and, consequently, to the motion of the arms [3]. We mapped the hand movement with the shoulder, elbow and wrist of each arm.

— **Hand placements** are the result of a movement of the whole arm possibly involving a movement of the torso. However, in the synthesis work, hand placement only corresponds to the overall position of the wrist joint [4].

— **Hand orientations** are managed by the wrist joints [5].

Phonological component	**Channel**
Hand configuration	Finger joints
Hand movement	Shoulder, elbow and wrist
Hand placement	Wrist
Hand orientation	Wrist

Table 8.1 – Mapping between the phonological components and the avatar's joints.

2. Even if we did not separate them explicitly in Table 8.1, there are two instances (right and left) of each component. The "right hand orientation" will thus be mapped to the "right wrist" and the "left hand orientation" to the "left wrist".

3. We remind the reader that, in this thesis, we consider that the secondary movements performed by the fingers correspond to changes in the **hand configuration** and that the movements performed by the hand with respect to the wrist correspond to changes in the orientation of the hands.

4. In the annotation work, hand placement designates the position of the base of the middle finger. The wrist was chosen for synthesis because placement management is typically a problem of inverse kinematics. As such, we chose to treat the arms and hand independently and to consider the arm (from shoulder to wrist) as the articulated chain controlled by inverse kinematics. Even if a different joint was used to compute the hand placement in the annotation, the position of the wrist is still available in the database; the use of different joints to designate hand placement in annotation and synthesis is therefore not problematic.

5. Hand orientation does not constitute a focus of this thesis. We nonetheless specified the joints assigned to this component as it can be of use for further work.

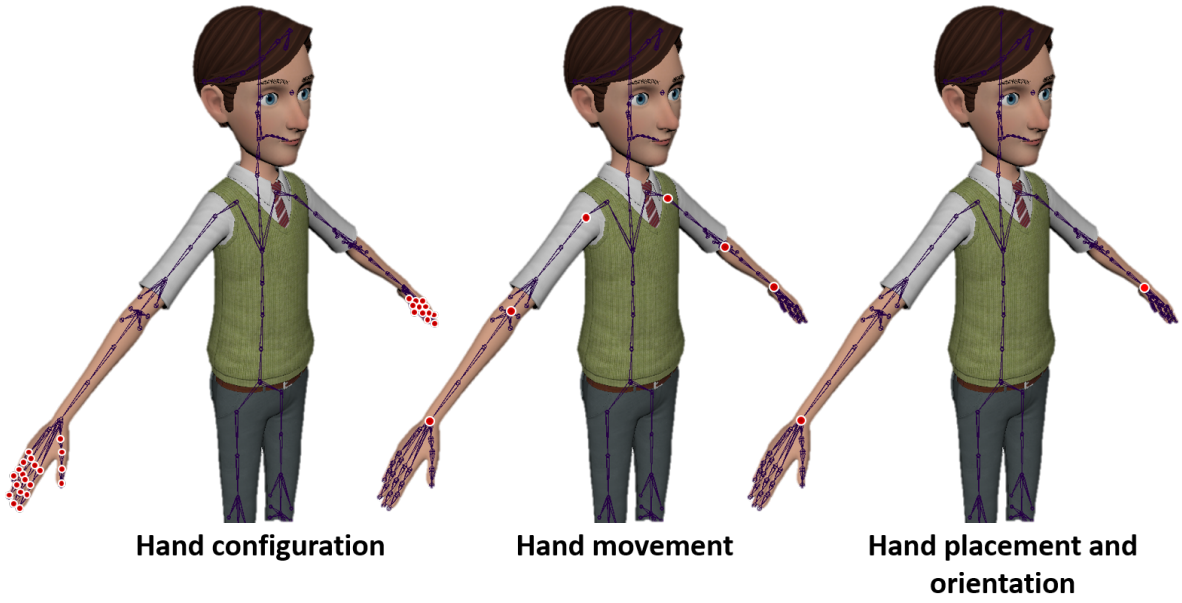

Figure 8.3 – Set of joints corresponding to each phonological component.

8.2 Building New Signs with Phonological Recombination

Following the linguistic works presented in Chapter 3, we propose to enrich our initial database by taking advantage of a phonological definition of French Sign Language and by working on three components: hand placement, hand configuration and hand movement.

We call phonological recombination the act of modifying each phonological component independently to create new content and, in this way, be less limited by the initial database. Thus, we noticed that new signs and inflections could be created by modifying the value taken by only one of the phonological components independently of the others (e.g., changing the hand placement of a sign makes it possible to create its spatialized form). We thus constructed Table 8.2 that lists the SL mechanisms we are working on in this section with respect to the phonological components involved.

Considering a database annotated at a phonological level, it is possible to retrieve and modify the value taken by one phonological component during the realization of a sign and to overwrite the existing value for the component with this new value to create a new sign. This technique has the advantage of enabling the creation of new realistic content as it is based on human data, however, in practice, a motion retrieval/overwriting operation alone is not enough to obtain correct content. Several challenges must be addressed in

Level	Hand Placement	Hand Configuration	Hand Movement
Phonological		Dactylology	Articulatory motion
Sub-lexical		Derivative base	
Lexical	Spatialization	Size and Shape Specifiers	Size and shape specifiers
Syntactic	Pointing	Proform	Trajectories

Table 8.2 – List of the SL mechanisms we are working on in this section with respect to the phonological components involved (inspired by [1]).

order to achieve an acceptable result:

1. If the value of the component with which we want to overwrite the current value does not exist in the database, it is necessary to create it. To do this, it is possible to modify existing values or to synthesize them from scratch.
2. The modification of a channel can have an impact on joints that are not part of the channel. For example, if the hand placement channel of a sign is changed, the entire position of the arm will have to be modified as well.
3. It may be necessary to synchronize the value of the modified channel with the other channels.
4. A component can take several successive values, if one or more of these values are changed, it is necessary to synthesize the transition between each of the values taken.

In this section, we aim to provide technical solutions to synthesize the phenomena listed in Table 8.2. The structure of the chapter reflects the structure of the table with the first three sections presenting the synthesis methods developed to create new linguistic content with the modification of the value of each one of the phonological components. Section 8.2.1 relates to the creation of new content with the modification of the hand placement component, Section 8.2.2 refers to the hand configuration and Section 8.2.3 deals with the hand movement. Finally, Section 8.2.4 gives some information about the synchronization of the different manual components.

8.2.1 Hand Placement Mechanisms

The hand placement represents the location of the hands in the signing space. In terms of animation, we define hand placement as the position of the wrist joint in the signing space. Hand placement is often specified thanks to a finite set of areas around and on the signer. Indeed, the exact position of the wrist is less relevant than the overall zone where

the wrist is positioned [6]. Each sign possesses a default hand placement that corresponds to the placement of the sign in its citation form.

To enrich our database, we propose to work on two SL mechanisms involving the hand placement component: spatialization (Section 8.2.1.1) and pointing gestures (Section 8.2.1.2). These application cases are interesting because they make it possible to describe a wide variety of new situations by simply taking advantage of the possibilities of inverse kinematics combined with a database containing the relevant signs.

8.2.1.1 Spatialization: Modification of the Placement by Inverse Kinematics

Some signs, in their form of citation, are performed on a specific location on the body (e.g., [*PENSER*] (*think*) on the side of the head, see Figure 8.4, left) or on a specific place of the signing space (e.g. [*CHIEN*] (*dog*) on the signer's side, see Figure 8.4, middle). The hand placement of these signs is thus invariant. Other signs have a default hand placement in the neutral place like [*BOL*] (bowl) (see Figure 8.4, right). In this case, it is possible to change the hand placement according to the context of the sign in an utterance. This mechanism of sign relocation is called **spatialization**.

Spatialization is an inflection mechanism that makes it possible to precisely place objects in a scene in an absolute ("the house is on the left") or relative way ("the house is near the swimming pool"). In the first example, the [HOUSE] will be performed on the left of the signer, while, in the second case, the same sign will be done near the place where [SWIMMING POOL] has been placed.

Spatialization therefore consists in the sole modification of the hand placement channel while maintaining a realistic avatar pose. With a limited data set containing only the signs in their citation form, we propose to use inverse kinematics techniques on the hand placement channel of those signs to generate their spatialized version (see Figure 8.5). Thus, Inverse Kinematics (IK) applied on a skeleton's arms makes it possible to specify the pose of the arms given the hand placement, that is the position (and, possibly, the orientation) of the wrists.

In the remainder of this section, we present theoretical background on IK, compare

6. This is true also for hand movements and, in a more relative manner as they should stay accurate to be understood, for hand configurations. According to us, this supports the theory on the phonological role of hand placement, configuration and movement as opposed to a phonetic function. Indeed, while phonetics deals with the physical production and can distinguish two placements separated by a few centimeters, phonology is more interested in the meaning and the intent of the signer. With phonology, small variations in the component values will be noticed only if it changes the meaning intended by the signer.

Figure 8.4 – Left: the sign [*PENSER*] (to think). Right: the sign [*CHIEN*] (dog). Both signs have a precise placement in the signing space that cannot be modified.

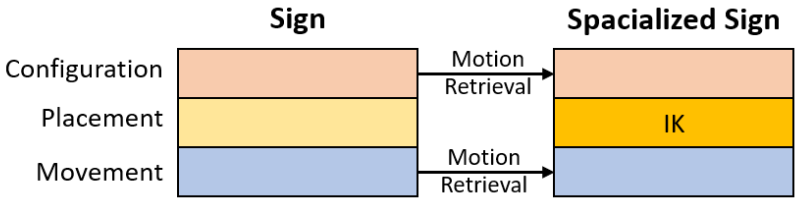

Figure 8.5 – The spatialization principle.

different IK techniques and explain how we take advantage of this technique to implement spatialization.

a) Theoretical Background on Inverse Kinematics An *articulated chain* is made of rigid segments connected by joints which have three Degrees of Freedom (DoF) corresponding to the rotations along the three axes (x, y, z). An articulated chain has a root (the fixed base of the chain) and an end-effector (the extremity of the chain).

Forward Kinematics (FK) consists in animating an articulated chain by specifying the angles of the joints. The rotations of each joint are known and the position of the end-effector of the chain will depend on these rotations.

$$X = F(\theta) \tag{8.1}$$

with F being the forward kinematic function that is used to compute the position of the articulated chain end-effector X with respect to the vector of the angles of the joints θ.

Inverse Kinematics (IK) represents the inverse problem. It is a matter of finding the angles of rotation for each joint so that the end of the articulated chain reaches a given spatial target. We can formulate the problem as:

$$\theta = F^{-1}(X) \tag{8.2}$$

F is a non-linear function: Equation (8.2) may not have a solution or may have several solutions [232]. The IK problem can either be solved by finding an analytical solution or by using an iterative process to obtain an acceptable numerical approximation. However, the likelihood of finding the analytical solutions rapidly decreases with the complexity and number of joints of the articulated chain. Numerical approximations are thus favored in the computer animation community.

A classical solution is to locally linearize the IK problem using the Jacobian matrix J in an iterative process. J is used as a mapping between the small variations of the joint angles ($d\theta$) and the variation of the position of the end-effector of the articulated chain. The inverse of the Jacobian matrix thus allows to measure the impact of a small variation of the position of the end-effector on each angle of the joints simultaneously. Given X_T the target position and X the current position of the end-effector, the desired change in the position of the end-effector is equal to $(X_T - X)$.

Thus, the Jacobian-based IK problem can be formulated as such:

$$d\theta = \alpha J^{-1}(X_T - X) \tag{8.3}$$

with α being the rate of convergence as the process is iterative and must be repeated until $(X_T - X) \approx 0$.

However, the Jacobian is rarely invertible. Indeed, the vector X, which corresponds to the XYZ coordinates of the end-effector has a dimension of 3 while the dimension of $d\theta$ depends on the length and DoF of the articulated chain and is most of the time considerably greater than the dimension of X. As a consequence, the Jacobian which is a $dim(X) \times dim(d\theta)$ matrix is almost never a square matrix. The Equation 8.3 shows therefore an inverse problem that can have no solution, or many solutions, some of which resulting in implausible poses of the skeleton.

A comparison of the Jacobian-based IK methods is given on Appendix B. It presents

the common methods to approximate the inverse of the Jacobian matrix and the performances of these methods.

Another, more recent solution, was proposed by Aristidou et al. called Forward And Backward Reaching Inverse Kinematics (FABRIK) [233]. It presents a completely geometrical approach based on the conservation of the length of the rigid body between two joints of the articulated chain. The process is iterative and composed of two stages. In the first one, the "forward reaching", the end-effector is brought to the target and the configuration of the chain is modified to take this new position into account. As a consequence, the position of the root of the articulated chain is changed. In the second one, the "backward reaching", the root of the articulated chain is brought back to its initial position and the rest of the chain follows. The end-effector gets closer and closer to the target through the iteration of these two stages. FABRIK is an heuristic method that rapidly provides a result by only dealing with the position and not the orientation of the joints. In practice, we found that FABRIK rapidly produces realistic poses.

b) Inverse Kinematics Specification and Application on the Corpus We found that Jacobian-based methods can present too long convergence rates and can display jerky behaviors around singularities (i.e. situations when the end-effector cannot reach the target regardless of the changes in θ, typically in the case of out-of-reach targets). We therefore favored the FABRIK solution in the case of spatialization and, more generally, to deal with hand placement and hand movement modifications (see Sections 8.2.1.2 and 8.2.3.2).

In order to modify the hand placement value of signs, we defined two articulated chains, one for each arm. Each chain has its root at the shoulder and end-effector at the wrist and is controlled by a FABRIK solver (see Figure 8.6).

If no constraint is added to the IK solver, it can produce physiologically improbable or impossible poses. In computer animation, to obtain plausible human poses, it is common to restrict the orientation of each joint of the IK-controlled articulated chain to a validity domain. This approach is known as **clamping** [234]. To clamp the IK chain, biomechanical joint limits must first be determined. Those limits are specific to each joint as each joint of the human body has a different number of DoF and a different range of plausible motion within each DoF. The simplest way to deal with angular limits is to consider each DoF independently using Euler angles. Thus, each DoF has its own axis

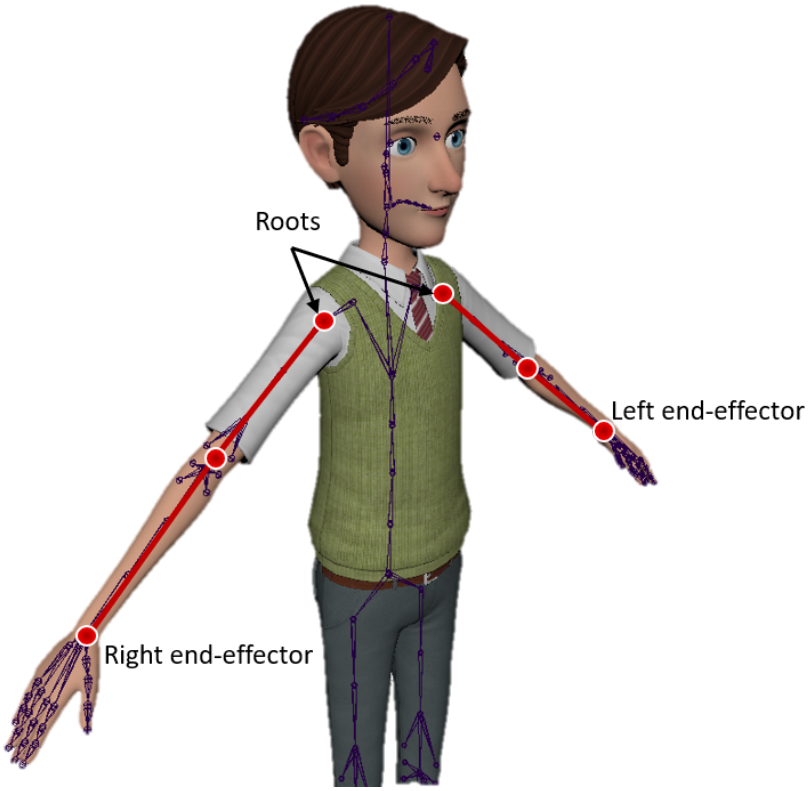

Figure 8.6 – The IK-controlled articulated chains of the two arms.

of rotation as well as two angular limits (θ_{min} and θ_{max}). After each iteration of the IK solver, the new angles on each rotation axis ($\theta_{new} = \theta_{prev} + d\theta$) is compared with the corresponding angular limits. If $\theta_{min} \leq \theta_{new} \leq \theta_{max}$, θ_{new} remains unchanged. Otherwise, it is set to the closest angular limit (i.e. $\theta_{new} = \theta_{min}$ if $\theta_{new} < \theta_{min}$ or $\theta_{new} = \theta_{max}$ if $\theta_{new} > \theta_{max}$).

However, we empirically found that FABRIK generated very few unusual poses even without any additional constraint partially due to the fact that our chain is only composed of three joints. Given a position target for the wrist, FABRIK mainly produces realistic poses of the arm. Nevertheless, we determined bio-mechanical joint limits by parsing our *MoCap* data and setting θ_{min} and θ_{max} with respectively the minimum and maximum angular values reached in the data for each joint. This joint limits could be added to the system if needed.

To change the hand placement channel of chosen signs while maintaining the intra-sign movement and relative placement of the two hands, we specify the targets of the wrists in terms of a translation between the targeted position in the world coordinates and the

actual position of the wrist, still in the world coordinates, when performing the citation form of the sign.

To evaluate the accuracy and realism of our spatialization method, we chose to modify the hand placement value of 3 different signs of our corpus on the basis that those signs were composed of movements of different types. This way, it was possible to determine if the spatialization method accurately maintained the movement and hand configuration components and produced realistic results for different examples. The signs [*MAISON*] (house), [*BOL*] (bowl) and [*TABLE*] were chosen: the sign [*MAISON*] has little hand movement and the hand configuration is stable, [*BOL*] keeps the same hand configuration with a semi-circular movement. Finally, [*TABLE*] is composed of two rectilinear movements with a change in the hand orientation. Figure 8.7 presents visual results of the spatialization of those three signs.

8.2.1.2 Pointing Gesture: Adding a Motion

To designate the subject(s) of an action, to associate virtual objects to $3D$ locations in the signing space (the *locus*), or to refer to those loci, signers use **pointing gestures**. Index pointing gestures are the most common type of pointing gestures and we limited our study to them as we focus on manual features but, with our phonological synthesis engine, we are capable of associating any hand configuration to the pointing gesture (see Section 8.2.2.1).

In pointing gestures, the movement performed by the arm to reach the placement target is important in order to meet the realism requirements. This movement cannot be qualified as "hand movement" in the phonological sense as it is closer to a transient movement with a goal (the pointed object) than an intra-sign hand movement. Moreover, the placement of the hand during the actual pointing is the element that carries the meaning. We therefore believe that pointings should be categorized as hand placement mechanisms and not as movement mechanisms. However, the motion going to the pointing pose, called *reaching* motion, and the motion going back from the pointing pose, called *retraction* motion, must be synthesized anyway. So, in order to synthesize realistic index pointing gestures, we propose to use both **inverse kinematics** and **interpolation** techniques.

In practice, the *pointing pose* for which the pointing gesture reaches its apex is computed by inverse kinematics using FABRIK while the *reaching* and *retraction* motions are synthesized by interpolation.

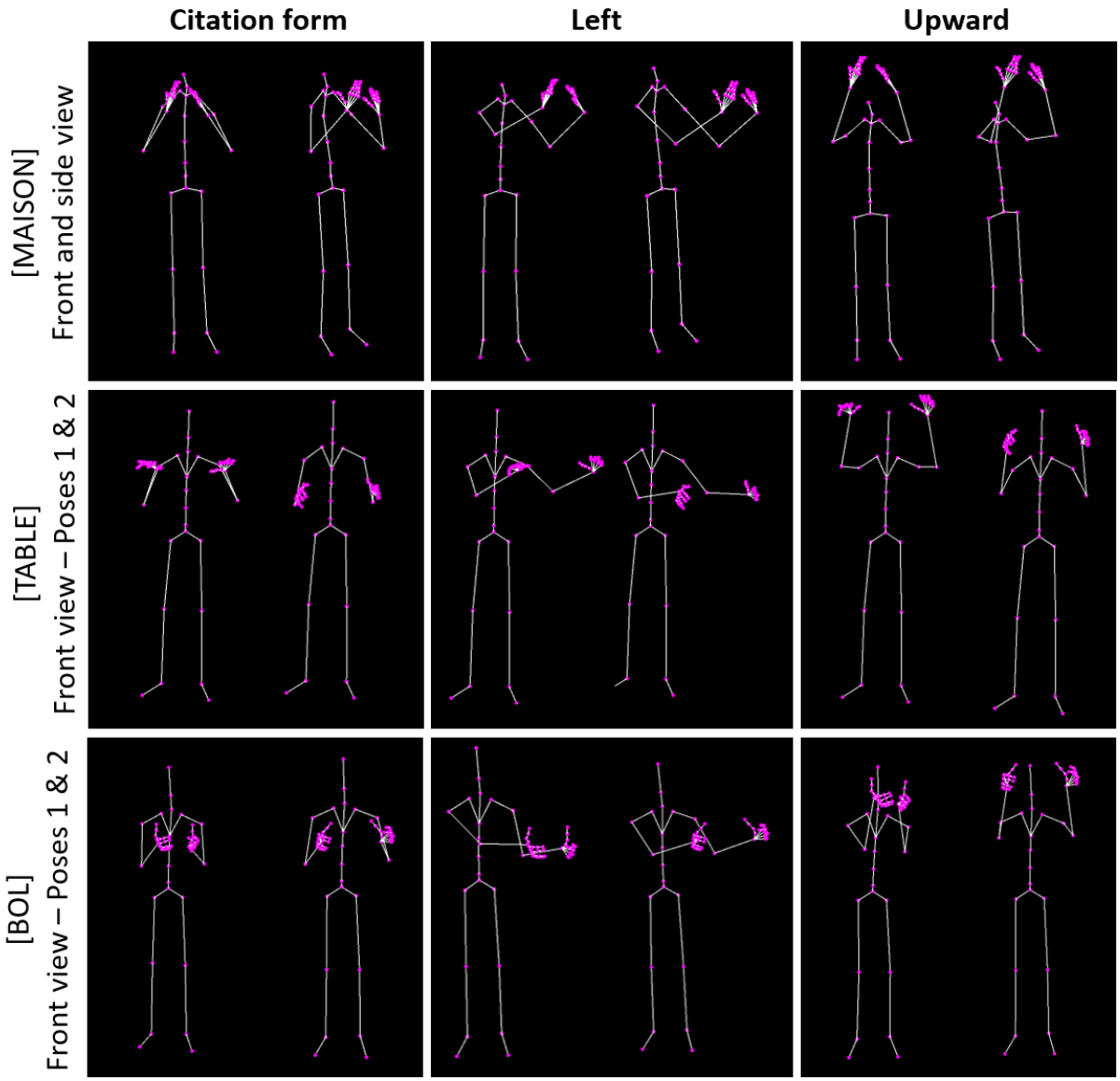

Figure 8.7 – The spatialization of [MAISON] (top row), [TABLE] (middle) and [BOL] (bottom). The citation form of each sign is visible in the images of the first column. The next two present a left and upward spatializations.

To compare the synthesized **pointing poses** with the actual pointing poses present in our database, we retrieved the position of the wrist at the pointing pose in the captured data and fed it to the IK solver as a target. We thus obtained pointing poses (real and synthesized) aiming at the same virtual targets in Figure 8.8. In the real pointing pose, the whole body is more involved than with our IK computation as we only deal with

arm motions. Moreover, we can see that we tolerate a small difference between the actual position of the wrist and the target in the IK model. We believe that this difference is too small to impact the precision of the pointing gestures and adds realism in the sense that two different postures will not result in exactly the same pointing pose [7].

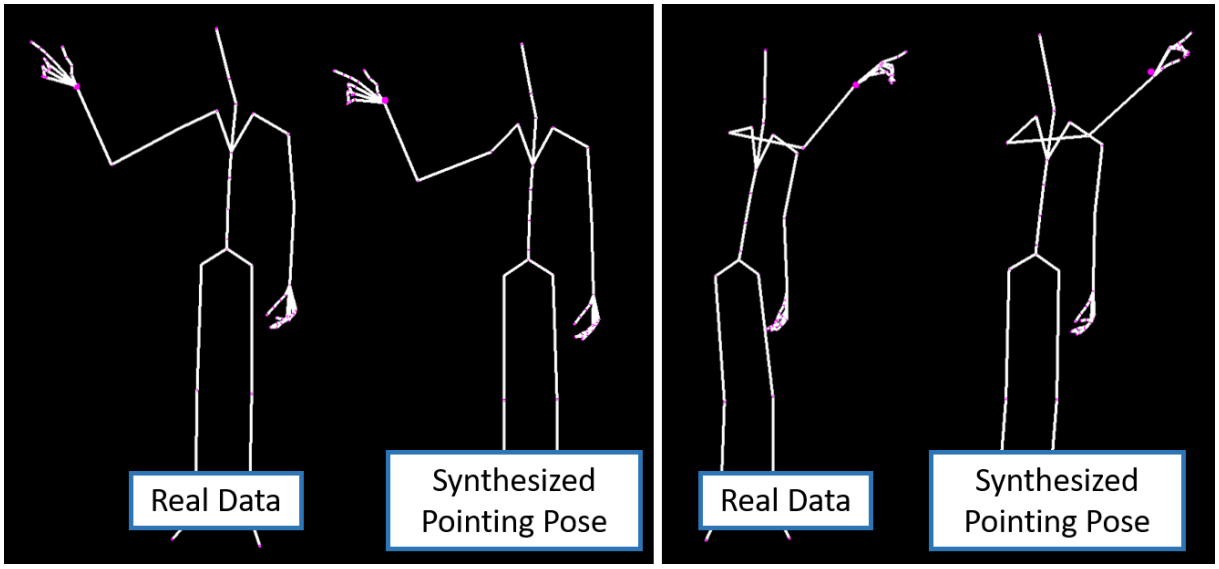

Figure 8.8 – Visual comparison of two real and synthesized pointing poses. The wrist targets are in purple.

To generate the **reaching motion**, before the pointing begins, we interpolate the pose extracted from the data set and the pointing pose produced by inverse kinematics. Then, we produce the **retraction motion** by interpolating the pointing pose with another neutral pose extracted from the data set (see Figure 8.9).

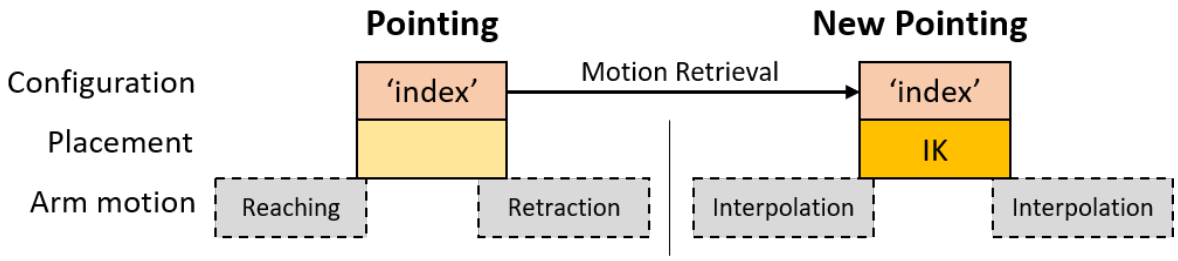

Figure 8.9 – Synthesis of a new pointing motion.

7. It can also be interesting to add noise to achieve a similar effect.

a) Comparison of the Interpolation Techniques We compare here different interpolation techniques to select the most suited to the synthesis of the arm movement for pointing gestures.

Given an initial orientation q_i and a final orientation q_f of a joint j, it is possible to compute the current orientation q with respect to an interpolation weight w in the interval [0,1]. If t_i is the timestamp at the beginning of the transition, t_f the timestamp at the end of the transition and t the current time, w can be defined as:

$$w = \frac{t_i - t}{t_f - t_i} \tag{8.4}$$

Linear interpolation is defined in Equation (8.5). With linear interpolation, the linear velocity during the interpolation is uniform.

$$q_{linear}(w) = (1 - w)q_i + wq_f \tag{8.5}$$

However, linear interpolation can be inaccurate for quaternion computations as it gives a solution positioned on a straight line between q_i and q_f. This solution must be normalized to obtain a unit quaternion. Another more appropriate interpolation exists for the particular application of quaternions: spherical linear interpolation (slerp) is a linear interpolation along a circular arc. It is defined in Equation (8.6) with Ω being the angle between q_i and q_f. In this case, the angular velocity is uniform during the interpolation.

$$q_{slerp}(w) = \frac{sin((1 - w)\Omega)}{sin(\Omega)}q_i + \frac{sin(w\Omega)}{sin(\Omega)}q_f \tag{8.6}$$

However, both those linear interpolations are characterized by a uniform velocity that human motions rarely possess. We therefore propose two complementary interpolations that provide both ease-in and ease-out animations: the cosine (Equation (8.7)) and sigmoid (Equation (8.8)) interpolations.

$$q_{cosine}(w) = (1 - \frac{1 - cos(w\pi)}{2})q_i + \frac{1 - cos(w\pi)}{2}q_f \tag{8.7}$$

$$q_{sigmoid}(w) = (1 - \frac{m}{1 + e^{-\lambda(w-off)}})q_i + (\frac{m}{1 + e^{-\lambda(w-off)}})q_f \tag{8.8}$$

The sigmoid depends on 3 parameters:

— m, value of the sigmoid asymptote when w $\rightarrow +\infty$,

— *off*, an offset that represents the w-value of the inflection point whose y-value is 0.5, and

— λ, slope at the inflection point.

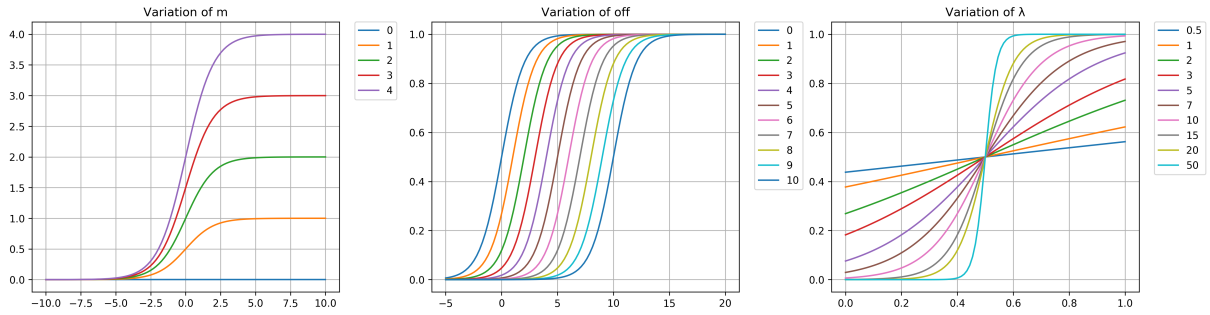

Figure 8.10 – The impact of the different parameters of the sigmoid. To be used in an interpolation task, the sigmoid parameters should be set at $m = 1$, $off = 0.5$ and $\lambda \geq 10$.

The impact of each parameter is shown on Figure 8.10. In order to use the sigmoid function in an interpolation task, it must respect the definition domain for interpolation which is $[0;1] \rightarrow [0;1]$. It means that the sigmoid curve must be completely defined when w varies between 0 and 1. As the maximum value reached must be one, m is fixed at 1. The offset must be at the center of the [0;1]-window for the abscissa, off is thus set at 0.5. Finally, we want that $sigmoid(0) \approx 0$ and $sigmoid(1) \approx 1$. A $\lambda \geq 10$ makes it possible to approximate those values with $\epsilon \leq 0.01$

Figure 8.11 sums up the different interpolation profiles. The linear interpolation is naturally a straight line going from (0,0) to (1,1). The slerp provides an interpolation that begins rapidly and slows at the end (ease-out). The cosine and sigmoid begin and end slowly (ease-in and ease-out) with a more acute slope at the middle of the interpolation. The easings are, however, more pronounced in the sigmoid interpolation (with $\lambda \geq 10$) than in the cosine interpolation.

b) Application on Pointing Gestures To compare the interpolation methods, we extracted three poses in the data set: the two poses where the hand rests near the body before (pose A) and after (pose C) the pointing, and the pointing pose (pose B). We performed the interpolation of the poses A to B and B to C to compose the full pointing gesture. Figure 8.12 shows the kinematic profiles for the slerp, cosine and sigmoid interpolations. The sigmoid interpolation gives the kinematics profiles the closest to the ground truth. The sigmoid interpolation was thus used with the IK-generated pointing pose to

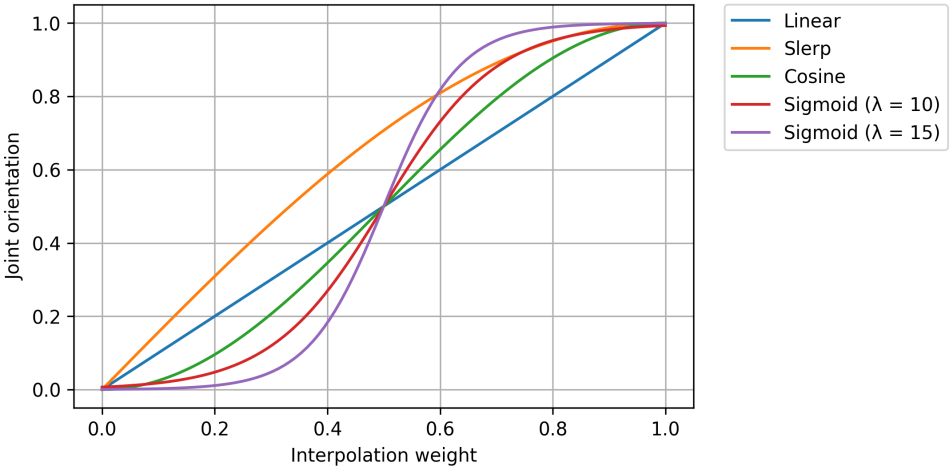

Figure 8.11 – The different interpolation profiles with respect to the interpolation weight.

create pointing motions.

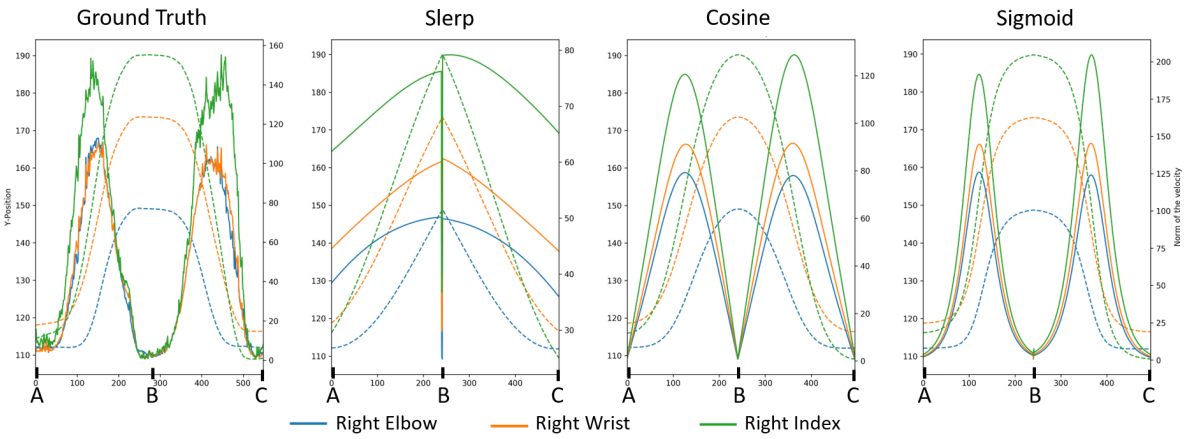

Figure 8.12 – The height position (dotted line) and norm of the velocity (continuous line) during one upward pointing. The real data is on the left and the different interpolated motions are shown from left to right: slerp, cosine and sigmoid ($\lambda = 10$) interpolations. A, B and C are the timestamps corresponding to the moment when the hand rests near the body before (A) and after (C) the pointing (B).

Figure 8.13 shows the visual results on an avatar for the real data, the linear and the sigmoid interpolation. Again, we can see that, in the real data, the whole body is involved in the motion which is not the case with our technique. In terms of arm motion, the dynamics of the real data is visually closer to the dynamics of the sigmoid interpolation (slow beginning, acceleration and slow down) than to the linear interpolation (constant

speed). It is especially visible on the fourth image of each sequence: for the real data and for the sigmoid, the wrist has almost reached the target while, in the linear case, the arm is still midway.

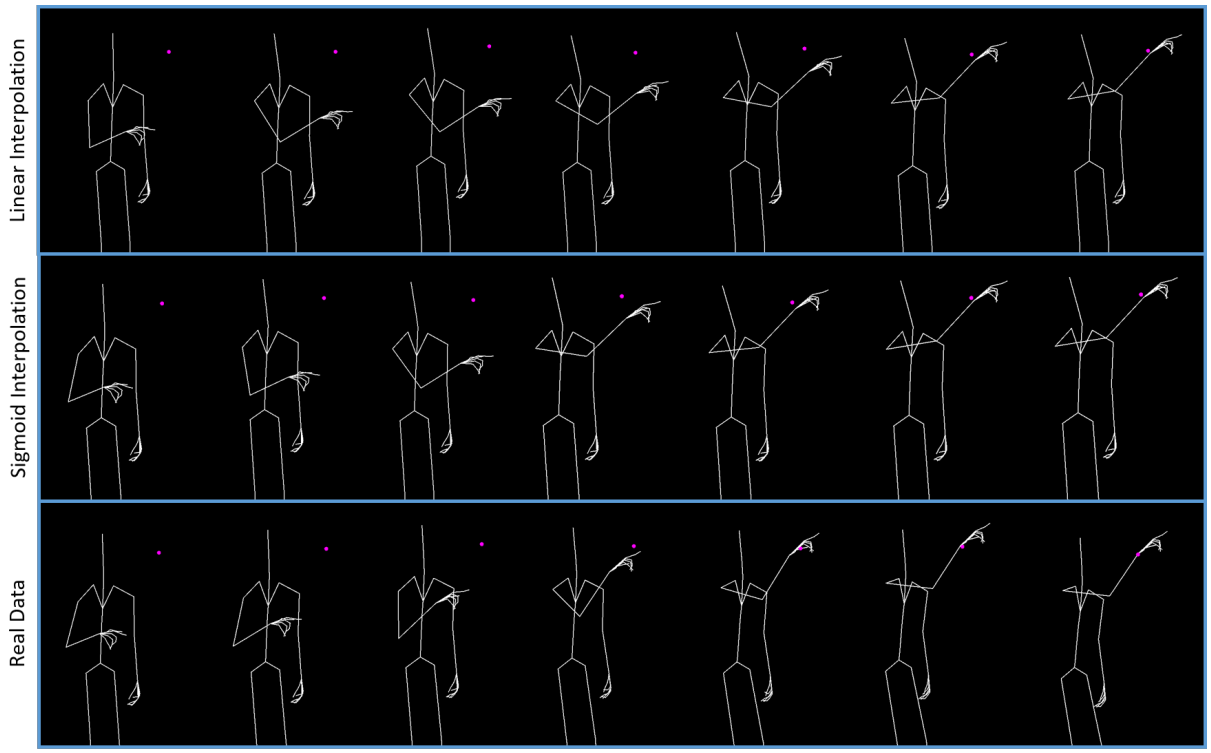

Figure 8.13 – Pointing motion: real (bottom line) and synthesized with the sigmoid (middle) and linear (top) interpolations.

8.2.2 Hand Configuration Mechanisms

The hand configuration corresponds to the overall shape of the hand. Concretely, at a computer animation level, it refers to the set of orientations of the finger joints. Many linguists try to establish a list of the hand configurations used in a specific sign language [1], [3], [8]. The length of this list usually varies from 30 to 80 items depending on the coarseness of the item. In either case, the hand configurations are seen as limited in numbers. It is then possible to capture every configuration at a low cost to have an exhaustive supply of sign language specific handshapes to be used in further synthesis work. The *LSF-ANIMAL* corpus contains 48 carefully chosen hand configurations. In this section, we propose to generate new linguistically relevant content by replacing the hand

configurations in specific signs (Section 8.2.2.1) and by synthesizing realistic transitions between the hand configurations (Section 8.2.2.2).

8.2.2.1 Derivative Base, Specifiers & Proforms: Replacing the Hand Configurations

The Hand Configuration (HC) is a relatively stable component inside a sign. Its value rarely varies during the production of individual signs and can generally be labeled unambiguously following pre-established categories. We thus believe that new SL content can be synthesized by modifying the values taken by the HC inside isolated signs.

The signs that can be created this way are limited by the initial database and by the language itself. Indeed, in the case of the placement component, it is possible to create generic content regardless of the content of the database: a pointing gesture cannot be incorrect in itself as long as it is not physically extravagant. In the case of the HC, it is necessary to have an *a priori* knowledge of the language: all the combinations sign/configuration do not exist. A study of the content of the database at a semantic level and knowledge of the language are necessary to find the signs that can be added to it by replacing the HC. It is then possible to increase the vocabulary of a database by identifying these signs and modifying their configuration.

Three LSF mechanisms seem to be particularly suited to such transformation: some specific **derivative bases**, **shape specifiers** and **proforms**. It is assumed, in those three cases, that the nature of the HC is discriminative and that new signs can be created by changing it. This principle is illustrated in Figure 8.14.

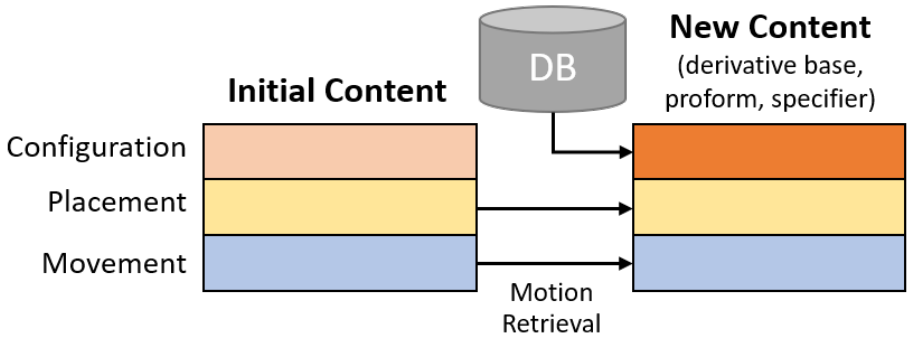

Figure 8.14 – Replacement of the hand configuration: given an initial sign, it is possible to create new signs by extracting (using motion retrieval from a database DB) and replacing the hand configuration.

A **derivative base** designates a set of signs with the same value for one phonological component which have a unity of meaning. For example, many signs with a placement on the side of the forehead will have a meaning related to a psychic activity (e.g., [*PENSER*] (to think) or [*REVER*] (to dream), see Figure 3.5). In the case of HC replacement, we only target the derivative base with the same motion and placement. In *LSF-ANIMAL*, we have such derivative bases. We choose to do our experiment on the snail/slug base: the sign [*ESCARGOT*] (snail) and [*LIMACE*] (slug) are part of the same derivative base regrouping slow rampant animals. The signs in this derivative base have the same slow yawing motion and the dominant hand is placed slightly above the non-dominant hand. They only differ by the hand configurations: configuration of the 'H' or 'Y' for [*ESCARGOT*] and 'U' for [*LIMACE*].

For our experiments, we replaced the 'Y' hand configuration in [*ESCARGOT*] to create an 'H' configuration snail and a 'U' configuration slug. The 'H' and 'U' configurations originate from the isolated HC sequences of the *LSF-ANIMAL* corpus. Concretely, we overwrite the 'Y' configuration of the skeleton during the realization of the snail sign with the orientation of the finger joints for the 'H' and 'U' configurations. The results are visible on figure 8.15.

Figure 8.15 – Replacement of the 'Y' configuration of the sign [*ESCARGOT*] (left) by the 'H' configuration to create a different sign [*ESCARGOT*] (middle) and the sign [*LIMACE*] (right).

In **shape specifiers**, the HC will vary depending on the object of the action. In the well-known example of [*DONNER*] (to give), different hand configurations are performed depending on the given objects: in "I give you a glass", the configuration used is the 'C' representing cylindrical objects while, in "I give you a coin", it's the configuration of the

'closed_pliers' representing small circular objects that is used [8] (see Fig. 8.16).

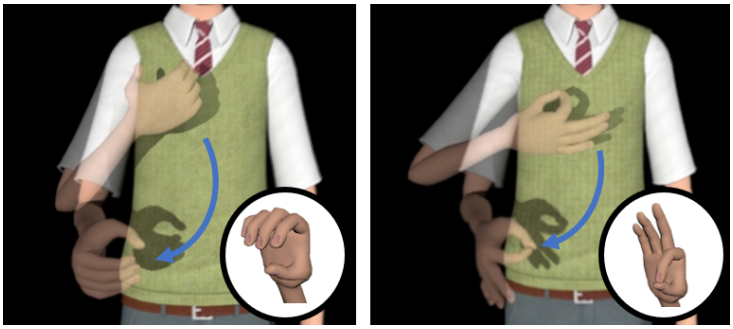

Figure 8.16 – Shape specifiers for the sign [*DONNER*]. Left: "I give you a glass" ('C' configuration). Right: "I give you a coin" ('closed_pliers' configuration).

In the *LSF-ANIMAL* corpus, we have an example of shape specifiers in the gait of our different animals. A character's gait can be precisely described by using the hands and arms of the signer to represent the legs of the described thing. Gaits are both the results of shape specifiers and role shift. Indeed, different animal gaits can be generated by changing the hand configurations (e.g., a cat's walk can be changed to a lion's walk by changing the 'U' to a '5_folded' configuration): here, the hand configuration is the shape specifier. However, in the case of gaits, hand configuration is not the only thing that varies: the style of the arm movement is determinant when describing a character's type of walk: this is the expression of the role shift part that is not treated in this thesis. We tested HC replacement for three other gaits: we replaced the 'U' configuration of the cat's gait by a '3' configuration to obtain a chicken's gait, a 'S' configuration to obtain a cow's gait and a '5_folded' configuration to obtain the lion's walk (see results on Figure 8.17).

Proforms make it possible to describe situations by using the combination of one or two HC (the subjects and objects of the situation) and specific motion paths (the paths of the subjects/objects) or static locations (showing the locations of entities with respect to each other). The hand configurations taken in proforms are extremely codified: a flat hand represents a car, a closed fist and raised index a standing person, a raised thumb a cyclist, etc. Changing the HC in proforms thus leads to change the nature of the subject(s)/object(s) of the situations. We captured a few proforms in the *LSF-ANIMAL* corpus. The sequences describe cars, cyclists and people moving side by side or

8. Depending on the source, this type of example can be given to explain the concept of "shape specifiers" or "proforms". We have chosen to consider it as a shape specifier to distinguish it from proforms which, in the context of this thesis, are mainly used to describe situations.

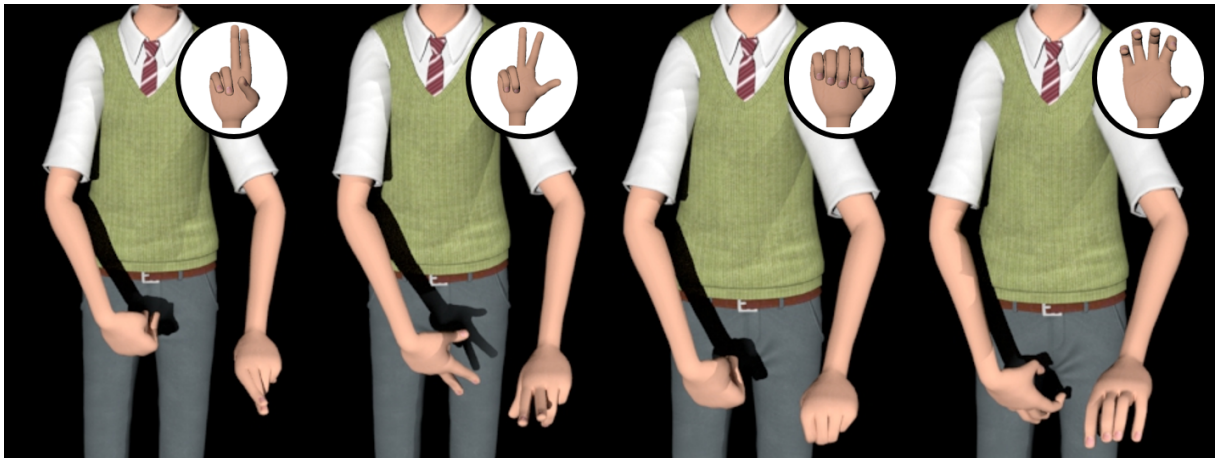

Figure 8.17 – Replacement of HC in gaits: the original motion is the cat's walk (left) with the 'U' configuration, the three others are synthesized (from left to right: chicken, cow and lion's walk).

crossing each others. By changing the hand configuration, we constructed new situations in which the subjects of the actions were different. Changing the situation itself would mean modifying the motion paths; this is proposed in Section 8.2.3.2.

8.2.2.2 Dactylology: Linking the Configurations

The previous section only deals with static hand configurations. However, some signs and SL mechanisms require a finer control on the transformations of the hand configuration over time. Some signs, in particular, see their configuration vary during their realization. The [*CHAT*] (cat) sign, for example, starts with a '3_folded' and ends with a '3' configuration (see Figure 8.18). For such signs, it is necessary to manage the profile and the timing of the transition from one configuration to another in a way that is both correct and realistic.

Considering the 48 hand configurations defined in the *LSF-ANIMAL* corpus, the number of possible transitions is equal to $48 * 47 = 2256$ (with two HC, there are 2 possibilities: HC1 $\rightarrow$ HC2, HC2 $\rightarrow$ HC1). The capture of each possible transition represents an enormous and unrewarding work. However, with only few instances of signs with a HC variation to study, we propose to induce a model to synthesize transitions from one HC to another. In the case of the [*CHAT*] (cat) sign and other signs with HC variations, the configuration is not the only parameter that varies. The synthesis of such signs requires to process the channel of the configuration but also the other channels. To remove any

Figure 8.18 – The sign for [*CHAT*] (cat) begins with one configuration ('3_folded') and ends with another ('3').

bias due to the consideration of the other channels, we choose to study only two types of SL mechanisms: dactylology and signs derived from it. As only the hand configuration varies in dactylological signs, they are a great application case to study this phonological component without being impacted by the other components.

Dactylology, also called fingerspelling, is the process of spelling a word by using a dactylological alphabet. The French dactylological alphabet consists of 19 hand configurations that – when held in certain orientations and/or are produced with certain movements – represent the 26 letters of the French alphabet. Fingerspelling is used in many sign languages to spell proper nouns like names of people (see Figure 8.19) or places before assigning them a "sign name", or to spell words whose sign is unknown by the signer or by the interlocutor. In LSF, the spelling is mainly performed with the dominant hand in the neutral space: the placement component is thus fixed, orientation modifications and movements are minimal.

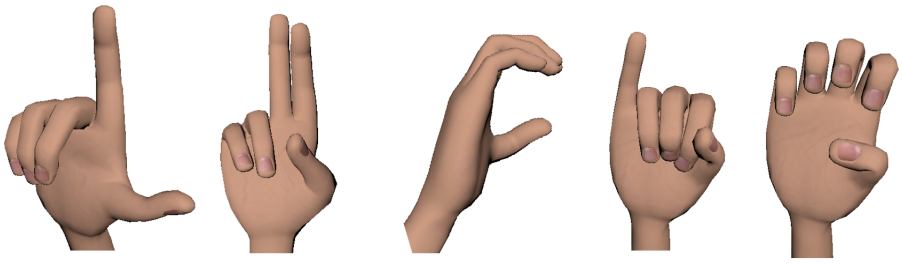

Figure 8.19 – The sign for [LUCIE] using the fingerspelling alphabet.

A fingerspelled word is not a sign in itself. However, some signs are **derived from fingerspelled words** like [*OK*] (hand configuration of the 'O' followed by the 'K'), [*WEEK-END*] ('W' followed by 'E' with a rectilinear movement) in LSF, or the sign [*LSF*] itself ('L', 'S', 'F' configurations with a descending and then ascending rectilinear hand movement, see Figure 3.9). In this section, we focus only on the realistic synthesis of the transition between two HC. The sign [*OK*] and only the hand configurations of the signs such as [*WEEK-END*] or [*LSF*] are thus considered. The challenges of synchronizing a change of hand configuration to a movement of the hand are treated in Section 8.2.4.

In the remainder of this section, we study the hand configurations of 7 signs derived from dactylology and 1 non-dactylological sign with a variation of the configuration (see Table 8.3).

Signs	**Translation**	**Hand configurations**
[*OK*]	ok	'O' → 'K'
[*OR*]	gold	'O' → 'R'
[*WEEK-END*]	week-end	'W' → 'E'
[*SALON*]	living room	'O' → 'C' (not dactylological)
[*LSF*]	LSF	'L' → 'S' → 'F'
[*FIN*]	end	'F' → 'I' → 'N'
[*DODO*]	nap	'D' → 'O' → 'D' → 'O'
[*LÀ*]	to be present	'L' → 'A' → 'L' → 'A'

Table 8.3 – Signs of interest for the synthesis of hand configuration transitions (sorted by the number of different hand configurations).

To link two HC, we propose to use static hand configurations present in the isolated HC sequences of the *LSF-ANIMAL* corpus and to synthesize the transition procedurally. It is thus a matter of synthesizing the variation of the hand configuration from a fixed 1-frame initial configuration and a 1-frame final configuration with optionally 1-frame intermediate configurations (see Figure 8.20). There are two alternatives for synthesizing a transition from one hand configuration to the next, namely interpolation and inverse kinematics. They are discussed hereafter.

a) Interpolation Interpolation is the most common way to synthesize intermediary values between two points. Depending on the type of interpolation, the values of those intermediary points will vary. For hand configuration transitions, the interpolation, regardless of its type, is applied to the quaternions that control the orientation of the finger joints. The interpolation duration is set roughly following the values of the ground truth.

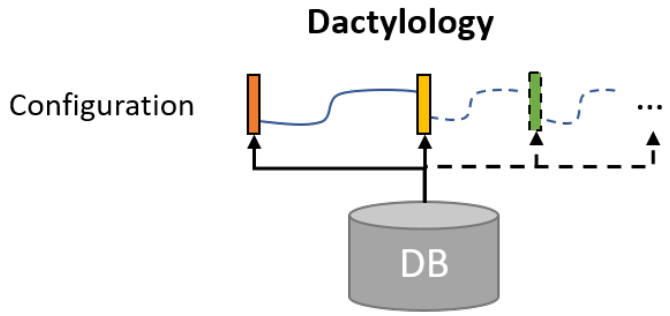

Figure 8.20 – Principle of the synthesis of dactylology phenomena: the isolated HC are extracted from the database (DB) and the transition between the HC are synthesized.

The four types of interpolation presented in Section 8.2.1.2 were applied on the 8 examples of signs derived from dactylology. Figure 8.21 presents the norm of the position and of the velocity of the tip of the fingers for the dactylological sign [*DODO*] for the ground truth and using the different interpolations. Whether for the velocity or the position of the fingers over time, the sigmoid and cosine interpolation seem to give the results the most similar to the ground truth.

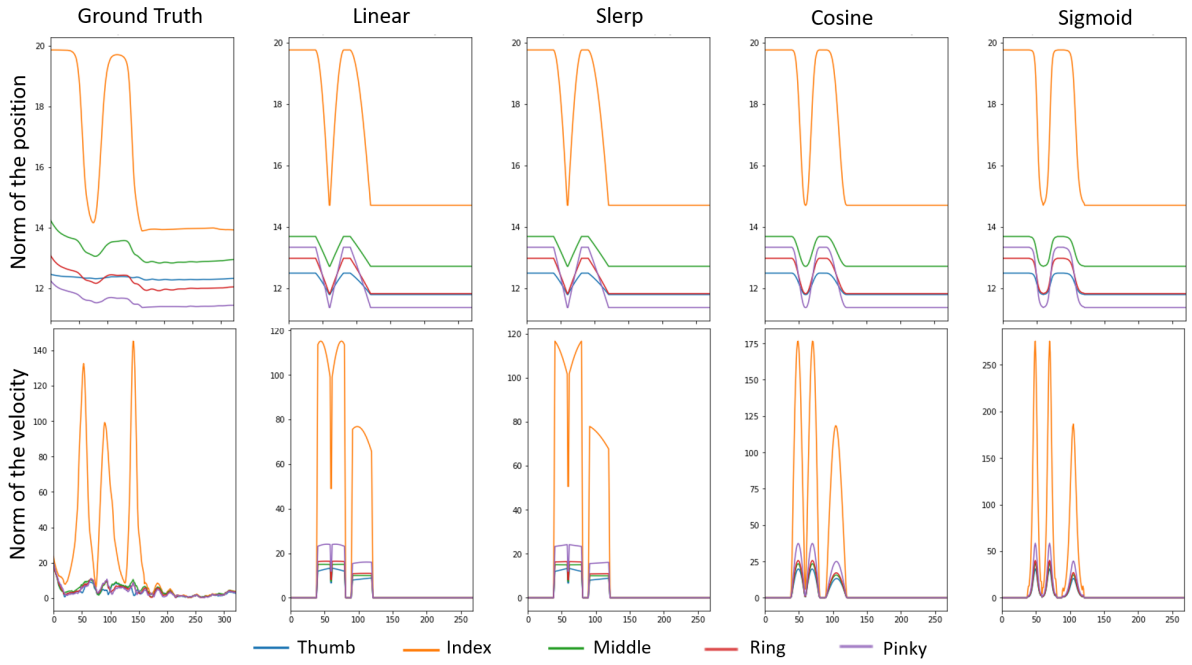

Figure 8.21 – The position (first row) and norm of the velocity (second row) of the tip of the fingers for the dactylological sign [*DODO*]. The columns represent, from left to right, the curves of the ground truth, linear, slerp, cosine and sigmoid ($\lambda = 10$) interpolations.

b) Inverse Kinematics Another way to synthesize the hand configurations and the transition from one to another using real data is to implement an inverse kinematics solver taking the targets extracted from our *MoCap* database. For hand configurations, these targets correspond to the positions of the fingertips relative to the position of the wrist. Since the hand configurations last more than one frame and are present several times in the data base, the fingertip positions of each hand configuration have been averaged.

The inverse kinematics algorithm used is Damped Least Square (DLS) [235] (see Appendix B for the comparison of the Jacobian-based inverse kinematics techniques). For this application, the advantage of the DLS inverse kinematics is that the algorithm works correctly even when the target is out of reach and creates a smooth motion unlike FABRIK for which only the final pose is important.

The DLS inverse kinematics solver is an iterative process whose equation can be written as:

$$d\theta = \alpha(J^T J + \lambda^2 I)^{-1} J^T (X_T - X) \tag{8.9}$$

with $d\theta$ corresponding to a small variation of the angles of the articulated chain, J^T the Jacobian transpose matrix, I the identity matrix, α the gain, λ the damping factor, X_T the targeted position of the end-effector and X the current position of the end-effector.

The damping factor (λ) used is 1 and the gain (α) is 0.1 to introduce a delay in reaching the targets. The DLS IK is thus used both to obtain the hand configuration pose and to create the transition motion. Figure 8.22 shows the IK-controlled articulated chains of the hand.

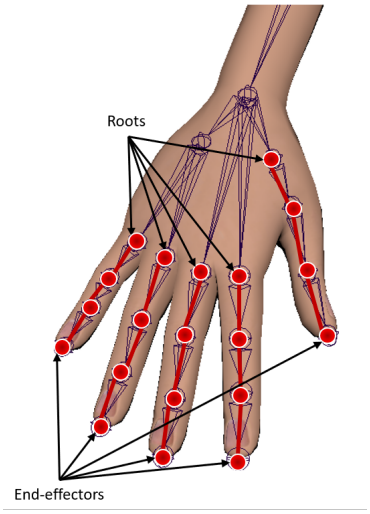

Figure 8.22 – An articulated chain per finger has been defined.

On Figure 8.23, the position and velocity of the fingertips for the ground truth, the sigmoid interpolation and the DLS IK algorithms are compared for 3 signs. IK results seem less realistic than the interpolation but has the advantage of not needing to give a transition duration to the algorithms. In return, the temporal control is very limited and cannot be tuned for each specific sign.

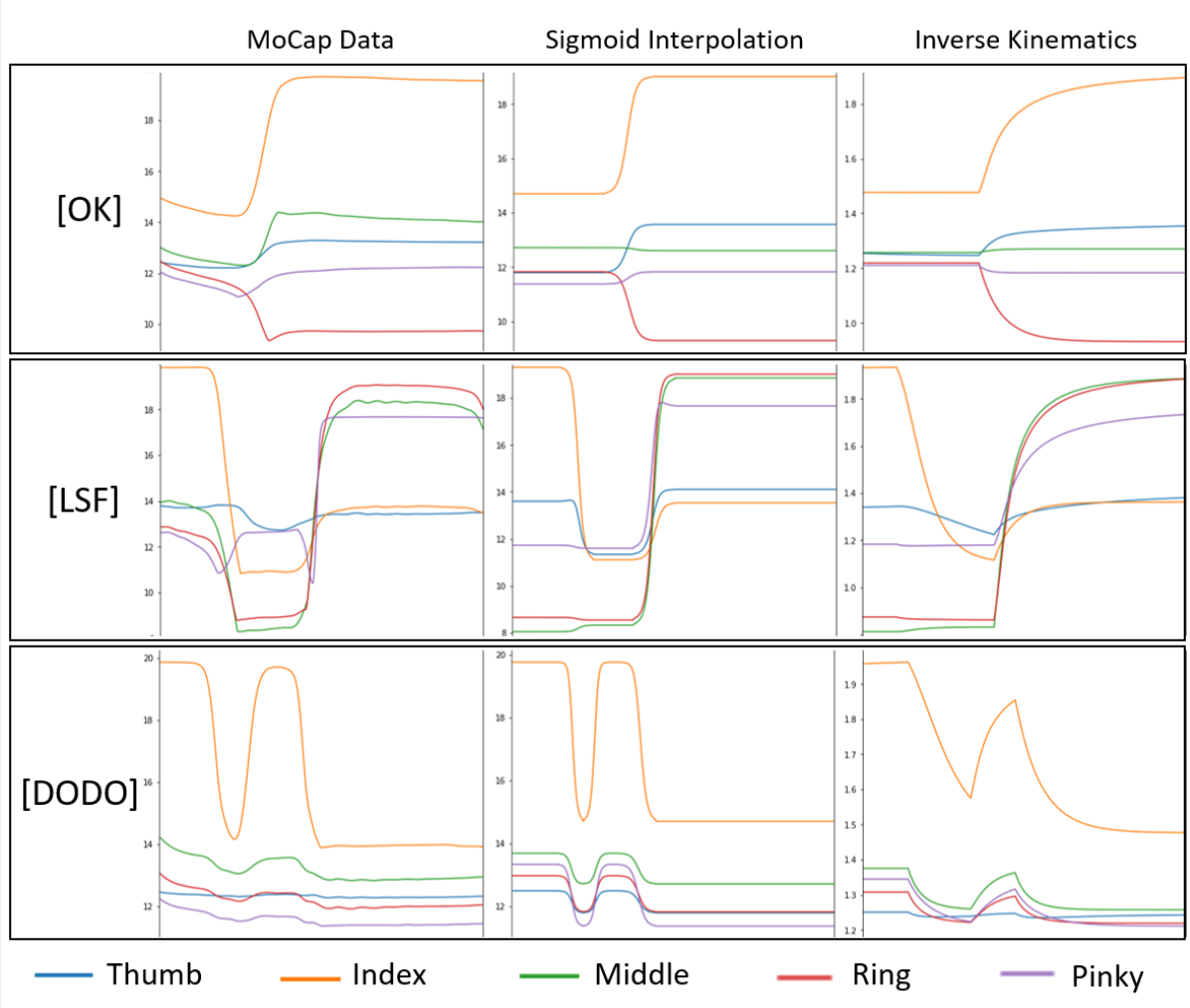

Figure 8.23 – The position of the tip of the fingers for the dactylological signs [*OR*], [*DODO*] and [*LSF*]. The columns represent, from left to right, the curves of the ground truth, sigmoid interpolation ($\lambda = 10$) and the synthesis using the DLS IK solver.

8.2.3 Hand Movement Mechanisms

Hand movement corresponds to the overall trajectory of the wrist during a sign and, sometimes, to secondary movements (i.e. small movements of the fingers) also performed during a sign. In this section, we only study the former as we consider the latter to be closer of a transformation of the hand configuration than of a movement phenomenon. Hand movement is a dynamic component of SL as opposed to the static components that are hand configuration or hand placement [9]. The movements of the wrists can be seen as time series as they correspond to a trajectory over time.

Furthermore, hand movement can have different roles depending on the linguistic level considered. At a phonological level, hand movement is an *articulatory motion* that can be modified through different time series operators (Section 8.2.3.1). At a syntactic level, in the description of situations using proforms, with indicating verbs or with size and shape specifiers, hand movements can be seen as *motion paths*, trajectories that can be controlled through inverse kinematics methods (Section 8.2.3.2).

8.2.3.1 Modifying the Articulatory Motion

We define the articulatory motion as the hand movement performed as part of an isolated sign (unlike inflection mechanisms involving the hand movement, like *motion paths* in the description of situation using proforms, or *iconic flexions* to precisely describe the motion of an object). The articulatory motion makes it possible to distinguish two signs, [*FILM*] (movie) and [*CAROTTE*] (carrot), for example, have the same hand placement and configuration. Only the motion, circular for [*FILM*] and rectilinear and repeated for [*CAROTTE*], differentiate the two signs (see Figure 3.6).

By modifying the articulatory motion of the hands, it is therefore possible to create new signs absent from the original database. To that end, we propose to edit the hand movements when they act as articulatory motions by treating them as time series. Indeed, time plays a predominant role in the production of SL as a gesture can be seen as a series of spatiotemporal targets to be reached. If the temporality is changed, the sign will be modified: it will lose its meaning or gain in subtlety. As in oral languages, an SL signer can play with the rhythm of his signs and statements to create effects or new signs. The execution of these mechanisms for the animation of an avatar involves motion editing processes and, in this case, the movement is seen as a time series to alter. A sign will take

9. Considering placement zones and not exact numerical values.

on a different meaning or style through temporal inversion, mirroring or repetition of the hand movement.

a) Temporal Inversion In LSF, some signs have an opposite that can be achieved by temporally reversing the movement. For example, the sign for [*CLAIR*] (light) can be changed to [*FONCÉ*] (dark) if it is played backwards (see visual results on Figure 8.24). The same goes for [*AIMER*]/[*NE PAS AIMER*] (like/don't like) or [*DONNER*]/[*PRENDRE*] (give/take) [236]. The presence of signs with potential opposite by inversion in the initial base is very interesting for the enrichment of the base.

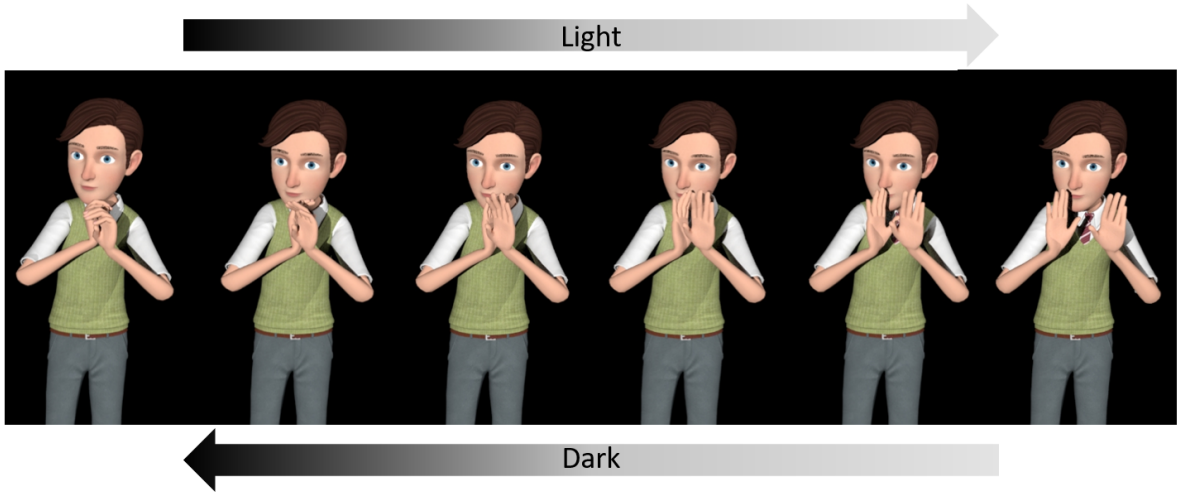

Figure 8.24 – Synthesis of the sign [*FONCÉ*] (dark) by temporally inverting the sign [*CLAIR*] (light).

b) Swap and Mirror In order to increase the number of variations in the database, it is also possible to transform the signs performed with the right hand (resp. the left hand) into left-handed signs (resp. right-handed signs). A *hand swap* operator makes it possible to symmetrize the hand movement performed by the dominant and non-dominant hands. This swap operator is also useful for utterance synthesis and, particularly, for the parallel production of two signs as signs can be done with either hand, in particular if the preferred hand is "busy" doing another sign (see Section 8.3.2). In addition, some signs can be done with either one or two hands (e.g., the sign [WEEK-END] or [FIN] (*end*) in LSF). Generating the two-handed version from the single handed one can be done with the *mirror* operator by copying the movement of one arm on the other arm.

More precisely, for each joint of one specified arm i_{right} [10], the mirror operator assigns a position of the same joint on the other arm i_{left} so that:

$$\begin{pmatrix} x_{i_{left}} \\ y_{i_{left}} \\ z_{i_{left}} \end{pmatrix} = \begin{pmatrix} 2 * x_{hips} - x_{i_{right}} \\ y_{i_{right}} \\ z_{i_{right}} \end{pmatrix}$$

In the case of the swap operator, the two hand movements are exchanged using the same principle. The swapped and mirrored signs thus generated keep the realism of the captured motions. Examples of application of the hand swap and mirror operators are shown in Figure 8.25.

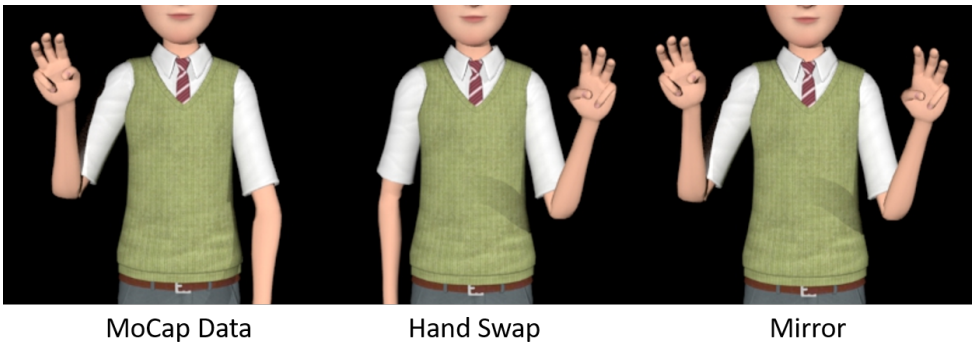

Figure 8.25 – Results of the application of the swap (middle) and mirror (right) operators for the sign [WEEK-END]. The original version of the sign is on the left.

c) Repetition A repetition in the movement will mark the repetition of an event or the notion of weariness. The sentence "I work all the time, I'm tired of it" can potentially be executed with only the sign [TRAVAIL] (*work*) repeated many times. The amplitude and number of repetitions will change according to the desired meaning [11]. Repetition is also a means of constructing new signs. For instance, the sign [PUNIR] (*punish*) repeated twice results in the sign [TRAVAIL] (*work*) (see Figure 8.26). In order to have a smooth transition between two repetitions of the same motion, a short slerp interpolation is added between the repeated instances.

The swap, mirror, temporal inversion and repetition operators presented in this section can be combined to create new content (see Figure 8.27).

10. This is also true when exchanging the right and left arms.
11. Not to mention the overall attitude and facial expressions that are not part of this thesis.

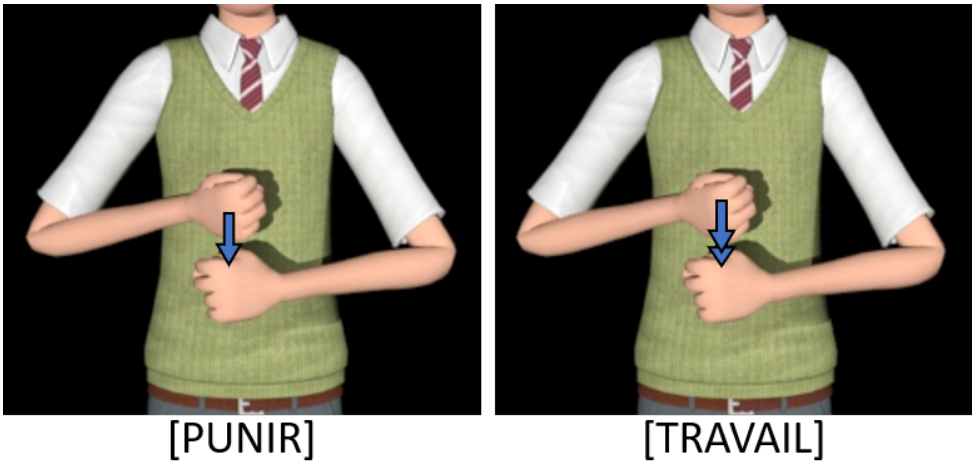

Figure 8.26 – The sign [PUNIR] (left) repeated twice gives the sign [TRAVAIL] (right).

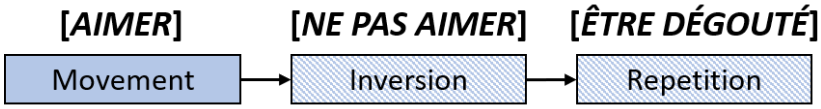

Figure 8.27 – [ÊTRE DÉGOUTÉ] (*disgusted*) from [AIMER] (*like*): inversion then repetition.

8.2.3.2 Perspectives - Indicating Verbs, Proforms & Size Specifiers: Creating New Motion Paths by IK

In the previous section, we directly edited the *MoCap* data to create new content based on signal processing operators (repetition, inversion, etc.). In this section, we propose to generate new inflected signs by creating new motion paths. The signs considered are signs in which the trajectory of the hand changes according to the relationship between the entities, the motions, or the size of the entities themselves. We did not implement the proposed solutions but we believe that a dedicated section can be interesting to show the similarities between the three mechanisms presented hereafter in terms of computer animation techniques.

In **indicating verbs**, the hand trajectory will vary according to the identity and position of the subject and object of the action. The hand movement corresponding to the verb [*DONNER*](to give) in the sentence "I give him" will be performed from the signer ("I") to his/her side ("him") while the same verb [*DONNER*] in the sentence "you give me" will begin in front of the signer ("You") and finish near the signer ("me") (see

Figure 3.7).

In situation description using **proforms**, the hand trajectory will copy the trajectory of the described entities. If the signer wants to communicate that two persons walked side by side in a straight road, he/she will use two fingers to represent the two persons and a rectilinear hand movement to express their walk. If the road becomes a sinuous path, the hand movement will change to a serpentine trajectory.

Size specifiers use the amplitude of the motion to specify the size of objects. In those specifiers, the modification of the articulatory motion of the sign in its citation form permits to add information about the size of the described entity.

In those three cases, the hand movement itself is highly iconic and possesses a meaning. Moreover, the hand movement is almost independent from the other components: a change in the hand movement (and in the amplitude of the movement in the particular case of size specifiers) without changing the other components will, most of the time, result in a correct sign or utterance with a different meaning (this is rarely the case for articulatory motions that, if changed in a sign, will often deprive the sign of its meaning). In addition, in those three cases, and particularly with proforms, the number of possible hand movements is infinite. This huge variety of movements cannot be fully captured and cannot be synthesized by only transforming existing movements.

Given a set of spatiotemporal targets specifying a motion path, we propose to use an inverse kinematics solver to compute the successive poses of the avatar's skeleton. This constitutes a hybrid synthesis system where the hand movement is fully synthesized while the hand configurations and orientations used correspond to data present in the captured motions (see principle in Figure 8.28). In the particular case of size specifiers, the motion paths generated must keep the same trajectory as the original movement of the sign in its citation form. Scaling the original movement to the appropriate size by using the captured motion as reference and applying IK to the resulting motion paths can permit generating such mechanisms. We could not implement and test this hybrid synthesis principle but we believe that it is a very interesting perspective for future synthesis work.

8.2.4 Synchronizing the channels

In the whole Section 8.2, we have treated the phonological components independently from each other and we have found solutions to change the value of each component without looking at the other components. However, a sign is the result of values taken

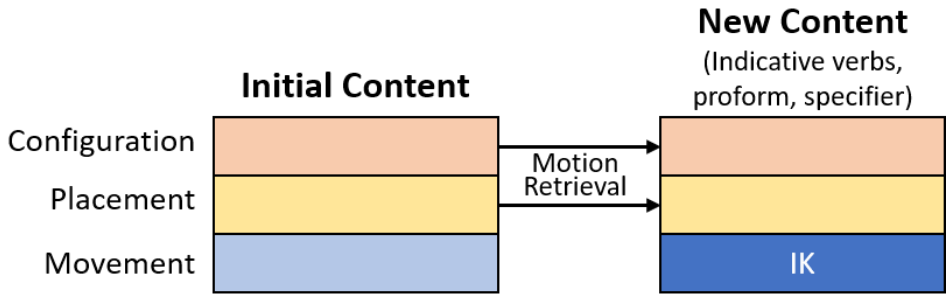

Figure 8.28 – The creation of new content thanks to the synthesis of new motion paths by inverse kinematics.

simultaneously (or with a slight offset) by all the components. We can therefore try to combine the techniques we have presented in the previous sections in order to build a sign from isolated values on the **three** phonological components and not on only one of these components. However, in order to be able to generate correct signs, it is necessary to have an idea of the inter-component timing when constructing a sign, i.e. to understand how the components are synchronized with each other (see Figure 8.29).

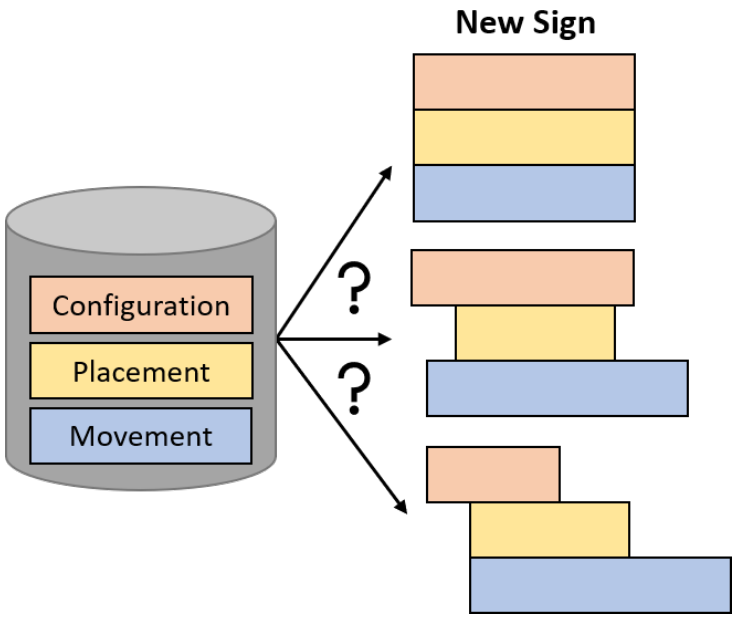

Figure 8.29 – The recombination problem: how to synchronize the different hand components in the case of signs built as the result of the combination of the three manual components ?

For this purpose, we visually analyzed the kinematic trends of the wrist movement

(corresponding to the hand placement and movement) and of the fingertips (to visualize the variations of hand configurations) for selected signs belonging to the database. The aim is to understand how the hand configuration, movement and placement components are temporally organized in the ground truth so as to reproduce, or approach, the same timings in the synthesis. We believe that this analysis is not only interesting for synthesis purposes. It gives us information on the cognitive aspects of sign production in LSF, and SL in general.

In Figure 8.30, we can see, for different signs, the norm of the position of the wrist in relation to the hips (red curve) as well as the norm of the position of the tip of the index (or ring finger) in relation to the position of the wrist (blue curve). The first curve provides a view of the wrist movement and thus of the "hand movement" and "hand placement" components. The second curve gives an overview of the sequence of hand configurations.

In these figures, the wrist velocity minima are indicated by letters (**A**, **B**, **C**...). These minima correspond to decelerations or stops indicating hand locations [12] (i.e. the beginning and end of a movement). **A** thus represents the starting location of the sign and the beginning of the intra-sign movement, i.e. the hand movement, if it exists. **B** thus marks the end of the first intra-sign movement (if it exists). The last letter marks the beginning of the transition to the next sign. In addition, the blue areas correspond to time intervals that mark the moments when the hand configuration has reached a stable state. The type of configuration is specified in the blue zones. Finally, the area delimited by the black dotted lines represents the segment labeled as the produced sign. The segment boundaries are the result of the automatic refinement of the manual segmentation described in Section 7.3.

For the purpose of motion synthesis, we analyzed 6 signs: *[LSF]* (which contains a hand movement and 3 hand configurations), *[WEEK-END]* (a hand movement and 2 hand configurations), *[CHAT]* (cat) (a hand movement and 2 hand configurations), *[SALON]* (living-room) (a hand movement and 2 hand configurations), *[POURQUOI ?]* (why?) (a hand movement and 1 hand configuration) and *[CANARD]* (duck) (no hand movement and 2 repeated hand configurations mimicking the motions of the duck's beak).

On these figures, we can make at least 4 different observations about component synchronization.

Firstly, the first hand configuration reaches its starting value before the wrist reaches the starting location of the sign (A). On average, on those signs, the location is reached

12. Here, "location" designates the specific position of the wrist that can be numerically defined. It corresponds to decelerations or stops in the velocity of the wrist. It differs from the "hand placement" which is a zone in the signing space.

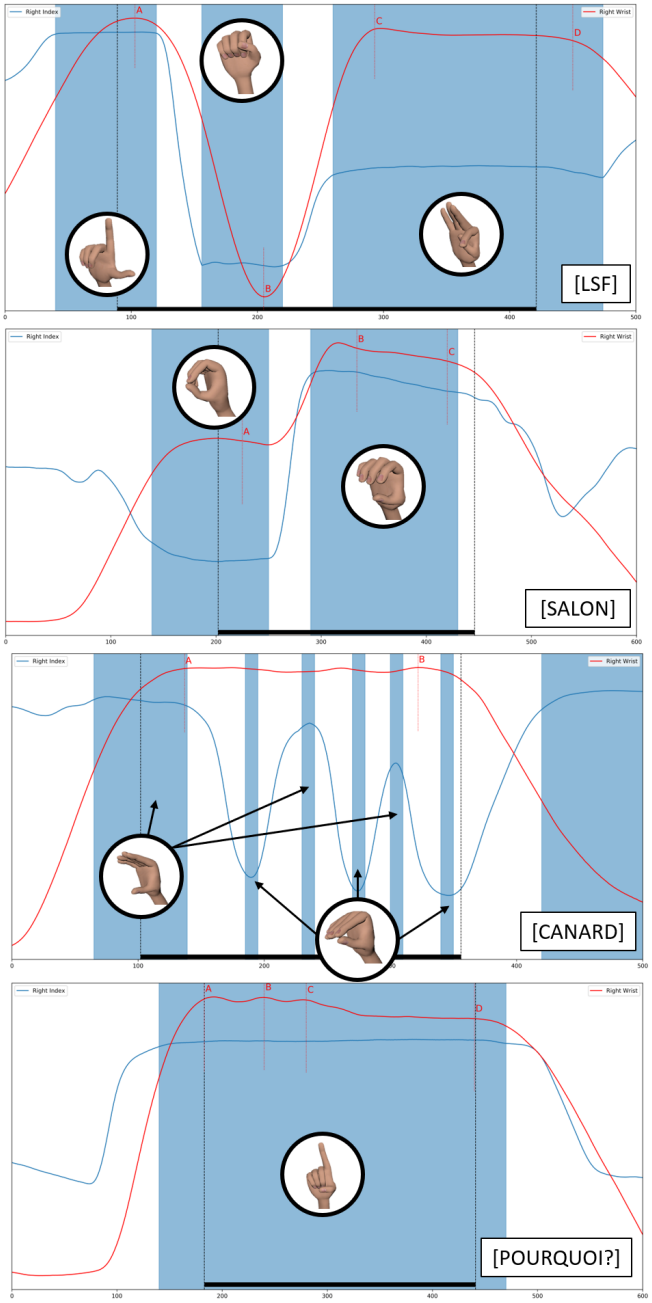

Figure 8.30 – The synchronization features of the signs *[LSF]*, *[SALON]* (living-room), *[CANARD]* (duck) and *[POURQUOI?]* (why?). The norm of the position of the wrist with respect to the hips corresponds to the red curve while the norm of the position of the tip of the index (or ring finger) with respect to the position of the wrist is in blue.

at 76% of the hand configuration interval corroborating the results of Duarte that who found that the hand configuration was always reached before the hand placement on the

45 signs that he studied [57].

Secondly, when there are changes both in the hand configuration and in the hand location (i.e. for the signs *[LSF]*, *[WEEK-END]*, *[CHAT]* and *[SALON]* which have a hand movement), the second (or third) hand configuration is reached **before** the second (or third) hand location.

Thirdly, at the end of the signs, the hand configuration changes **after** the beginning of the transition motion towards the next sign. In general, the hand configuration is reached before the end of the hand movement and remains stable slightly after the beginning of the next hand movement. The result of those three first observations is visible on Figure 8.31.

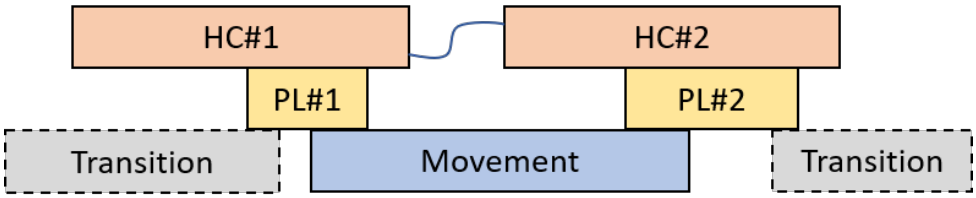

Figure 8.31 – Scheduling of the different hand component for signs with two hand configurations following the observations made on the captured data.

Lastly, the beginning of the sign is identified as such in the annotation (black segment) when the manual configuration is reached **and** the wrist is close to its starting location. On average on the 6 signs, the sign begins 0.24s (std of 0.07s) after the beginning of the hand configuration [13].

This last observation is less important in the context of sign synthesis as the synthesized signs do not have to be annotated but could be of use for determining new annotation methods or to enrich the database with new annotated signs. To take the three other observations into consideration, we constrained the general ordering of the components. Practically, we forced the hand configuration to be reached before the end of a movement and after the beginning of the subsequent movement.

We perceptually found that slight variations in timing are hardly perceptible when the signs are played by the avatar as long as the general scheduling of events is respected (e.g., hand configuration reached before placement). However, in order to gain precision in a context of movement analysis in cognitive sciences or linguistics, it would be beneficial to

13. Those numbers should be considered carefully as data were annotated by only one human annotator which naturally biased the results. Ideally, the results of several annotators should be compared.

make these same calculations on a larger number of signs in order to have the possibility to use statistical tools.

8.3 Utterance Synthesis

The previous section detailed the synthesis of signs thanks to the modification of the values taken by each of the phonological components. The sign synthesis techniques presented make it possible to generate signs in their form of citation or inflected signs. Utterances are composed of both those types of signs. They are sets of signs, performed simultaneously or sequentially, that are close to the concept of "sentences" of oral languages.

The synthesis of utterances involves different mechanisms than the synthesis of signs. First of all, utterances have a sequential aspect even if they do not always correspond to a succession of non-overlapping signs. It is therefore necessary, after having synthesized signs or inflection mechanisms from real data, or retrieved them from the database, to be able to link them together with motions without any meaning of their own. We call these motions *transitions*[14]. The first part of this section (8.3.1) will be devoted to the study of these transitions and to the implementation of simple techniques to improve their realism.

However, utterances are not an exclusively sequential phenomenon. In many situations, the signer is required to reproduce signs previously made simultaneously with other signs to give contextual information. In a second part (8.3.2), we will see how our proposal for phonological synthesis allows us to take simultaneity into account.

8.3.1 Utterance as a Sequence: Generation of Believable Transitions

This section is based on a paper published in the proceedings of the International Conference on Human-Computer Interaction 2017 [52].

In this part, we will consider an utterance as a sequence of signs. We will revisit this assumption in Section 8.3.2.

To create transitions, the most common technique consists in interpolating the last posture of the first sign with the first posture of the next sign. We implemented this method

14. Those transitions can include coarticulation phenomena.

with different interpolation profiles (slerp, cosine, sigmoid) on the joints orientations and called it the *motion interpolation* method (see Figure 8.32, left). Interpolated transitions are simple to compute but do not take into account the transition data recorded in the database. They are purely artificial and, therefore, do not innately possess the features of realistic motions. As a result, it gives visually poor results when their duration exceeds a certain threshold.

We therefore propose to use the data from the database to define a new way of creating transitions called *motion blending*. In this case, transitions are computed as a linear blending of two motions: the movement following the first sign (*retraction* of S_1) and the movement preceding the second sign (*preparation* of S_2) (see Figure 8.32, right). This method gives better results in terms of realism (partial conservation of the context) and robustness with respect to a longer duration than the interpolation method. However, the quality of the resulting movement greatly depends on the content of the database, of the nature of the required signs and of the annotation. For instance, if there is no captured motion before the second sign or after the first sign in the database, the transition will be less realistic (use of an idle skeleton with a default posture instead). This occurs at the beginning or end of recorded motion sequences or for synthesized signs. In those cases, motion interpolation can provide more realistic results.

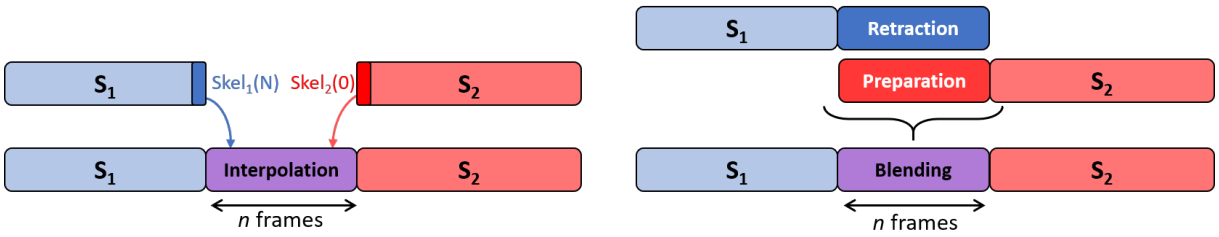

Figure 8.32 – Transition synthesis methods: motion interpolation (left) and motion blending (right).

An important parameter of transition synthesis is the length of the transition. For the interpolation method, it corresponds to the number of frames n being added between the signs. A too long transition duration reveals the imperfections of the method in terms of realism. For the blending technique, it is the same number of frames n extracted from the retraction phase of S_1 and the preparation phase of S_2 to be blended together. If the computed duration is too long, part of another sign could be extracted in addition to the *preparation* or *retraction* part. For example, in our database, a transition lasts on average about $0.30s$ (i.e. 60 frames at $200Hz$) and never exceeds 100 frames. It is

straightforward that, if the computed transition time exceeds 100 frames, the transition will be computed using part of the signs present **before** S_2 and **after** S_1 when it should only use the preparation phase of S_2 and the retraction phase of S_1. This results in a quite unrealistic motion that will appear as a hesitation from the avatar in the final synthesis [15]. This illustrates the need for a good computation of the transition duration.

8.3.1.1 Analysis of the Transition Duration

The length of the transitions impacts the quality of the synthesized animations of LSF utterances. A too short or too long transition will be perceived as strange and will often have repercussions on the general comprehension of the sentence [237]. The computation of a correct duration for transitions is therefore necessary.

In order to define some empirical laws and invariants for the transition duration, we considered 89 transitions extracted from two sequences of sign language *MoCap* data from the *Sign3D* database [69]. The 89 transitions are the motions between two consecutive signs. They are extracted from the sequences using the automatically refined annotations presented in Chapter 7. The duration of each transition was obtained by computing the difference between the timestamp of the beginning of S_2 and the timestamp of the end of S_1.

a) Duration with Respect to the Type of the Surrounding Signs To determine if the length of the transitions is related to the nature of the surrounding signs, two different features of signs in LSF were examined in order to quantify their impact on the transition duration:

1. The number of hands used in the execution of the sign:
   - $1H$ One hand ([*SALON*] (living-room), for example, see Figure 8.33).
   - $2H$ Two hands as in the [*CHAT*] (cat) sign (see Figure 8.33).
   - $2H_{ctxt}$ One hand is doing a one hand sign and the other is preserving the context: for example a pointing gesture (one hand sign) toward the other hand showing the remnant of the previous [*MAISON*] (house) sign (context). This is a case of contextualized signs.

15. Incidentally, if the **annotation of gloss** is shifted by $0.10s$ or more from the actual point in time from where it should be, it will impact the final synthesis by creating the same "hesitations" or truncated motions. For both methods, the quality of the original annotation that will identify the signs is of prime importance. The segmentation refinement presented in Chapter 7 makes it possible to obtain more accurate sign segments than the segments provided by the manual annotators.

2. The symmetry of the sign (only in the case of a two-hands, non contextualized sign). A sign is considered as symmetric if the two hands perform a symmetric motion regardless of the axis of symmetry:

   $2H_{noSym}$ The movement is not symmetric (e.g., [*ESCARGOT*] (snail) sign, see Figure 8.33).

   $2H_{sym}$ The movement is symmetric as in the [*CHAT*] sign.

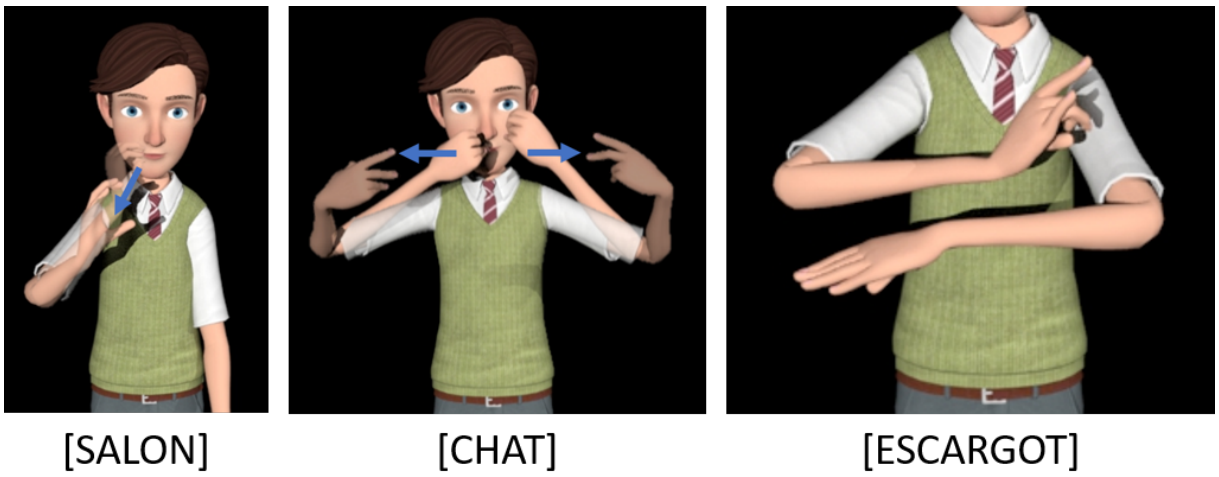

Figure 8.33 – The one-hand sign [*SALON*], the two-hands symmetrical sign [*CHAT*] and the two-hands asymmetrical sign [*ESCARGOT*].

A transition was defined by the nature of the previous and following signs. Table 8.4 lists all of the possible types of transitions, their distribution in our data set and the mean duration and standard deviation according to the type of transition and the manual segmentation.

The shortest transitions are obtained for the transitions from two-hand symmetric signs to two-hand asymmetric signs and for the passage from one-hand signs to other one-hand signs. Apart from the transitions from two-hand asymmetric signs to one hand signs whose result is impacted by the outlier (average without outlier: $0.315s$), the longest transitions are between two-hand asymmetric signs. Figure 8.34 shows the mean duration of the transitions only depending on the number of hands. The transition between two one-hand signs (1H $\rightarrow$ 1H) and two two-hand signs (2H $\rightarrow$ 2H) is shorter (and might be interpreted as easier) than adding or removing a hand between signs (1H $\rightarrow$ 2H and 2H $\rightarrow$ 1H). However, as the standard deviation is quite high compared to the mean values, the conclusion that can be made is that the number of hands and the symmetry of the

S_1	S_2	Number	Mean duration (std) in seconds
$1H$	$1H$	6	**0.268** (0.095)
	$2H_{noSym}$	7	0.310 (0.090)
	$2H_{sym}$	5	0.320 (0.069)
	$2H_{ctxt}$	0	/
$2H_{noSym}$	$1H$	9	0.368 (0.162)*
	$2H_{noSym}$	9	**0.334** (0.098)
	$2H_{sym}$	11	0.292 (0.054)
	$2H_{ctxt}$	0	/
$2H_{sym}$	$1H$	4	0.305 (0.077)
	$2H_{noSym}$	16	**0.258** (0.081)
	$2H_{sym}$	6	0.317 (0.145)
	$2H_{ctxt}$	6	0.295(0.089)
$2H_{ctxt}$	$1H$	0	/
	$2H_{noSym}$	1	0.300 (0)
	$2H_{sym}$	5	0.304 (0.107)
	$2H_{ctxt}$	4	0.303 (0.078)
Total		89	0.303 (0.098)

* Without outlier: 0.315 (0.041)

Table 8.4 – List of all the transition types, mean duration and standard deviation.

signs surrounding a transition do not significantly impact the duration of the transition. An analysis of a higher number of transitions could lead to more conclusive results.

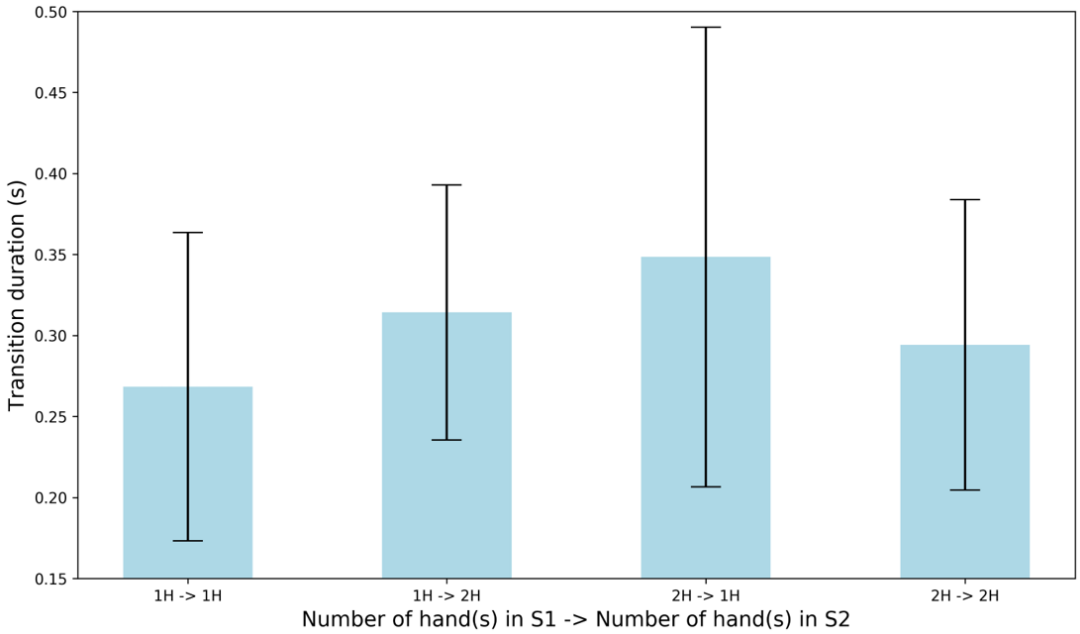

Figure 8.34 – Mean duration considering the number of hands involved in the first and second signs.

b) Duration with Respect to the Distances Between Postures The average distances between the last posture of the previous sign and the first posture of the next sign were computed. Two types of distances were used: the **Geodesic Distance** between the joint orientations and the **Euclidean Distance** between the joint positions of the two skeletons. Each distance was averaged on the number of joints.

Considering two skeletons $Skel_1$ and $Skel_2$ composed of oriented joints, the Geodesic distance between the orientations (quaternions) of $Skel_1$ and $Skel_2$ is defined as the mean of the Geodesic distances between the orientations of each joint of $Skel_1$ and the corresponding joint of $Skel_2$ with:

$$\text{GeodesicDistance}(orient\ a, orient\ b) = ||log(a^{-1} * b)|| \tag{8.10}$$

The Euclidean distance between $Skel_1$ and $Skel_2$ is defined as the mean of the Euclidean distances between the positions of each joint of $Skel_1$ and the corresponding joint of $Skel_2$:

$$\text{EuclideanDistance}(pos\ a, pos\ b) = ||a - b|| \tag{8.11}$$

To equally take into account the two types of distances, we normalized them (by subtracting the mean value and dividing by the range of values) and computed the average distance.

Figure 8.35 shows the duration of the transitions with respect to the distances between the postures at the beginning and end of the transitions.

Considering our examples, we can note that:

1. The general tendency of the duration is to increase with the distance.
2. Apart from a single outlier, the duration never exceeds $0.5s$.
3. The duration never goes under $0.1s$.

8.3.1.2 Computation of Transition Duration

By using the results of our analysis, we aim to find a transition duration that best emulates the behavior of a real LSF signer in order to synthesize more natural and intelligible utterances. As the analysis concluded that the number of hands and symmetry of the surrounding signs did not have a great impact on the transition duration, these two parameters were not taken into account in the duration computation. We developed four different computations of the transition duration to take into account the distances

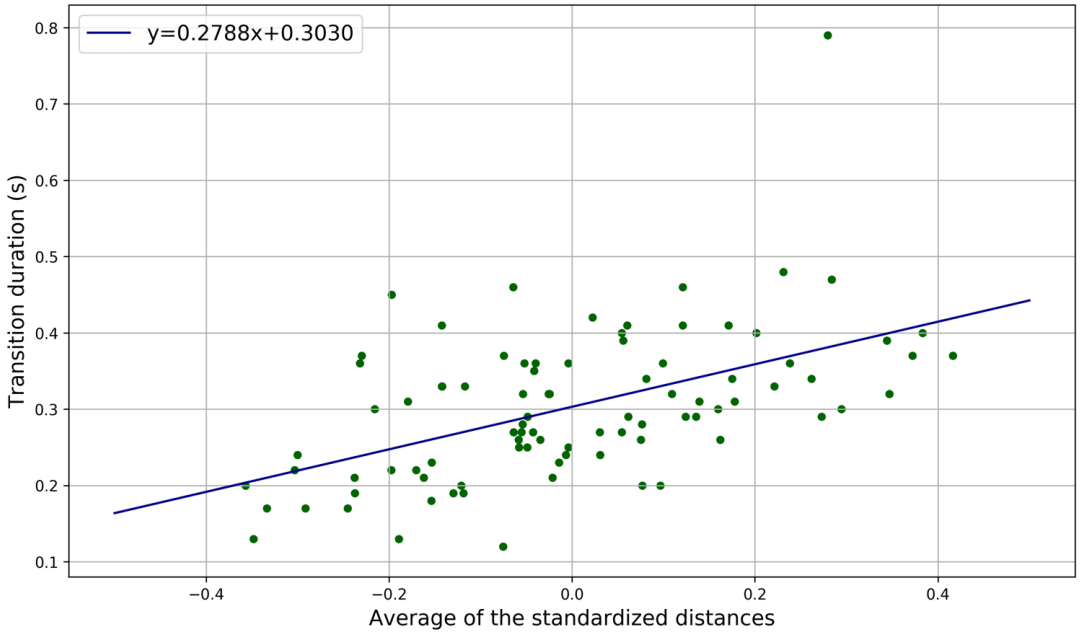

Figure 8.35 – Duration of the transitions in function of the average of the two normalized distances.

between the postures. They are detailed hereafter.

a) Velocity-Based Duration First, we implemented a simple duration computation based on the distance between the extreme positions $Skel_1$ (end of the first movement S_1) and $Skel_2$ (beginning of the second movement S_2) and the mean velocity of the two surrounding signs without taking into account the results of the analysis:

$$velDuration = (\alpha * \frac{2 * \text{EuclideanDistance}(Skel_1, Skel_2)}{MeanVel_{S_1} + MeanVel_{S_2}} + (1-\alpha) * \frac{2 * \text{GeodesicDistance}(Skel_1, Skel_2)}{MeanAngVel_{S_1} + MeanAngVel_{S_2}} \quad (8.12)$$

This method produces visually acceptable results for small distances and high velocities (short duration) but the computed transition duration is often longer than the ground truth equivalent. It sometimes reaches unacceptable values (sometimes as high as $1.5s$) that give unrealistic results with a slow down or a "hesitation" depending on the type of synthesized transition (*interpolation* or *blending* respectively).

Directly using the results of our analysis we propose three other computation methods for the transition duration.

b) Bounded Duration A first, simple measure was to put an empirical lower limit at $0.1s$ and a higher limit at $0.5s$ on the *velocity-based duration* using the items #2 and #3 of the observations on the data. The transitions with inconsistent duration are thus automatically changed to more correct values. We visually noted an improvement in the rendering of the animation for the transitions involved.

c) Linear Duration A second attempt consisted in exploiting the data of Figure 8.35 to compute the coefficients of a trend line (the coefficients are visible in the legend of Figure 8.35) and use those coefficients to predict the value of a new transition with respect to the mean of the two normalized distances. Here again, we noticed an improvement with respect to the *velocity-based duration*.

d) Surface Duration Finally, we computed the equation of a surface using normal equations to minimize the linear least square distance between the surface and the data (Euclidean and Geodesic distances of the 89 transitions). In this case, we do not average the two distances: they form two distinct axes of a three-dimensional space with the transition duration. With this method, we find the optimal coefficients and use those to predict the duration of a new transition. Figure 8.36 shows the surface computed thanks to normal equations and the data that has been used to do the computation.

$$\theta = (X^T * X)^{-1} * X^T * y \tag{8.13}$$

with

- θ: the optimal parameters of the surface,
- X: a $89 * 3$ matrix containing the inputs (the first column contains only the value 1, the other two columns contain the Geodesic and Euclidean distances of each transition respectively), and
- y: a vector containing the durations of the transitions.

Using the equation (8.13) and our data, we found:

$$\theta = \begin{pmatrix} 0.148777 \\ 0.663958 \\ 0.275229 \end{pmatrix}$$

The mean error between the real and computed duration for our 89 transitions can be calculated as:

$$MeanError = \frac{1}{89}\sum_{i=1}^{89}\sqrt{(y_i - (X_i * \theta))^2} = 0.06075s \tag{8.14}$$

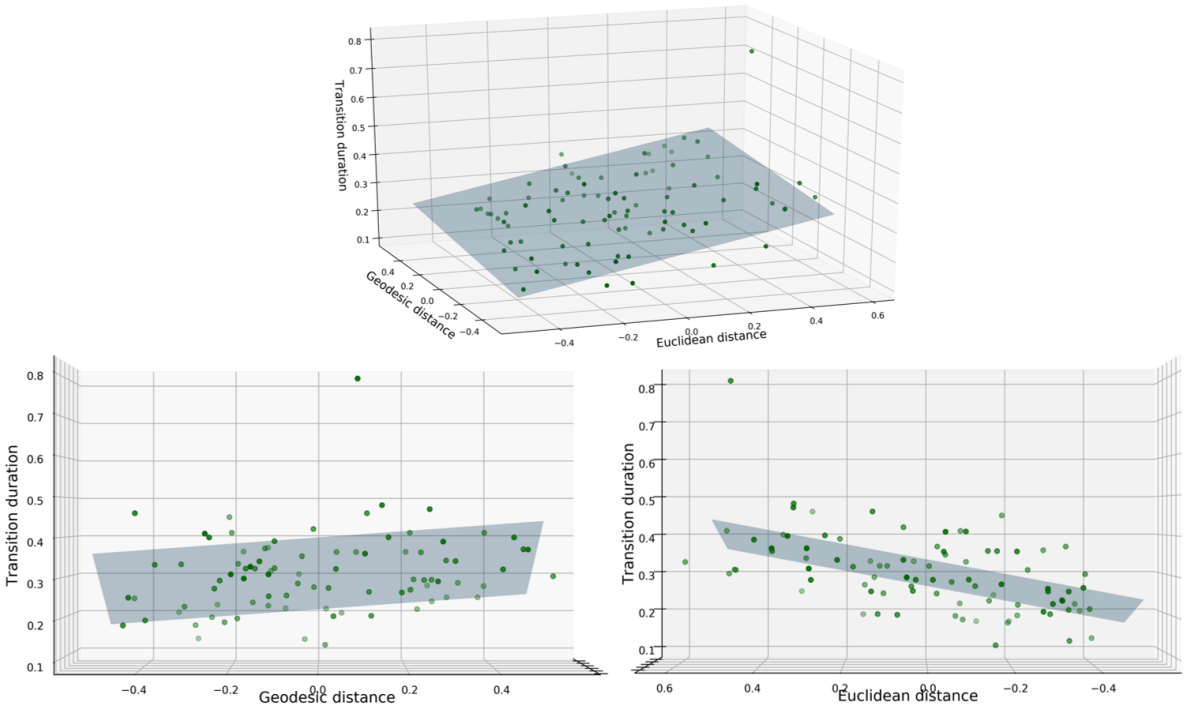

Figure 8.36 – Surface computed using normal equations (8.13) with the annotation data.

8.3.1.3 Results

In this section, we compare and visualize the results of the various computations methods of the transition duration.

We first compare **synthesized transitions with their ground truth equivalent** by picking 22 transitions from our *MoCap* database that were not part of the 89 transitions used for the analysis and by retrieving their actual duration (according to the

automatically refined manual annotation). Then, we computed the duration according to our four different computations and measured the difference between the ground truth duration and the computed durations. The results are in Table 8.5 and the details are shown in Appendix C. We can see that the *surface duration* gives the results the closest, on average, with the ground truth. Moreover, the *bounded*, *linear* and *surface durations* are the computation with the least "absurd" values (i.e. values that differ from the ground truth of more than $0,25$s). The *velocity-based duration* that does not take advantage of our analysis results gives, on average, the results the farthest from the ground truth.

	Velocity-based	**Bounded**	**Linear**	**Surface**
Average (s)	0.16456	0.11301	0.11281	**0.10153**
Std (s)	0.12151	0.08597	0.09163	0.08136
Values more than 0.25s away from the GT	18.2%	**9.1%**	**9.1%**	**9.1%**
Values less than 0.1s away from the GT	36.4%	50.0%	50.0%	**63.6%**

Table 8.5 – Average and standard deviation of the difference with the ground truth (GT).

In a second experimentation, we chose to pair any, not necessarily consecutive, signs in the database. It is thus impossible to compare the performances of the generated transition with the ground truth which does not exist but, instead, we can compare the **synthesized transitions with one another**.

On Figure 8.37, we have chosen to analyze the transition between the sign [*PAYER*] (to pay) and the sign [*MUSÉE*] (museum). The duration of the corresponding transition has been computed for each of the methods. We can see that the *velocity-based duration* method gives an abnormally high value of $1.47s$. Indeed, the Euclidean distance between the two extreme skeletons of the transition is equal to 1.08172 and the Geodesic distance is 0.103753. The high value of the Euclidean distance can be explained by the fact that [*PAYER*] is a two-hand sign and [*MUSÉE*] is a one-hand sign (see Figure 8.37).

The resulting transition is not convincing using either the *Interpolation* (slowdown) or *Blending* (artifacts due to unwanted sign chunks added to the animation) generation techniques.

The *surface duration* gives a much more acceptable result with a transition of $0.51539s$:

$$\begin{aligned} duration &= \theta_0 + \theta_1 * GeoDist + \theta_2 * EucDist \\ &= 0.148777 + 0.663958 * 0.103753 + 0.275229 * 1.08172 \\ &= 0.51539s \end{aligned}$$

The other methods do not allow the duration to exceed the $0.5s$ boundary and their results are also more convincing than the *velocity-based duration* method.

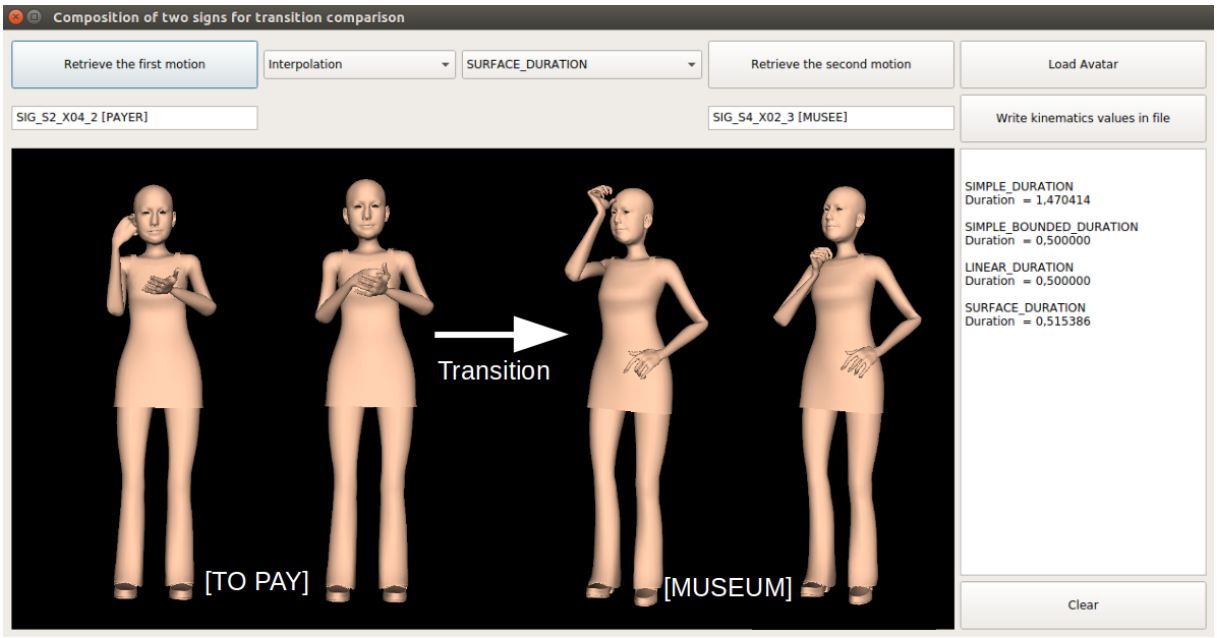

Figure 8.37 – Composition of two signs for transition comparison. The last posture of the first movement is quite far from the first posture of the second movement.

8.3.1.4 Consequence: Improvement of Motion Retrieval

The analysis of the transitions shows that the distance between two consecutive motions has an impact on the transition duration. We also know that a long duration will often be less realistic than a shorter duration. We thus implemented a new motion retrieval method based on those observations. This method retrieves a motion from a set of possible motions in order to minimize the average distance of this motion with the previous and/or following motion(s). By minimizing the distance, the duration of the transition is naturally shorter than if we had taken a random motion among all of the possible motions.

8.3.2 Adding Simultaneity: Back to Phonology

In sign languages, unlike in oral languages, signers have the possibility of using their body, in particular, their two hands, to make several signs at the same time. This is mainly used to keep the context in a description in order to place an entity B with respect to another entity A defined beforehand. In this case, A is designated by its sign, a part of its sign or a proform and placed in the signing space, then B, also designated by its sign, part of it or a proform is added to the scene and placed in relation to A.

A hand, often the non-dominant one, thus sometimes serves as a reference point that can be maintained beyond a single sign. This hand will remain fixed to represent a fixed element of the scene in relation to which the other elements will be located and interact. One can imagine the description of a busy street. The fixed hand would represent a building while the other hand could sign the different entities (pedestrians, cars, cyclists, motorbikes) that pass by this building.

In the same way, a head movement, signifying negation for example, could be set at a higher level than the sign level. The utterance "I don't dance" presented in Section 5.3.1 is an example of an overlay of the head movement channel that is not synchronized to a particular sign.

The ability to make several signs simultaneously constitutes a richness of sign languages that is underexploited when one tries to recreate it using an avatar. The sequential vision of sign languages is a simplification often adopted when generating sentences for avatars. As we have seen in Section 5.3.2.1, this vision results in the synthesis of utterances by the concatenation of isolated signs with the addition of acceptable transitions between each sign.

It is then possible to make simple correct utterances but it becomes limited when it comes to making complex sentences, descriptions, or to tell stories... Section 8.3.1, on the generation of credible transitions, is part of a work based on concatenative synthesis and a sequential vision of sign language. It remains interesting for the construction of basic utterances but, if left as it is, does not allow to take into account the mechanisms of simultaneity of sign languages.

In the section 8.2, we presented a method of constructing new signs that we called *phonological recombination* that focuses on the phonological level of French Sign Language to construct new signs or inflection mechanisms from existing data and existing techniques parameterized in order to preserve the realism of the original motion. We have, for the

moment, presented the use of this technique at the sign level. However, there is nothing to prevent its use at the utterance level. Here we present two consequences of the phonological processing which, if exploited, can increase the realism at an utterance level as well as the ability of the synthesis system to create novel utterances.

First, we saw in Section 8.2.4 that the desired values of the phonological components in a sign were not reached at the same time for all phonological components (the hand configuration was reached before the arrival at the starting location and before the beginning of the articulatory movement) but that the beginning of a sign was indicated by human annotators when all phonological elements had reached their value, or a value close to it. Thus, if one merely synthesizes an isolated sign, it is possible to start it with all the values of its phonological components reached. However, if one wishes to be rigorous and improve the realism at the utterance level, it may be interesting to take into account this slight discrepancy in the phonological components. The synthesized transition should thus not be between two complete skeletons $Skel_1$ and $Skel_2$ corresponding to the posture at the end of sign $S1$ and at the beginning of sign $S2$ with a duration fixed in advance but rather between each phonological element with different transition durations but dependent on the global duration as calculated in the previous section (see Figure 8.38).

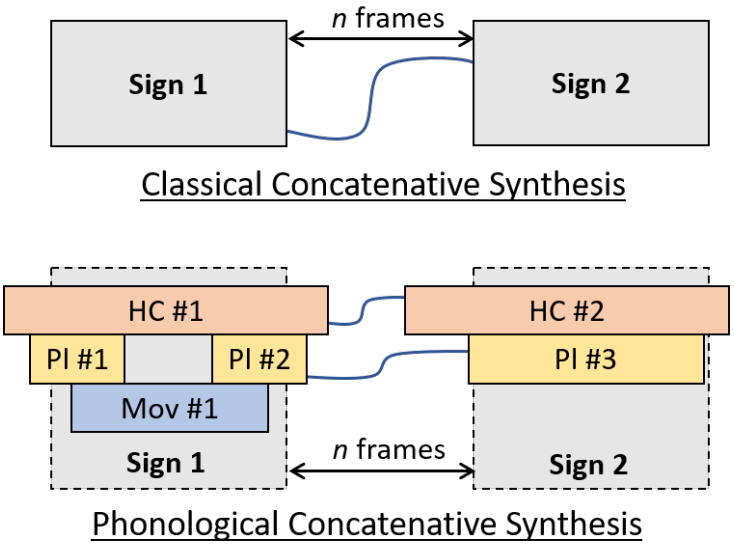

Figure 8.38 – The classical concatenative synthesis (top) and our proposition of an improved concatenative synthesis based on a phonological division (bottom).

Second, this processing by phonological components also allows us, in theory, to fix the value of one (or more) component(s) during the realization of more than one sign. For

example, a negative head movement can be assigned to the head channel for the duration of two signs, allowing us to generate phrases such as the aforementioned "I do not dance". This also allows to treat both hands independently and to add simultaneity by playing a sign on one hand while one or more signs are played on the other (see Figure 8.39). This phonological element processing, which translates concretely into the independent modification of the physical channels (neck, hand, fingers) of the skeleton controlling the movement, allows to add simultaneity in the generated utterances in order to create richer and more accurate sentences. Figure 8.40 shows the avatar positioning, by pointing with the right hand, an object in relation to a table represented by the left hand. This synthesis result is not a whole sign, but rather a composition of phonological elements to obtain this result.

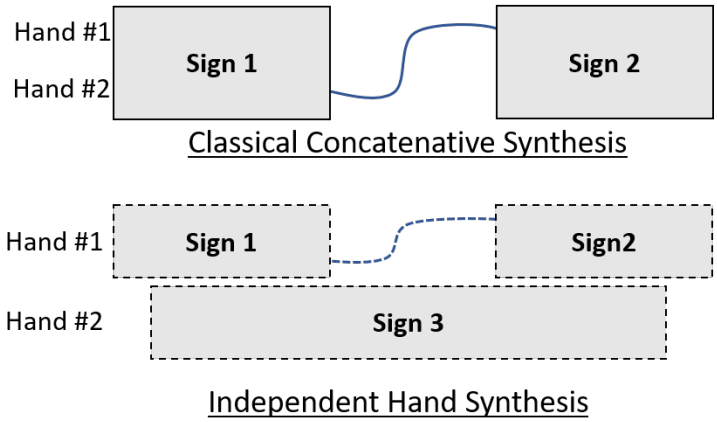

Figure 8.39 – Second improvement of the utterance synthesis by treating the hands independently.

8.4 Summary and Discussions

A large part of the existing work in the field of signing avatar animation uses exclusively, for sign synthesis, either replay of real motion data or pure procedural synthesis often based on phonological specification systems such as HamNoSys. In the first case, the movements of the avatar keep the realism of the real movements but the number of signs that can be generated is limited by the initial database. In the second case, the variety of signs that can be synthesized is far less restricted – the expressiveness of the specification system being almost the sole limitation – but the resulting signs are robotic and unrealistic.

Figure 8.40 – Example of the synthesis of a simultaneous phenomenon: the avatar is positioning, by pointing with the right hand, an object in relation to a table represented by the left hand.

We propose a hybrid system that takes advantage of the two philosophies in order to create new signs, absent from an annotated database of initially captured movements, but retaining the realism properties of real movements. For this, we rely on a vision centered around phonological components as defended by linguistic work and at the base of many sign representation systems. We thus propose to constitute new signs in their citation form as well as inflection phenomena thanks to techniques of (i) motion retrieval per phonological component and (ii) recombination of these elements. Based on Millet's classification of iconicity dynamics, which allows us to sort the LSF mechanisms according to the main phonological elements involved, we describe different techniques that allow us to multiply the possibilities of a database by modifying the values taken by the individual phonological elements. In addition to simple recombination, we have implemented pure synthesis techniques, namely inverse kinematics and interpolation techniques, that we have adapted to be as close as possible to the ground truth resulting, for example, in the definition of a parametrized sigmoid interpolation.

Indeed, in addition to using the data as the basic unit of our synthesis, we use the knowledge extracted from the analysis of these data to improve the final result. The data thus constitutes a building material and a knowledge base on motion, thanks to the kinematic profiles of the different trajectories of the joints.

As for utterance synthesis, we first worked on transition generation for a concatenative synthesis engine. In this case, an utterance is seen as a sequence of non-overlapping signs. However, this work, combined with our previous analyses of sign synchronization, can be reused to create realistic transitions in the case of signs performed simultaneously (e.g., in a description situation). Moreover, since we work on both signs in their citation form and on inflection mechanisms, we have the possibility to build rich utterances that are not made up exclusively of uninflected signs.

The team's previous work relied on an animation system developed to control an avatar from captured data [64]. This work was based on a multi-channel control approach. These channels were associated with specific controllers for animation, each controller for body parts being associated with motion retrieval in the dual database. Although some body parts were associated with annotated phonological tracks (left and right hand movements and hand configurations), the previous system was more dedicated to animation modeling than to linguistic modeling. It was thus possible to create, with such a platform, some examples of linguistic mechanisms as the ones described in [238], but the aim was above all to assess the animation capabilities of the system.

In this thesis, the approach is reversed. Based on a linguistic theory:

— we have covered a large number of linguistic mechanisms at different levels (phonological, lexical, utterances) and constructed synthesis models that reproduce these mechanisms; thus we thus propose a linguistically oriented generalization of the previously proposed examples;

— we have clearly distinguished the synthesis process at the sign level and at the utterance level;

— we created inflected signs (e.g., specifiers, spatialization), whereas the examples of elements synthesized previously corresponded to signs in their citation form;

— we have extended the synthesis system, by proposing a synthesis solution when motion playback is not available (e.g., for spatialization or pointings). We thus use either motion retrieval or motion synthesis;

— we use *MoCap* data as analysis material and transfer the results of the analysis for the synthesis. This intertwined approach of analysis and synthesis is innovative, as most of the previous work proposed studies of analysis or synthesis, but not for both at the same time.

The following paragraphs are discussions and perspectives for future synthesis work.

In our synthesis of pointing motions, we have broken down the mechanism into three phases: the reaching and retracting motions computed by interpolation and the pointing pose synthesized by IK. It would be more efficient to propose a single temporal synthesis model that preserves the natural characteristics of real movements. This approach has already been developed using a sensorimotor model for LSF movements [239], and improved by techniques of reinforcement learning for pointing gestures to respect upper-arm synergies [240]. But the use of such approaches in our phonological system would require the integration of a time constraint so that the synchronization constraints between the phonological components are respected. Alternatively, following the approach used in this thesis, it would be interesting to propose different "template" models so that we incorporate the various velocity profiles observed in intra-sign movements or inter-sign transitions [57]. Finally, some temporal mechanisms have been observed in repetitive movements, with a trajectory profile that repeats itself with a progressive decrease in amplitude [57], [241]. We aim at integrating all these mechanisms in future work.

In the context of this thesis, we focus on three phonological components: hand configuration, hand placement and hand movement. However, the non-manual components are of crucial importance for the proper understanding of a signed utterance. It could be interesting to include them to the work done in this thesis. Their inclusion can constitute an overlay of the existing system and will not require its modification. For example, the addition of synthesized facial expressions corresponding to emotions [210] or precise brow motions [14], can increase the expressiveness of our system. Hand orientation is another phonological component of sign language that has not been discussed here. We have considered that hand orientation is strongly dependent on the arm and hand configurations and have not done further studies on the subject. The analysis of the role of hand orientation and its synchronization with arm movements is a promising perspective for the continuation of this work.

Chapter 9

Conclusion

- On est pas là pour "philosopher", Carpentier.
- Ben, quand même ...

P'tit Quinquin
Bruno Dumond, 2014

9.1 Contributions

This thesis deals with the capture, annotation, synthesis and evaluation of the arm and hand motions for the animation of avatars communicating in sign languages. We aimed to create new linguistically relevant French Sign Language (LSF) content that possesses the realism of human motions. To this end, we have proposed to use a database of captured movements as: (i) raw material for our synthesis system and, (ii) analysis material to identify and recreate the features that make signing motion human-like and semantically meaningful.

For this purpose, we have set up a complete processing chain ranging from the creation of an LSF corpus and its automatic annotation to the synthesis of new sign language content. In each contribution, we have placed the manual phonological components that are hand movement, hand placement and hand configuration at the heart of our work, following existing linguistic work.

Design, Recording and Perceptual Evaluation of a Dedicated LSF Corpus

Data constitute the material from which the information used to analyze and synthesize sign language movements is extracted: they are therefore the basis of our thesis

work. *MoCap* techniques capture the motions of the entire human body with great spatial and temporal precision and, in general, an avatar animated with captured data will have more realistic and natural movements than an avatar animated in a procedural way, without pre-captured data. We therefore reviewed the few publicly available LSF *MoCap* databases and concluded that none of them were ideally suited to our work.

Our first contribution is the design and recording of a new French Sign Language *MoCap* corpus. This corpus, called *LSF-ANIMAL*, includes different LSF mechanisms needed for our synthesis work: isolated hand configurations, isolated signs, contextualized signs, iconic mechanisms such as shape and size specifiers, proforms or pointing gestures and full utterances containing all of the former items. A complete marker set adapted to the variability and subtlety of sign language movements was designed and tested before applying it on two deaf LSF instructors for the recording of the *LSF-ANIMAL* corpus.

The data were processed and evaluated through one perceptual study involving 50 participants. We found that the signs were recognized far above chance level and that the participants, apart from a generally deplored lack of facial expressions, mainly found the motions of the avatar to be precise, natural and believable when performing signs and utterances [208].

Development of Automatic Segmentation and Labeling Techniques for Different Annotation Tracks

Annotation consists in the segmentation and labeling of the timed data contained in a database. An accurate annotation of data is paramount, whether for data analysis or data-driven synthesis. Indeed, it adds a semantic layer to raw motions that can then be queried according to linguistic constraints. However, if done manually, it is a tedious and error-prone task.

Our second contribution is the definition of an annotation scheme for our *LSF-ANIMAL* corpus and the development of automatic and semi-automatic methods for annotating the data on different tracks. Following a parametrical approach of SL, we first defined a multi-track annotation scheme with a dedicated track for each of the hand components (configuration, placement, movement and orientation). We then proposed an automatic refinement of the manual segmentation of the gloss tracks based on the kinematic properties of the two wrists to obtain a less biased, more accurate, and more homogeneous segmentation.

We also compared the combined use of various feature sets and machine learning algorithms for the automatic segmentation and labeling of the hand configuration tracks. This method was trained and tested on one corpus, namely the *Sign3D* corpus, and was then applied to the *LSF-ANIMAL* data set to reduce its annotation time. Finally, we presented a computation of the hand placement with respect to the height, distance and radial orientation of the wrists and a discretization of the signing space.

Implementation of Data-Driven Motion Synthesis Techniques to Generate New Sign Language Content

Existing work on signing avatars animation often rely on pure procedural techniques or on the replay of *MoCap* data. While the first solution results in robotic, unnatural motions, the second one is very limited in the number of signs that it can propose.

Our third contribution is the implementation of data-driven motion synthesis techniques to increase the variety of SL motions that can be made from a limited database. In order to generate new signs, inflection mechanisms and utterances based on our annotated LSF *MoCap* corpus, we relied on *phonological recombination*, i.e. on the motion retrieval and modular reconstruction of SL content at a phonological level.

We focused on three phonological components of SL: hand placement, hand movement and hand configuration. We proposed to modify the values taken by those components in different signs to create their inflected version or completely new signs by (i) applying motion retrieval at a phonological level to exchange the value of one component without any modification, (ii) editing the retrieved data with different operators (temporal inversion, repetition, mirroring, etc.), or, (iii) using conventional motion generation techniques such as interpolation or inverse kinematics, parameterized to comply to the kinematic properties of real motion observed in the data set. For example, we proposed to use the sigmoid interpolation with precise parameters to create natural transition motions that are close to the kinematic features of real data.

9.2 Future Work

Finally, we present here various avenues of research for the continuation of this work.

Improving the *MoCap* Processing Chain for Finger Motions

During this thesis, a lot of time has been dedicated to the technical considerations concerning passive *MoCap* and avatar animation for the particular application of sign language. We discovered that little work had been done in the field of *MoCap* for the precise capture and reconstruction of the finger motions. Hand and finger configurations are often reconstructed with inverse kinematics techniques by only capturing the position of the fingertips and of the back of the hand. The main shortcoming of this technique is that the reconstruction of the hand posture is based on several assumptions on the limits of the joints and on the dependencies of the fingers. In this thesis, we chose to put aside these assumptions to capture the hand and finger motions as truthfully as possible. It resulted in the definition of a new marker set and in extensive post-processing work to correctly identify the markers and reconstruct the finger motions from their positions. Further work dedicated to the definition of a *MoCap* processing chain that takes into account not only the general body motions but also the subtle finger motions would be a useful improvement for all data-driven avatars.

Automating the Annotation of Each Track

The manual annotation of sign language data is a time-consuming process that we partially automatize in this thesis. We have completely annotated (i.e. automated segmentation as well as labeling) 8 of the 18 tracks we defined (the 2 hand configuration tracks and the 6 hand placement tracks). It would be extremely useful to propose a complete *MoCap* data annotation process that would allow to annotate (i) the channels of hand movement using, for example, the techniques developed in handwriting recognition [242], [243] to qualify the trajectories realized by the wrist, (ii) the orientation of the hand by calculating the direction of the normal to the palm and the axis of the hand, and (iii) non-manual components. And, then, the values annotated on each component track can help the recognition of the performed signs in order to obtain an annotation at a gloss level. The work of this thesis allows us to evaluate each element of the process, from capture to synthesis. In order to obtain generalized results on other data sets and to increase the generation capacity of our system, it will be necessary to automate the annotation. For this purpose, more data will have to be captured with a higher number of LSF native signers, thus promoting the use of deep learning techniques.

Performing Further Perceptual Evaluations

Signing avatars are intended for a specific population, that is people communicating in a sign language. It is thus mandatory that this population validates the results of our work. Quantitative validations allow to have a first idea of the value of our methods but only perceptual evaluations have a real worth in this field. At the time of this writing, we have carried out one perceptual evaluation of our corpus and two other evaluations of the synthesized content are in progress. Those evaluations were sent to the LSF community and their analysis constitutes an interesting perspectives for the work of this thesis.

Moreover, during the course of this thesis, we did our best to propose evaluations adapted to a deaf population by translating the questions into LSF videos with subtitles and by carefully choosing the questions themselves to be visual and illustrated. Not wishing to develop our own platform from scratch, we reviewed the existing evaluation tools and chose the platform proposed by Google (Google Forms [1]), for reasons of price, automation possibilities and the possible number of participants. However, we had great difficulty in adapting it to the specific needs of a more visual deaf community with, in particular, the presence of many videos, drop-down lists with images and interactive images. The development of evaluation platforms dedicated to people with disabilities could be very interesting for research.

Extending the Kinematic Analysis using *MoCap* Data

During this thesis, the analysis of the ground truth took a great importance, whether for the annotation or the synthesis work. We found that the kinematic profiles of the joint trajectories contained a lot of information, even when only considering the position and velocity of the joints. These profiles allowed us to choose the best synthesis approaches and, notably, was determinant for the selection of the sigmoid interpolation profile. *MoCap* is an extremely rich tool for accurate motion analysis: it is already widely used in the medical and physiotherapy fields to analyze postural problems. Using it for statistical analysis of SL data could lead to discoveries in the fields of SL linguistics, SL synthesis, cognitive sciences and biomechanics. Indeed, we have developed a wide range of tools for synthesis and have discovered, as we went along, various other possibilities for the synthesis of new content. However, it is necessary to develop an appropriate dataset, containing the ground truth, for each new idea which was not possible at the time. This thesis therefore

1. `https://www.google.fr/intl/fr/forms/about/`

constituted an exploratory work that could be increased with further analysis and the development of other synthesis tools.

Automating the Synthesis of New Content

The proposals of Chapter 8 are, in part, "proofs of concept" of our phonological recombination theory. For this theory to be more exploitable, we could improve our system of analysis of the kinematic profiles of joint trajectories in order to automatically create *synthesis templates*, either by statistical analysis or by machine learning. A synthesis template would give the structure of a specific sign language mechanism. Therefore, a template would be defined for each sign language mechanism (e.g., dactylology, size specifier, derivational base). It would contain information on the variable and invariant elements of a mechanism (e.g., for dactylology, the invariant parts would be a null motion, and a placement in front of the signer, the variables would be the nature of the hand configurations), the sources of the different components (IK, interpolation or database, for example) and the synchronization between the elements. Thus, only the specification of the type of mechanism and the variable elements would be necessary to build new content. In the thesis, this is done manually using the knowledge extracted from the ground truth. An automation of this synthesis template would be a definite improvement.

Connecting our System to Existing Specification Systems

Finally, in order to be able to exploit our synthesis proposal and to develop its use and applications (translations, SL teaching apps), it would be very interesting to be able to make our synthesis system co-exist with a specification language at the sign or sentence level. This language could be the output of a machine translation module, for example. Our work focuses on the synthesis of SL content from the specification of signs and utterances at a phonological level (i.e., low level). Connecting it to a higher level system would therefore increase the applications of our system and could, for example, allow our system to be used for machine translation.

BIBLIOGRAPHY

[1] A. Millet and A. Morgenstern, *Grammaire descriptive de la langue des signes française: dynamiques iconiques et linguistique générale*. UGA Editions, 2019.

[2] J. Woodward, *How you gonna get to heaven if you can't talk with Jesus: On depathologizing deafness*. TJ Publishers, 1982.

[3] C. Cuxac, *La langue des signes française (LSF) : les voies de l'iconocité (French) [French Sign Language: the iconicity ways]*, ser. Faits de langues. Ophrys, 2000, ISBN: 9782708009523. [Online]. Available: `https://books.google.fr/books?id=UuS7AAAAIAAJ`.

[4] A. Martinet, « La double articulation linguistique », *Travaux du Cercle linguistique de Copenhague*, vol. 5, no. 30-37, 1949.

[5] W. C. Stokoe, « Sign language structure: An outline of the visual communication systems of the American deaf », *Studies in Linguistics, Occasional Papers*, vol. 8, 1960.

[6] R. Battison, *Lexical borrowing in American sign language.* ERIC, 1978.

[7] E. S. Klima and U. Bellugi, *The signs of language*. Harvard University Press, 1979.

[8] L. Boutora, « Vers un inventaire ordonné des configurations manuelles de la Langue des Signes Française », in *Journées d'Études sur la Parole (JEP)*, 2006, pp. 12–16.

[9] B. Moody, *La langue des signes, Tome 1 : Histoire et grammaire (French) [French Sign Language - First Volume: History and grammar]*. International Visual Theatre (IVT), 1983.

[10] R. Battison, « Phonological deletion in american sign language », *Sign language studies*, vol. 5, no. 1, pp. 1–19, 1974.

[11] A. Millet, « La langue des signes française (LSF): une langue iconique et spatiale méconnue », *Recherche et pratiques pédagogiques en langues de spécialité. Cahiers de l'Apliut*, vol. 23, no. 2, pp. 31–44, 2004.

[12] W. Sandler and D. Lillo-Martin, *Sign language and linguistic universals*. Cambridge University Press, 2006.

[13] M. Huenerfauth, P. Lu, and H. Kacorri, « Synthesizing and Evaluating Animations of American Sign Language Verbs Modeled from Motion-Capture Data », *6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, pp. 22–28, 2015.

[14] R. Wolfe, P. Cook, J. C. McDonald, and J. Schnepp, « Linguistics as structure in computer animation: Toward a more effective synthesis of brow motion in American Sign Language », *Sign Language & Linguistics*, vol. 14, no. 1, pp. 179–199, 2011.

[15] M Sallandre, « Simultaneity in French sign language discourse », *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, vol. 281, p. 103, 2007.

[16] F. De Saussure, *Cours de linguistique générale*. Otto Harrassowitz Verlag, 1916, vol. 1.

[17] M.-A. Sallandre, « Les unités du discours en Langue des Signes Française. Tentative de catégorisation dans le cadre d'une grammaire de l'iconicité. », PhD thesis, Université Paris VIII Vincennes-Saint Denis, 2003.

[18] A. Collomb, A. Braffort, and S. Kahane, « L'anatomie du proforme en langue des signes française:. Quand il sert à introduire des entités dans le discours », *TIPA. Travaux interdisciplinaires sur la parole et le langage*, no. 34, 2018.

[19] B. Schick, « The acquisition of classifier predicates in American Sign Language », *ETD Collection for Purdue University*, Jan. 1987.

[20] T. Supalla, « The classifier system in American sign language », *Noun classes and categorization*, pp. 181–214, 1986.

[21] F. De Saussure, *Cours de linguistique generale. 1st*, 1964.

[22] C. Cuxac, « Compositionnalité sublexicale morphémique-iconique en langue des signes française », *Recherches linguistiques de Vincennes*, no. 29, pp. 55–72, 2000.

[23] A. Morgenstern, S. Caët, M. Collombel-Leroy, F. Limousin, and M. Blondel, « From gesture to sign and from gesture to word: Pointing in deaf and hearing children », *Gesture*, vol. 10, no. 2-3, pp. 172–202, 2010.

[24] L. De Beuzeville, T. Johnston, and A. C. Schembri, « The use of space with indicating verbs in Auslan: A corpus-based investigation », *Sign Language & Linguistics*, vol. 12, no. 1, pp. 53–82, 2009.

[25] E. Ormel, O. Crasborn, G. Kootstra, and A. de Meijer, « Coarticulation of handshape in Sign Language of the Netherlands: A corpus study », *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, vol. 8, no. 1, 2017.

[26] S. Ojala, T. Salakoski, and O. Aaltonen, « Coarticulation in sign and speech », 2009.

[27] C. Cuxac, A. Braffort, A. Choisier, C. Collet, P. Dalle, I. Fusellier, G. Jirou, F. Lejeune, B. Lenseigne, N. Monteillard, *et al.*, *Corpus LS-COLIN*, 2002.

[28] M. Blondel, « Acquisition bilingue LSF-français: L'enfant qui grandit avec deux langues et dans deux modalités », *Acquisition et interaction en langue étrangère*, no. Aile... Lia 1, pp. 169–194, 2009.

[29] A. Millet, « Le jeu syntaxique des proformes et des espaces dans la cohésion narrative en LSF », *Glottopol*, vol. 7, pp. 96–111, 2006.

[30] T. Hanke, R. Konrad, G. Langer, A. Müller, and S. Wähl, *Detecting Regional and Age Variation in a Growing Corpus of DGS*, 2017.

[31] A. Schembri and T. Johnston, « Sociolinguistic variation in Auslan (Australian Sign Language): A research project in progress », *Deaf Worlds*, vol. 20, no. 1, S78–S90, 2004.

[32] J. Mesch and L. Wallin, « Gloss annotations in the Swedish Sign Language corpus », *International Journal of Corpus Linguistics*, vol. 20, no. 1, pp. 102–120, 2015.

[33] A. Balvet, C. Courtin, D. Boutet, C. Cuxac, I. Fusellier-Souza, B. Garcia, M. T. L'Huillier, and M. A. Sallandre, « The Creagest Project: a digitized and annotated corpus for French sign language (LSF) and natural gestural languages », in *LREC 2010*, 2010.

[34] M. Blondel, L. Boutora, A.-M. Parisot, and S. Villeneuve, « Étude exploratoire de marqueurs intonatifs en LSF et LSQ », in *Traitement Automatique des Langues Naturelles*, ser. TALN, Avignon, France, Jun. 2008. [Online]. Available: `https://hal.archives-ouvertes.fr/hal-01803190`.

[35] O. A. Crasborn and I. Zwitserlood, « The Corpus NGT: an online corpus for professionals and laymen », in *3rd Workshop on the Representation and Processing of Sign Languages*, Paris ELRA, 2008.

[36] A. M. Martínez, R. B. Wilbur, R. Shay, and A. C. Kak, « Purdue RVL-SLLL ASL database for automatic recognition of American Sign Language », in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, IEEE, 2002, pp. 167–172.

[37] P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff, and H. Ney, « Benchmark Databases for Video-Based Automatic Sign Language Recognition. », in *LREC*, 2008.

[38] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, « Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. », in *LREC*, 2014, pp. 1911–1916.

[39] U. Von Agris, M. Knorr, and K.-F. Kraiss, « The significance of facial features for automatic sign language recognition », in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, IEEE, 2008, pp. 1–6.

[40] N. C. Camgöz, A. A. Kındıroğlu, S. Karabüklü, M. Kelepir, A. S. Özsoy, and L. Akarun, « BosphorusSign: a Turkish sign language recognition corpus in health and finance domains », in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 1383–1388.

[41] S. Ebling, N. C. Camgöz, P. B. Braem, K. Tissi, S. Sidler-Miserez, S. Stoll, S. Hadfield, T. Haug, R. Bowden, S. Tornay, *et al.*, « SMILE Swiss German sign language dataset », in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[42] Y.-H. Chiu, C.-H. Wu, H.-Y. Su, and C.-J. Cheng, « Joint optimization of word alignment and epenthesis generation for Chinese to Taiwanese sign synthesis », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 28–39, 2006.

[43] A. Almohimeed, M. Wald, and R. Damper, « An Arabic Sign Language corpus for instructional language in school », 2010.

[44] S. Morrissey, H. Somers, R. Smith, S. Gilchrist, and S. Dandapat, « Building a Sign Language corpus for use in Machine Translation », in *Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.

[45] R. San-Segundo, J. M. Montero, R Córdoba, V. Sama, F Fernández, L. D'Haro, V. López-Ludeña, D Sánchez, and A. García, « Design, development and field evaluation of a Spanish into sign language translation system », *Pattern Analysis and Applications*, vol. 15, no. 2, pp. 203–224, 2012.

[46] J. Bungeroth, D. Stein, P. Dreuw, H. Ney, S. Morrissey, A. Way, and L. van Zijl, « The ATIS Sign Language Corpus », in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, European Language Resources Association, 2008. [Online]. Available: `http://www.lrec-conf.org/proceedings/lrec2008/summaries/748.html`.

[47] E. Efthimiou, S. Fontinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and F. Goudenove, « Dicta-sign–sign language recognition, generation and modelling: a research effort with applications in deaf communication », in *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010, pp. 80–83.

[48] E. Efthimiou, S.-E. Fotinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and F. Lefebvre-Albaret, « Sign Language technologies and resources of the Dicta-Sign project », in *Proc. of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon. LREC*, 2012, pp. 23–27.

[49] E. Efthimiou and S.-E. Fotinea, « GSLC: creation and annotation of a Greek sign language corpus for HCI », in *International Conference on Universal Access in Human-Computer Interaction*, Springer, 2007, pp. 657–666.

[50] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, « OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields », in *arXiv preprint arXiv:1812.08008*, 2018.

[51] E. Ormel, O. Crasborn, and E. van der Kooij, « Coarticulation of hand height in Sign Language of the Netherlands is affected by contact type », *Journal of Phonetics*, vol. 41, no. 3, pp. 156–171, 2013.

[52] L. Naert, C. Larboulette, and S. Gibet, « Coarticulation analysis for sign language synthesis », in *International Conference on Universal Access in Human-Computer Interaction*, Springer, 2017, pp. 55–75.

[53] P. Lu and M. Huenerfauth, « CUNY American Sign Language motion-capture corpus: first release », in *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, The 8th International Conference on Language Resources and Evaluation (LREC 2012)*, 2012.

[54] S. Gibet, A. Héloir, N. Courty, J.-F. Kamp, P. Gorce, N. Rezzoug, F. Multon, and C. Pelachaud, « Virtual agent for deaf signing gestures », *AMSE, Journal of the Association for the Advancement of Modelling and Simulation Techniques in Enterprises (Special edition HANDICAP)*, vol. 67, pp. 127–136, 2006.

[55] A. Braffort, M. Benchiheub, and B. Berret, « APLUS: A 3D Corpus of French Sign Language », in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility, ASSETS 2015, Lisbon, Portugal, October 26-28, 2015*, Y. Yesilada and J. P. Bigham, Eds., ACM, 2015, pp. 381–382. DOI: 10.1145/2700648.2811380. [Online]. Available: https://doi.org/10.1145/2700648.2811380.

[56] M.-e.-F. Benchiheub, B. Berret, and A. Braffort, « Collecting and Analysing a Motion-Capture Corpus of French Sign Language », in *Workshop on the Representation and Processing of Sign Languages*, Portoroz, Slovenia, Jan. 2016.

[57] K. Duarte, « Motion Capture and avatars as Portals for Analyzing the Linguistic Structure of Sign Languages », PhD thesis, Université Bretagne Sud, 2012.

[58] A. Heloir, S. Gibet, F. Multon, and N. Courty, « Captured motion data processing for real time synthesis of sign language », in *International Gesture Workshop*, Springer, 2005, pp. 168–171.

[59] J. M. De Martino, P. D. P. Costa, A Benetti, L. A. Rosa, K. M. O. Kumada, and I. Silva, « Building a brazilian portuguese-brazilian sign language parallel corpus using motion capture data », in *Proceedings of Workshop of Corpora and Tools for Processing Corpora*, 2016, pp. 56–63.

[60] S. Gibet, « Building French Sign Language Motion Capture Corpora for Signing Avatars », in *Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, LREC 2018*, Miyazaki, Japan, May 2018.

[61] H. Brock and K. Nakadai, « Deep JSLC: A multimodal corpus collection for data-driven generation of Japanese sign language expressions », in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[62] A. Schembri, J. Fenlon, R. Rentelis, S. Reynolds, and K. Cormier, « Building the British sign language corpus », *Language Documentation & Conservation*, vol. 7, pp. 136–154, 2013.

[63] R. Nishio, S.-E. Hong, S. König, R. Konrad, G. Langer, T. Hanke, and C. Rathmann, « Elicitation methods in the DGS (German Sign Language) corpus project », in *Workshop on the Representation and Processing of Signed Languages, LREC*, 2010, pp. 178–185.

[64] S. Gibet, N. Courty, K. Duarte, and T. Le Naour, « The SignCom System for Data-Driven Animation of Interactive Virtual Signers: Methodology and Evaluation », *Transactions on Interactive Intelligent Systems*, 2011.

[65] P. Lu and M. Huenerfauth, « Collecting a motion-capture corpus of American Sign Language for data-driven generation research », in *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*, Association for Computational Linguistics, 2010, pp. 89–97.

[66] ——, « Collecting and evaluating the CUNY ASL corpus for research on American Sign Language animation », *Computer Speech & Language*, vol. 28, no. 3, pp. 812–831, 2014.

[67] S. Alexanderson and J. Beskow, « Towards Fully Automated Motion Capture of Signs–Development and Evaluation of a Key Word Signing Avatar », *ACM Transactions on Accessible Computing (TACCESS)*, vol. 7, no. 2, p. 7, 2015.

[68] Limsi and CIAMS, *MOCAP1*, ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr, 2017. [Online]. Available: `https://hdl.handle.net/11403/mocap1/v1`.

[69] S. Gibet, F. Lefebvre-Albaret, L. Hamon, R. Brun, and A. Turki, « Interactive editing in French Sign Language dedicated to virtual signers: requirements and challenges », *Universal Access in the Information Society*, vol. 15, no. 4, pp. 525–539, 2016.

[70] T. Johnston and L De Beuzeville, « Auslan corpus annotation guidelines », *Centre for Language Sciences, Department of Linguistics, Macquarie University*, 2014.

[71] M. P. I. for Psycholinguistics, *ELAN v.4.9.4*, `http://tla.mpi.nl/tools/tla-tools/elan/`, 2017.

[72] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, « ELAN: a professional framework for multimodality research », in *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006, pp. 1556–1559.

[73] M. Kipp, « Anvil - A generic annotation tool for multimodal dialogue », in *Seventh European Conference on Speech Communication and Technology*, 2001.

[74] A. Millet and I. Estève, « Segmenter et annoter le discours d'un locuteur de LSF: permanence formelle et variablité fonctionnelle des unités (Segment and annotate a discourse of a French Sign Language speaker: formal continuity and fonctional variation of units)[in French] », in *JEP-TALN-RECITAL 2012, Workshop DEGELS 2012: Défi GEste Langue des Signes (DEGELS 2012: Gestures and Sign Language Challenge)*, 2012, pp. 57–72.

[75] P. Dreuw and H. Ney, « Towards automatic sign language annotation for the elan tool », in *3rd Workshop on the Representation and Processing of Sign Languages*, 2008.

[76] T. Hanke, S. Matthes, A. Regen, and S. Worseck, « Where Does a Sign Start and End? Segmentation of Continuous Signing », *Language Resources and Evaluation Conference*, 2012.

[77] J. Lin, M. Karg, and D. Kulic, « Movement Primitive Segmentation for Human Motion Modeling: A Framework for Analysis », *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 1–15, Jan. 2016. DOI: `10.1109/THMS.2015.2493536`.

[78] A. Fod, M. Mataric, and O. Jenkins, « Automated Derivation of Primitives for Movement Classification », *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[79] K. Förger, C. Jouflineau, and A. Bachrach, « Kinetic predictors of spectators' segmentation of a live dance performance », in *MOCO 2017*, 2017.

[80] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, « Exact discovery of time series motifs », in *Proceedings of the 2009 SIAM international conference on data mining*, SIAM, 2009, pp. 473–484.

[81] C. Lins, S. M. Müller, M. Pfingsthorn, M. Eichelberg, A. Gerka, and A. Hein, « Unsupervised Temporal Segmentation of Skeletal Motion Data using Joint Distance Representation. », in *HEALTHINF*, 2018, pp. 478–485.

[82] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, « Skeleton-based dynamic hand gesture recognition », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9.

[83] N. Koenig and M. J. Mataric, « Behavior-based segmentation of demonstrated tasks », in *Proceedings of the international conference on development and learning*, Citeseer, 2006.

[84] K. Papoutsakis, C. Panagiotakis, and A. A. Argyros, « Temporal action co-segmentation in 3d motion capture data and videos », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6827–6836.

[85] J. Barbic, A. Safonova, J. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, « Segmenting Motion Capture Data into Distinct Behaviors », in *Proceedings of the Graphics Interface 2004 Conference, May 17-19, 2004, London, Ontario, Canada*, W. Heidrich and R. Balakrishnan, Eds., ser. ACM International Conference Proceeding Series, vol. 62, Canadian Human-Computer Communications Society, 2004, pp. 185–194.

[86] F. Bashir, W. Qu, A. Khokhar, and D. Schonfeld, « HMM-based motion recognition system using segmented PCA », in *IEEE International Conference on Image Processing 2005*, IEEE, vol. 3, 2005, pp. 1288–1291.

[87] M. Brand and V. M. Kettnaker, « Discovery and Segmentation of Activities in Video », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 844–851, 2000. DOI: 10.1109/34.868685. [Online]. Available: https://doi.org/10.1109/34.868685.

[88] D. Endres, A. Christensen, L. Omlor, and M. Giese, « Segmentation of Action Streams Human Observers vs. Bayesian Binning », in *Proceedings of KI 2011: Advances in Artificial Intelligence, 34th Annual German Conference on AI*, Berlin, Germany, 2011, pp. 75–86.

[89] D. Gong, G. Medioni, S. Zhu, and X. Zhao, « Kernelized Temporal Cut for Online Temporal Segmentation and Recognition », in *Proceedings of the 12th European*

Conference on Computer Vision ECCV - Part III, Florence, Italy, 2012, pp. 229–243.

[90] J. Kohlmorgen and S. Lemm, « A dynamic hmm for on-line segmentation of sequential data », in *Advances in neural information processing systems*, 2002, pp. 793–800.

[91] D. Kulic, W. Takano, and Y. Nakamura, « Online segmentation and clustering from continuous observation of whole body motions », *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.

[92] F. Zhou, F. De la Torre, and J. K. Hodgins, « Aligned cluster analysis for temporal segmentation of human motion », in *2008 8th IEEE international conference on automatic face & gesture recognition*, IEEE, 2008, pp. 1–7.

[93] M. Müller, A. Baak, and H.-P. Seidel, « Efficient and Robust Annotation of Motion Capture Data », *Symposium on Computer Animation*, 2009.

[94] F. Lv and R. Nevatia, « Recognition and segmentation of 3-d human action using hmm and multi-class adaboost », in *European conference on computer vision*, Springer, 2006, pp. 359–372.

[95] I. S. Vicente, V. Kyrki, D. Kragic, and M. Larsson, « Action recognition and understanding through motor primitives », *Advanced Robotics*, vol. 21, no. 15, pp. 1687–1707, 2007.

[96] A. Y. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari, « Distributed segmentation and classification of human actions using a wearable motion sensor network », in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2008, pp. 1–8.

[97] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll, « Human activity recognition in the context of industrial human-robot interaction », in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 2014, pp. 1–10. DOI: 10.1109/APSIPA.2014.7041588.

[98] C. Pérez-D'Arpino and J. Shah, « Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification », in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6175–6182. DOI: 10.1109/ICRA.2015.7140066.

[99] H. Liu and L. Wang, « Gesture recognition for human-robot collaboration: A review », *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[100] R. Yang and S. Sarkar, « Detecting Coarticulation in Sign Language Using Conditional Random Fields », in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 2, Washington, DC, USA: IEEE Computer Society, 2006, pp. 108–112.

[101] J.-B. Kim, K.-H. Park, W.-C. Bang, and Z. Bien, « Continuous Korean Sign Language Recognition Using Gesture Segmentation and Hidden Markov Model », *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, 2001.

[102] Y. Ye, Y. Tian, M. Huenerfauth, and J. Liu, « Recognizing American Sign Language Gestures from within Continuous Videos », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2064–2073.

[103] F. c. Lefebvre-Albaret, F. Gianni, and P. Dalle, « Toward a computer-aided sign segmentation », *Language Resources and Evaluation Conference (LREC). European Language Resources Association*, 2008.

[104] M. Gonzalez, « Un système de segmentation automatique de gestes appliqué à la langue des signes (French) [An automated gesture segmentation system applied to sign language] », in *JEP-TALN-RECITAL*, 2012, 93–98.

[105] S. Nayak, S. Sarkar, and B. Loeding, « Automated extraction of signs from continuous sign language sentences using iterated conditional modes », in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 2583–2590.

[106] M. J. Cheok, Z. Omar, and M. H. Jaward, « A review of hand gesture and sign language recognition techniques », *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, 2019.

[107] N. B. Ibrahim, H. H. Zayed, and M. M. Selim, « Advances, Challenges and Opportunities in Continuous Sign Language Recognition », *Journal of Engineering and Applied Sciences*, vol. 15, no. 5, pp. 1205–1227, 2020.

[108] M. M. Zaki and S. I. Shaheen, « Sign language recognition using a combination of new vision based features », *Pattern Recognition Letters*, vol. 32, no. 4, pp. 572–577, 2011.

[109] K. Grobel and M. Assan, « Isolated sign language recognition using hidden Markov models », in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, IEEE, vol. 1, 1997, pp. 162–167.

[110] C. Vogler and D. Metaxas, « A framework for recognizing the simultaneous aspects of american sign language », *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 358–384, 2001.

[111] M. Dilsizian, P. Yanovich, S. Wang, C. Neidle, and D. 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: European Language Resources Association (ELRA), 2014, ISBN: 978-2-9517408-8-4.

[112] D. Metaxas, M. Dilsizian, and C. Neidle, « Linguistically-driven Framework for Computationally Efficient and Scalable Sign Recognition », in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds., Miyazaki, Japan: European Language Resources Association (ELRA), 2018, ISBN: 979-10-95546-00-9.

[113] W. Kong and S. Ranganath, « Towards subject independent continuous sign language recognition: A segment and merge approach », *Pattern Recognition*, vol. 47, no. 3, pp. 1294–1308, 2014.

[114] O. Koller, J. Forster, and H. Ney, « Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers », *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015.

[115] A. Heloir, S. Gibet, F. Multon, and N. Courty, « Captured Motion Data Processing for Real Time Synthesis of Sign Language », *Motion in Games*, 2005.

[116] J. G. Abreu, J. M. Teixeira, L. S. Figueiredo, and V. Teichrieb, « Evaluating sign language recognition using the myo armband », in *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, IEEE, 2016, pp. 64–70.

[117] A. Karami, B. Zanj, and A. K. Sarkaleh, « Persian sign language (PSL) recognition using wavelet transform and neural networks », *Expert Systems with Applications*, vol. 38, no. 3, pp. 2661–2667, 2011.

[118] B. Gupta, P. Shukla, and A. Mittal, « K-nearest correlated neighbor classification for Indian sign language gesture recognition using feature fusion », in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, 2016, pp. 1–5.

[119] R. Jayaprakash, J. Bhattacharya, and S. Majumder, « Shape, texture and local movement hand gesture features for indian sign language recognition », in *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*, IEEE, 2011, pp. 30–35.

[120] K. Symeonidis, « Hand gesture recognition using neural networks », *Master Thesis*, 2000.

[121] C. Reverdy, S. Gibet, C. Larboulette, and P.-F. Marteau, « Un système de synthèse et d'annotation automatique à partir de données capturées pour l'animation faciale expressive en LSF », in *Journées Françaises d'Informatique Graphique (AFIG)*, 2016.

[122] H. Kacorri, « TR-2015001: A Survey and Critique of Facial Expression Synthesis in Sign Language Animation », CUNY Academic Works. `https://academicworks.cuny.edu/gc_cs_tr/403`, 2015.

[123] S. Prillwitz and H. Z. für Deutsche Gebärdensprache und Kommunikation Gehörloser, *HamNoSys: Version 2.0; Hamburg Notation System for Sign Languages; An Introductory Guide*. Signum-Verlag, 1989.

[124] A. Othman and M. Jemni, « Statistical sign language machine translation: from english written text to american sign language gloss », *arXiv preprint arXiv:1112.0168*, 2011.

[125] A. Braffort, L. Bolot, M. Filhol, and C. Verrecchia, « Démonstrations d'Elsi, la signeuse virtuelle du LIMSI », in *Conférence sur le Traitement Automatique des Langues Naturelles (TALN), Traitement automatique des langues des signes (atelier TALS)*, Toulouse, France, 2007. [Online]. Available: `https://hal.archives-ouvertes.fr/hal-02285133`.

[126] A. Bébian, *Mimographie, ou Essai d'écriture mimique propre à régulariser le langage des sourds-muets*. L. Colas, 1825.

[127] W. C. Stokoe, D. C. Casterline, and C. G. Croneberg, *A dictionary of American Sign Language on linguistic principles*. Linstok Press, 1976.

[128] V. Sutton, *Sign writing for everyday use*. Sutton Movement Writing Press, 1981.

[129] M. Kato, « A study of notation and sign writing systems for the deaf », *Intercultural Communication Studies*, vol. 17, no. 4, pp. 97–114, 2008.

[130] A. S. C. Lessa-de Oliveira, « Libras escrita: o desafio de representar uma língua tridimensional por um sistema de escrita linear », *ReVEL*, vol. 10, no. 19, 2012.

[131] S. K. Liddell and R. E. Johnson, « American sign language: The phonological base », *Sign language studies*, vol. 64, no. 1, pp. 195–277, 1989.

[132] R. E. Johnson and S. K. Liddell, « A segmental framework for representing signs phonetically », *Sign Language Studies*, vol. 11, no. 3, pp. 408–463, 2011.

[133] ——, « Toward a phonetic representation of hand configuration: The fingers », *Sign Language Studies*, vol. 12, no. 1, pp. 5–45, 2011.

[134] ——, « Toward a phonetic representation of hand configuration: The thumb », *Sign Language Studies*, vol. 12, no. 2, pp. 316–333, 2012.

[135] O. Tkachman, K. C. Hall, A. Xavier, and B. Gick, « Sign Language Phonetic Annotation meets Phonological CorpusTools: Towards a sign language toolset for phonetic notation and phonological analysis », in *Proceedings of the Annual Meetings on Phonology*, vol. 3, 2016.

[136] P. Eccarius and D. Brentari, « Handshape coding made easier: A theoretically based notation for phonological transcription », *Sign Language & Linguistics*, vol. 11, no. 1, pp. 69–101, 2008.

[137] A. C. da Rocha Costa and G. P. Dimuro, « Signwriting-based sign language processing », in *International Gesture Workshop*, Springer, 2001, pp. 202–205.

[138] R. Kennaway, « Avatar-independent scripting for real-time gesture animation », *arXiv preprint arXiv:1502.02961*, 2006.

[139] ——, « Experience with and Requirements for a Gesture Description Language for Synthetic Animation », in *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 300–311.

[140] J. Glauert and R. Elliott, « Extending the SiGML notation—a progress report », in *Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, vol. 23, 2011.

[141] W. M. do Amaral, J. M. De Martino, and L. M. G. Angare, « Sign language 3D virtual agent », in *Education and Information Systems, Technologies and Applications Conference*, 2011.

[142] J. M. De Martino, I. R. Silva, C. Z. Bolognini, P. D. P. Costa, K. M. O. Kumada, L. C. Coradine, P. H. da Silva Brito, W. M. do Amaral, Â. B. Benetti, E. T. Poeta, *et al.*, « Signing avatars: making education more inclusive », *Universal Access in the Information Society*, pp. 1–16, 2016.

[143] T. Lebourque and S. Gibet, « High level specification and control of communication gestures: the GessyCA system », in *Computer Animation, 1999. Proceedings*, IEEE, 1999, pp. 24–35.

[144] O. Losson, « Modélisation du geste communicatif et réalisation d'un signeur virtuel de phrases en langue des signes grançaise », Theses, Université des Sciences et Technologie de Lille - Lille I, Jan. 2000. [Online]. Available: `https://tel.archives-ouvertes.fr/tel-00003332`.

[145] A. Heloir and M. Kipp, « Real-time animation of interactive agents: Specification and realization », *Applied Artificial Intelligence*, vol. 24, no. 6, pp. 510–529, 2010.

[146] M. Kipp, A. Heloir, and Q. Nguyen, « Sign Language Avatars: Animation and Comprehensibility », in *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, 2011.

[147] M. Filhol, « Modèle descriptif des signes pour un traitement automatique des langues des signes. (A descriptive model of signs for Sign Language processing) », PhD thesis, University of Paris-Sud, Orsay, France, 2008. [Online]. Available: `https://tel.archives-ouvertes.fr/tel-00300591`.

[148] S. Gibet, T. Lebourque, and P.-F. Marteau, « High-level specification and animation of communicative gestures », *Journal of Visual Languages & Computing*, vol. 12, no. 6, pp. 657–687, 2001.

[149] Autodesk, *Maya*, `https://www.autodesk.fr/products/maya/overview`, Accessed: 2019-11-12.

[150] Blender, *Blender*, `https://www.blender.org/`, Accessed: 2019-11-12.

[151] M. Shantz and H. Poizner, « A computer program to synthesize American Sign Language », *Behavior Research Methods & Instrumentation*, vol. 14, no. 5, pp. 467–474, 1982, ISSN: 1554-3528. DOI: 10.3758/BF03203314. [Online]. Available: http://dx.doi.org/10.3758/BF03203314.

[152] O. Losson and J.-M. Vannobel, « Sign language formal description and synthesis », in *Proc. 2. Euro. Conf. Disability, Virtual Reality & Assoc. Tech., Skövde, Sweden*, 1998.

[153] J. McDonald, R. Wolfe, J. Schnepp, J. Hochgesang, D. G. Jamrozik, M. Stumbo, L. Berke, M. Bialek, and F. Thomas, « An automated technique for real-time production of lifelike animations of American Sign Language », *Universal Access in the Information Society*, vol. 15, no. 4, pp. 551–566, 2016, ISSN: 1615-5297. DOI: 10.1007/s10209-015-0407-2. [Online]. Available: http://dx.doi.org/10.1007/s10209-015-0407-2.

[154] J. McDonald, R. Wolfe, R. B. Wilbur, R. Moncrief, E. Malaia, S. Fujimoto, S. Baowidan, and J. Stec, « A new tool to facilitate prosodic analysis of motion capture data and a data-driven technique for the improvement of avatar motion », in *Proceedings of Language Resources and Evaluation Conference (LREC)*, 2016, pp. 153–59.

[155] V. Lombardo, C. Battaglino, R. Damiano, and F. Nunnari, « An avatar-based interface for the Italian sign language », in *Complex, Intelligent and Software Intensive Systems (CISIS), 2011 International Conference on*, IEEE, 2011, pp. 589–594.

[156] V. Lombardo, F. Nunnari, and R. Damiano, « A virtual interpreter for the Italian sign language », in *International Conference on Intelligent Virtual Agents*, Springer, 2010, pp. 201–207.

[157] J. Ségouat, « Modélisation de la coarticulation en Langue des Signes Française pour la diffusion automatique d'informations en gare ferroviaire à l'aide d'un signeur virtuel. (French) [Modelling coarticulation in LSF for automatic broadcast of information in train stations using an avatar] », PhD thesis, Université Paris Sud - Paris XI, 2010.

[158] A. Irving and R. Foulds, « A parametric approach to sign language synthesis », in *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, ACM, 2005, pp. 212–213.

[159] R. Yorganci, A. A. Kindiroglu, and H. Kose, « Avatar-based Sign Language Training Interface for Primary School Education », 2016.

[160] A. B. Grieve-Smith, « SignSynth: A Sign Language Synthesis Application Using Web3D and Perl », in *Gesture and Sign Language in Human-Computer Interaction*, I. Wachsmuth and T. Sowa, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 134–145, ISBN: 978-3-540-47873-7.

[161] M. Papadogiorgaki, N. Grammalidis, D. Tzovaras, and M. G. Strintzis, « Text-to-sign language synthesis tool », in *Signal Processing Conference, 2005 13th European*, IEEE, 2005, pp. 1–4.

[162] Z. Krňoul, J. Kanis, M. Železnỳ, and L. Müller, « Czech text-to-sign speech synthesizer », in *International Workshop on Machine Learning for Multimodal Interaction*, Springer, 2007, pp. 180–191.

[163] S.-E. Fotinea, E. Efthimiou, G. Caridakis, and K. Karpouzis, « A knowledge-based sign synthesis architecture », *Universal Access in the Information Society*, vol. 6, no. 4, pp. 405–418, 2008.

[164] VCom3D, *Sign 4 Me*, `http://www.vcom3d.com/language/sign-4-me/`, Accessed: 2019-09-08, 2018.

[165] M. Delorme, M. Filhol, and A. Braffort, « Animation generation process for sign language synthesis », in *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*, IEEE, 2009, pp. 386–390.

[166] O. Losson and J.-M. Vannobel, « Sign specification and synthesis », in *International Gesture Workshop*, Springer, 1999, pp. 239–251.

[167] T. Lebourque, « Specification et generation de gestes naturels », PhD thesis, Paris 11, 1998.

[168] R. Kennaway, « Synthetic animation of deaf signing gestures », in *International Gesture Workshop*, Springer, 2001, pp. 146–157.

[169] R. Smith, S. Morrissey, and H. Somers, « HCI for the Deaf community: Developing human-like avatars for sign language synthesis », 2010.

[170] J. McDonald, R. Wolfe, S. Johnson, S. Baowidan, R. Moncrief, and N. Guo, « An Improved Framework for Layering Linguistic Processes in Sign Language Generation: Why There Should Never Be a Brows Tier », in *International Conference on Universal Access in Human-Computer Interaction*, Springer, 2017, pp. 41–54.

[171] M. Kapadia, I.-k. Chiang, T. Thomas, N. I. Badler, J. T. Kider Jr, *et al.*, « Efficient motion retrieval in large motion databases », in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 2013, pp. 19–28.

[172] C. Awad, N. Courty, K. Duarte, T. Le Naour, and S. Gibet, « A Combined Semantic and Motion Capture Database for Real-Time Sign Language Synthesis », in *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, ser. Lecture Notes in Artificial Intelligence, vol. 5773, Berlin, Heidelberg: Springer-Verlag, 2009, pp. 432–38.

[173] L. Hamon, S. Gibet, and S. Boustila, « Interactive editing of utterances in French sign language dedicated to signing avatars (Édition interactive d'énoncés en langue des signes française dédiée aux avatars signeurs) [in French] », in *Traitement Automatique des Langues Naturelles, TALN 2013, Les Sables d'Olonne, France, 17-21 Juin 2013, articles courts*, E. Morin and Y. Estève, Eds., The Association for Computer Linguistics, 2013, pp. 547–554. [Online]. Available: `https://www.aclweb.org/anthology/F13-2006/`.

[174] R. Brun, A. Turki, and A. Laville, « A 3D application to familiarize children with sign language and assess the potential of avatars and motion capture for learning movement », in *Proceedings of the 3rd International Symposium on Movement and Computing*, ACM, 2016, p. 48.

[175] MoCapLab, R. I. de France, and C. Digital, *Sign 360*, `http://www.mocaplab.com/fr/projects/sign-360/`, Accessed: 2018-09-14, 2016.

[176] A. Witkin and Z. Popovic, « Motion warping », in *Siggraph*, vol. 95, 1995, pp. 105–108.

[177] A. Héloir and S. Gibet, « A Qualitative and Quantitative Characterisation of Style in Sign Language Gestures », in *Gesture Workshop*, 2007.

[178] A. Héloir, N. Courty, S. Gibet, and F. Multon, « Temporal alignment of communicative gesture sequences », *Computer Animation and Virtual Worlds*, vol. 17, pp. 347–357, Jul. 2006.

[179] P. Carreno, S. Gibet, and P.-F. Marteau, « Synthèse de mouvements humains par des méthodes basées apprentissage: un état de l'art », *Revue Electronique Francophone d'Informatique Graphique*, vol. 8, no. 1, 2014.

[180] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott, « The Development and Evaluation of a Speech-to-Sign Translation System to Assist Transactions », *International Journal of Human–Computer Interaction*, vol. 16, no. 2, pp. 141–161, 2003. DOI: 10.1207/S15327590IJHC1602_02.

[181] F. Pezeshkpour, I. Marshall, R. Elliott, and A. Bangham, « Development of a legible deaf-signing virtual human », in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 1, 1999, pp. 333–338.

[182] N. Aouiti, M. Jemni, and S. Semreen, « Arab gloss annotation system for Arabic Sign Language », in *2015 5th International Conference on Information Communication Technology and Accessibility (ICTA)*, 2015, pp. 1–6. DOI: 10.1109/ICTA.2015.7426932.

[183] E. Ormel, O. Crasborn, E. van der Kooij, L. van Dijken, E Nauta, J. Forster, and D. Stein, « Glossing a multi-purpose sign language corpus », in *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and sign language technologies*, 2010, pp. 186–191.

[184] F. López-Colino and J. Colás, « The synthesis of LSE classifiers: From representation to evaluation », *Journal of Universal Computer Science*, 2011.

[185] ——, « Spanish sign language synthesis system », *Journal of Visual Languages & Computing*, vol. 23, no. 3, pp. 121–136, 2012.

[186] W. A. Woods, « Transition network grammars for natural language analysis », *Communications of the ACM*, vol. 13, no. 10, pp. 591–606, 1970.

[187] M. Huenerfauth, « Generating American Sign Language classifier predicates for English-to-ASL machine translation », PhD thesis, University of Pennsylvania, 2006.

[188] M. Filhol and G. Falquet, « Synthesising Sign Language from semantics, approaching "from the target and back" », *CoRR*, vol. abs/1707.08041, 2017. arXiv: 1707.08041. [Online]. Available: http://arxiv.org/abs/1707.08041.

[189] F. Nunnari, M. Filhol, and A. Heloir, « Animating AZee Descriptions Using Off-the-Shelf IK Solvers », in *8th Workshop on the Representation and Processing of Sign Languages: Involving the Language Community (SignLang 2018), 11th edition of the Language Resources and Evaluation Conference (LREC 2018)*, 2018, pp. 7–12.

[190] M. Filhol and M. N. Hadjadj, « Juxtaposition as a form feature; syntax captured and explained rather than assumed and modelled », in *Language resources and evaluation conference (LREC), Representation and processing of Sign Languages*, 2016.

[191] M. Filhol, J. McDonald, and R. Wolfe, « Synthesizing Sign Language by Connecting Linguistically Structured Descriptions to a Multi-track Animation System », in *Universal Access in Human–Computer Interaction. Designing Novel Interactions: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017, Proceedings, Part II*. Cham: Springer International Publishing, 2017, pp. 27–40, ISBN: 978-3-319-58703-5. DOI: 10.1007/978-3-319-58703-5_3. [Online]. Available: https://doi.org/10.1007/978-3-319-58703-5_3.

[192] M. Filhol, « A human-editable Sign Language representation for software editing—and a writing system? », *arXiv preprint arXiv:1811.01786*, 2018.

[193] N. Bertoldi, G. Tiotto, P. Prinetto, E. Piccolo, F. Nunnari, V. Lombardo, A. Mazzei, R. Damiano, L. Lesmo, and A. J. D. Príncipe, « On the creation and the annotation of a large-scale Italian-LIS parallel corpus », in *LREC 2010*, 2010.

[194] R. Kennaway, J. R. Glauert, and I. Zwitserlood, « Providing signed content on the Internet by synthesized animation », *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 14, no. 3, p. 15, 2007.

[195] R. Elliott, J. R. Glauert, J. Kennaway, I. Marshall, and E. Safar, « Linguistic modelling and language-processing technologies for Avatar-based sign language presentation », *Universal Access in the Information Society*, vol. 6, no. 4, pp. 375–391, 2008.

[196] M. Filhol and J. Mcdonald, « Extending the AZee-Paula shortcuts to enable natural proform synthesis », in *Workshop on the Representation and Processing of Sign Languages*, Miyazaki, Japan, May 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01848979.

[197] N. Adamo-Villani, K. Hayward, J. Lestina, and R. B. Wilbur, « Effective animation of sign language with prosodic elements for annotation of digital educational content. », in *SIGGRAPH Talks*, 2010.

[198] R. Elliott, J. Bueno, R. Kennaway, and J. Glauert, « Towards the integrationn of synthetic sl annimation with avatars into corpus annotation tools », in *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, Valletta, Malta*, 2010, p. 29.

[199] S. Ebling, « Automatic Translation from German to Synthesized Swiss German Sign Language », PhD thesis, University of Zurich, 2016.

[200] T. Hanke, S. Matthes, A. Regen, J. Storz, S. Worseck, R Eliott, J. Glauert, and R. Kennaway, « Using timing information to improve the performance of avatars », in *Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, 2011.

[201] I. Marshall and E. Safar, « Grammar development for sign language avatar-based synthesis », in *Proceedings HCII*, 2005, pp. 1–10.

[202] S. Verma, P. Bhatia, and S. Kaur, « Online Multilingual Dictionary Using Hamburg Notation for Avatar-Based Indian Sign Language Generation System », in *World Academy of Science, Engineering and Technology International Journal of Cognitive and Language Science*, vol. 12, 2018, pp. 1116–1122. DOI: 10.1999/1307-6892/10009433.

[203] M. Filhol, J. McDonald, and R. Wolfe, « Synthesizing Sign Language by connecting linguistically structured descriptions to a multi-track animation system », in *International Conference on Universal Access in Human-Computer Interaction*, Springer, 2017, pp. 27–40.

[204] D. University, *DePaul ASL Avatar Project*, http://asl.cs.depaul.edu, Accessed: 2020-01-28, 2020.

[205] M. Mori, « The uncanny valley (in Japanese) », in *Energy*, ser. LNCS, vol. 7, Japan, 1970, pp. 33–35.

[206] M. Kipp, Q. Nguyen, A. Heloir, and S. Matthes, « Assessing the Deaf User Perspective on Sign Language Avatars », in *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS '11, Dundee, Scotland, UK: ACM, 2011, pp. 107–114, ISBN: 978-1-4503-0920-2. DOI: 10.1145/2049536.2049557. [Online]. Available: http://doi.acm.org/10.1145/2049536.2049557.

[207] N. Adamo-Villani and S. Anasingaraju, « Toward the Ideal Signing Avatar », *EAI Endorsed Transactions on e-Learning*, vol. 3, no. 11, Jun. 2016. DOI: 10.4108/eai.15-6-2016.151446.

[208] L. Naert, C. Larboulette, and S. Gibet, « *LSF-ANIMAL*: A Motion Capture Corpus in French Sign Language Designed for the Animation of Signing Avatars », in *Language Resources and Evaluation Conference (LREC)*, 2020.

[209] P. Carreno Medrano, « Analysis and Synthesis of Expressive Theatrical Movements », Theses, Université de Bretagne Sud, Nov. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01497531.

[210] C. Reverdy, « Annotation et synthèse baseée donneées des expressions faciales de la Langue des Signes Française », English Title: Data-driven annotation and synthesis of facial expressions in French sign language, PhD thesis, Université Bretagne Sud, 2019. [Online]. Available: http://zabador.free.fr/Publications/2019/Rev19.

[211] F. Amauger, F. Bertin, S. Gonzalez, P. Tsopgni, and A. Vanbrugghe, *Langue des signes Française - A1 (French) [French Sign Language - A1]*, ser. Langue des signes Française. Belin, 2013, ISBN: 978-2-7011-6567-7.

[212] M. Sallandre and C. Cuxac, « Iconicity in Sign Language: A Theoretical and Methodological Point of View », in *Gesture and Sign Languages in Human-Computer Interaction, International Gesture Workshop, GW 2001, London, UK, April 18-20, 2001, Revised Papers*, I. Wachsmuth and T. Sowa, Eds., ser. Lecture Notes in Computer Science, vol. 2298, Springer, 2001, pp. 173–180. DOI: 10.1007/3-540-47873-6_18. [Online]. Available: https://doi.org/10.1007/3-540-47873-6_18.

[213] N. Courty and S. Gibet, « Why is the Creation of a Virtual Signer Challenging Computer Animation ? », in *Motion in Games 2010*, ser. LNCS, Netherlands, 2010, pp. 1–11. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00516624.

[214] P. Carreno, « Marker-set specification for Magician's motion capture database recording », 2015.

[215] C. Reverdy, S. Gibet, and C. Larboulette, « Optimal marker set for motion capture of dynamical facial expressions », in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, ACM, 2015, pp. 31–36.

[216] L. Hoyet, K. Ryall, R. McDonnell, and C. O'Sullivan, « Sleight of hand: perception of finger motion from reduced marker sets », in *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, ACM, 2012, pp. 79–86.

[217] Autodesk, *MotionBuilder*, `https://www.autodesk.com/products/motionbuilder/overview`, Accessed: 2019-04-19.

[218] Qualisys, *Qualisys Track Manager*, `https://www.qualisys.com/software/qualisys-track-manager/`, Accessed: 2020-01-14.

[219] T. Le Naour, N. Courty, and S. Gibet, « Kinematic driven by distances », working paper or preprint, Jul. 2018, [Online]. Available: `https://hal.archives-ouvertes.fr/hal-01838784`.

[220] L. Naert, C. Reverdy, C. Larboulette, and S. Gibet, « Per channel automatic annotation of sign language motion capture data », in *Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, LREC 2018*, 2018.

[221] L. Naert, C. Larboulette, and S. Gibet, « Annotation automatique des configurations manuelles de la Langue des Signes Française à partir de données capturées », in *Journees Françaises d'Informatique Graphique*, Rennes, France, Oct. 2017. [Online]. Available: `https://hal.archives-ouvertes.fr/hal-01649769`.

[222] F. Lefebvre-Albaret, S. Gibet, A. Turki, L. Hamon, and R. 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 (SLTAT) 2013*, Chicago, United States, Oct. 2013. [Online]. Available: `https://hal.archives-ouvertes.fr/hal-00914661`.

[223] Vicon, *Vicon*, `https://www.vicon.com/`, Accessed: 2020-01-14.

[224] Qualisys, *Qualisys*, `https://www.qualisys.com/`, Accessed: 2020-01-14.

[225] G. L. Kinzel and L. J. Gutkowski, « Joint Models, Degrees of Freedom, and Anatomical Motion Measurement », *Journal of Biomechanical Engineering*, vol. 105, no. 1, pp. 55–62, Feb. 1983, ISSN: 0148-0731. DOI: 10.1115/1.3138385. [Online]. Available: https://doi.org/10.1115/1.3138385.

[226] M. Schmidt, N. Roux, and F. Bach, « Minimizing Finite Sums with the Stochastic Average Gradient », *Mathematical Programming*, vol. 162, Sep. 2013. DOI: 10.1007/s10107-016-1030-6.

[227] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives*, 2014. arXiv: 1407.0202 [cs.LG].

[228] R. R. Sokal and C. D. Michener, *A Statistical Method for Evaluating Systematic Relationships*, ser. University of Kansas science bulletin. University of Kansas, 1958.

[229] B. Lenseigne and P. Dalle, « Using Signing Space as a Representation for Sign Language Processing », in *Gesture in Human-Computer Interaction and Simulation*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 25–36, ISBN: 978-3-540-32625-0.

[230] F. Lefebvre-Albaret and M. Gonzalez, *Sign3D - Specification de l'annotation*, Responsable : Websourd, 2013.

[231] M. Kipp, M. Neff, and I. Albrecht, « An annotation scheme for conversational gestures: how to economically capture timing and form », *Language Resources and Evaluation*, vol. 41, no. 3-4, pp. 325–339, 2007.

[232] S. R. Buss, « Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods », *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[233] A. Aristidou and J. Lasenby, « FABRIK: A fast, iterative solver for the Inverse Kinematics problem », *Graphical Models*, vol. 73, no. 5, pp. 243–260, 2011.

[234] D. Raunhardt and R. Boulic, « Progressive clamping », in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, pp. 4414–4419.

[235] C. W. Wampler, « Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[236] K. Duarte and S. Gibet, « Corpus Design for Signing Avatars », *LREC : Corpora and Sign Language Technologies*, 2010.

[237] M. Huenerfauth, « A Linguistically Motivated Model for Speed and Pausing in Animations of American Sign Language », *ACM Transactions on Accessible Computing (TACCESS)*, vol. 2, no. 2, 9:1–9:31, Jun. 2009, ISSN: 1936-7228. DOI: 10.1145/1530064.1530067.

[238] K. Duarte and S. Gibet, « Heterogeneous Data Sources for Signed Language Analysis and Synthesis:The SignCom Project », in *Language Resources and Evaluation Conference (LREC)*, 2010.

[239] T. Lebourque and S. Gibet, « A complete system for the specification and the generation of sign language gestures », in *International Gesture Workshop*, Springer, 1999, pp. 227–238.

[240] M. Aubry, P. De Loor, and S. Gibet, « Enhancing Robustness to Extrapolate Synergies Learned from Motion Capture », in *23rd International Conference on Computer Animation and Social Agents (CASA 2010)*, Saint Malo, France, May 2010, pp. 1–4. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00505188.

[241] F. c. Lefebvre-Albaret, « Traitement automatique de vidéos en LSF. Modélisation et exploitation des contraintes phonologiques du mouvement (French) [Automatic processing of LSF videos. Modelling and exploitation of the phonological constraints of motion] », PhD thesis, Université Paul Sabatier - Toulouse III, 2010.

[242] R. Plamondon and S. N. Srihari, « Online and off-line handwriting recognition: a comprehensive survey », *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[243] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. Daryin, M. Calvo, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais, « Fast multi-language LSTM-based online handwriting recognition », *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–14, 2020.

[244] R. Penrose, « A generalized inverse for matrices », in *Mathematical proceedings of the Cambridge philosophical society*, Cambridge University Press, vol. 51, 1955, pp. 406–413.

Annotation tracks of the *LSF-ANIMAL* database

Track Category	French Track Names	Track Content Description	Vocabulary	Annotation Type
Gloss	*Glose_2M*	Two-handed glosses	Open, see format on table 7.1	Manual + automatic refinement
	Glose_MD *Glose_MG*	Single handed glosses	Open, see format on table 7.1	
	Glose_complementaire	Complementary glosses	Open, see format on table 7.1	
Hand configuration	*Configuration_MD* *Configuration_MG*	Configurations of the hands	Closed, 48 labels: {*3, 3_plie, 4, 4_plie, 5, 5_plie, A, B, C, C_large, D, E, F, G, H, I, K, L, M, O, R, S, S_pouce_interieur, T, U, V, W, X, Y, angle_droit, angle_droit_index, ballon, bec_canard_ferme, bec_oie_rond, bec_oiseau_ferme, bec_oiseau_ouvert, clef, index, lit, main_plate, majeur, moufle, oui, personne, pince_fermee, pince_ouverte, u_pouce, wc*}	Automatic
Hand movement	*Mouvement_MD* *Mouvement_MG*	Nature of the movement performed by the hands	To be defined	To be performed
Hand orientation	*Orientation_MD* *Orientation_MG*	Orientation of the hands	To be defined	To be performed
Hand Placement	*Emplacement_Distance_MD* *Emplacement_Distance_MG*	Distance placement of the hands	Closed, 4 labels: {*Touch*, *Close*, *Normal*, *Far*}	Automatic
	Emplacement_Hauteur_MD *Emplacement_Hauteur_MG*	Height placement of the hands	Closed, 6 labels: {*BellowBelt*, *Abdomen*, *Chest*, *Neck*, *Head*, *AboveHead*}	Automatic
	Emplacement_Radial_MD *Emplacement_Radial_MG*	Radial placement of the hands	Closed, 5 labels: {*Front*, *Left*, *Right*, *BehindLeft*, *BehindRight*}	Automatic
Utterance	*Phrase*	Content of the utterances	open, use of lowercase	Manual
Symmetry	*Symetrie_Mains*	Presence and nature of symmetry axes	Closed, 3 labels: {*PV_Plan_Vertical*, *PH_Plan_Horizontal*, *Alternance*}	Manual

Table A.1 – The 18 annotation tracks.

Appendix B

Inverse Kinematics Based on the Jacobian

The Jacobian-based inverse kinematics problem can be formulated as:

$$d\theta = \alpha J^{-1}(X_T - X) \tag{B.1}$$

with $d\theta$ the small variations of the joints angles, J^{-1} the inverse of the Jacobian matrix, X_T the target position, X the current position of the end-effector and α the rate of convergence as the process is iterative and must be repeated until $(X_T - X) \approx 0$.

However, the Jacobian matrix is rarely invertible. We present here three alternatives to approximate the inverse of the jacobian.

Inversion of the Jacobian Using the Transpose

A simple solution to approach the Jacobian inversion is to transpose it. Indeed, all matrices are transposable, the operation is computationally cheap and produces a convincing approximation.

In this case, the equation B.1 becomes:

$$d\theta = \alpha J^{T}(X_T - X) \tag{B.2}$$

The inverse kinematics process is iterative: the gain α allows to control the angular increments between two iterations. It must be between 0 and 1.

The gain can also be computed at each iteration to be larger when the target is far away and smaller when it is close. Thus, Buss [232] proposes a variable α equal to :

$$\alpha = \frac{(X_T - X).(JJ^{T}(X_T - X))}{||(JJ^{T}(X_T - X)||^2}$$

In his work, Lebourque [167] proposes a gain of :

$$\alpha = \frac{||(X - X_T)||}{||(J^T(X - X_T)||}$$

Inversion of the Jacobian Using the Moore-Penrose Pseudo Inverse

An other solution is to use the pseudo-inverse of the Jacobian (noted J^+) as defined by Moore and Penrose [244]. In this case, the equation B.1 becomes:

$$d\theta = \alpha J^+(X_T - X) \tag{B.3}$$

When the number of rows of J is greater than the number of columns :

$$J(\theta)^+ = J^T(JJ^T)^{-1}$$

Otherwise:

$$J(\theta)^+ = (J^TJ)^{-1}J^T$$

The gain α can be set the same way as in the Jacobian transpose section.

Damped Least Square

To reduce the instabilities when the target is out of range, the *Damped Least Square* (DLS) technique is often used [235]. In this case, the equation B.1 becomes:

$$d\theta = \alpha(J^TJ + \lambda^2 I)^{-1}J^T(X_T - X) \tag{B.4}$$

with λ being the damping factor ($\lambda > 0$).

Comparison

Table B.1 compares the performances of the different IK techniques on one articulated chain composed of 10 joints and one end-effector.

IK	Parameters	Convergence Rate	Behavior around Singularities	Plausible Pose	Plausible Motion
JTranspose	$0 < \alpha < 0.3$	Low	Slightly Jerky	Yes	Yes
	$0.3 \leq \alpha \leq 1$	Very Low[a]	Jerky	No	No
Pseudo Inverse	$0 < \alpha < 0.3$	Very Low[a]	Correct	Yes	Yes
	$0.3 \leq \alpha < 0.5$	Medium	Slightly Jerky	Yes	Yes
	$0.5 \leq \alpha \leq 1$	High	Jerky	No	No
DLS	$0 < \alpha \leq 0.5$ [b]	Low	Correct	Yes	Yes
	$0.5 < \alpha \leq 1$ [b]	High	Correct	Yes	Yes
	$0 < \lambda \leq 0.3$ [c]	High	Jerky	No	No
	$0.3 > \lambda \geq 1$ [c]	High	Correct	Yes	Yes
	$\lambda > 1$ [c]	Very Low[a]	Correct	Yes	Yes
FABRIK	/	Very High	Correct	Yes	No[d]

[a] The target is sometimes not reached.
[b] With λ fixed at 1.
[c] With α fixed at 1.
[d] Not enough iterations.

Table B.1 – Comparison of the behavior of different IK algorithms (Jacobian-based and FABRIK) for one articulated chain composed of 10 joints and one end-effector.

Comparison of the Transition Durations

#	S1	S2	Duration (s)				
			Ground Truth	**Velocity-based**	**Bounded**	**Linear**	**Surface**
1	[PAYER]	[PRIX]	0.498	0.770514	0.5	0.430638	0.400005
2	[PRIX]	[12]	0.352	0.402331	0.402331	0.331447	0.329376
3	[12]	[EURO]	0.440	0.267967	0.267967	0.222374	0.256549
4	[EURO]	[50]	0.355	0.563915	0.5	0.184058	0.243529
5	[PRIX]	[17]	0.470	0.648562	0.5	0.444759	0.407496
6	[17]	[EURO]	0.305	0.127614	0.127614	0.165176	0.215599
7	[PRIX]	[31]	0.447	0.839399	0.5	0.455456	0.422985
8	[31]	[EURO]	0.392	0.319385	0.319385	0.206037	0.255116
9	[ENTRER]	[GRATUIT]	0.349	0.374111	0.374111	0.24527	0.290715
10	[POINTAGE]	[LA]	0.203	0.151246	0.151246	0.158968	0.196566
11	[LA]	[CARTE INVITATION]	0.364	0.291386	0.291386	0.148117	0.216041
12	[CARTE INVITATION]	[ALORS]	0.219	0.378107	0.378107	0.235402	0.274684
13	[ALORS]	[ENTRER]	0.319	0.132535	0.132535	0.188111	0.22384
14	[ENTRER]	[GRATUIT]	0.355	0.311788	0.311788	0.132572	0.237988
15	[FESTIVAL]	[CHEZ]	0.385	0.583948	0.5	0.431817	0.444757
16	[CHEZ]	[AMPHITHEATRE]	0.301	0.461027	0.461027	0.5	0.562276
17	[AMPHITHEATRE]	[THEATRE]	0.278	0.156934	0.156934	0.266869	0.291094
18	[TEMPS]	[MAUVAIS]	0.221	0.245358	0.245358	0.227326	0.290084
19	[MAUVAIS]	[ALORS]	0.262	0.209627	0.209627	0.284222	0.33191
20	[ALORS]	[FESTIVAL]	0.267	0.09095	0.1	0.173754	0.246231
21	[FESTIVAL]	[DEPLACE]	0.522	0.175287	0.175287	0.247192	0.302491
22	[CHEZ]	[BATIMENT]	0.241	0.718772	0.5	0.5	0.552332

Table C.1 – Durations of 22 transitions. The duration of each transition in the *MoCap* sequence ("ground truth") and the results of the four implemented computations of the duration are visible. Table C.2 shows the absolute difference between the ground truth duration and the different computations.

#	Difference with the ground truth (s)			
	Velocity-based	**Bounded**	**Linear**	**Surface**
1	0.27251	0.00200	0.06736	0.09800
2	0.05033	0.05033	0.02055	0.02262
3	0.17203	0.17203	0.21763	0.18345
4	0.20892	0.14500	0.17094	0.11147
5	0.17856	0.03000	0.02524	0.06250
6	0.17739	0.17739	0.13982	0.08940
7	0.39240	0.05300	0.00846	0.02401
8	0.07261	0.07261	0.18596	0.13688
9	0.02511	0.02511	0.10373	0.05829
10	0.05175	0.05175	0.04403	0.00643
11	0.07261	0.07261	0.21588	0.14796
12	0.15911	0.15911	0.01640	0.05568
13	0.18647	0.18647	0.13089	0.09516
14	0.04321	0.04321	0.22243	0.11701
15	0.19895	0.11500	0.04682	0.05976
16	0.16003	0.16003	0.19900	0.26128
17	0.12107	0.12107	0.01113	0.01309
18	0.02436	0.02436	0.00633	0.06908
19	0.05237	0.05237	0.02222	0.06991
20	0.17605	0.16700	0.09325	0.02077
21	0.34671	0.34671	0.27481	0.21951
22	0.47777	0.25900	0.25900	0.31133

Table C.2 – Absolute difference between each computation of the transition duration and the ground truth duration (in green: values that differ from the ground truth of less than 0.1s; in red: values that differ from the ground truth of more than 0.25s).

#	Difference with the ground truth (s)			
	Velocity-based	**Bounded**	**Linear**	**Surface**
Average (s)	0.16456	0.11301	0.11281	0.10153
Std (s)	0.12151	0.08597	0.09163	0.08136
Number of values > 0.25s	4	2	2	2
Number of values <0.1s	8	11	11	14

Table C.3 – Performances of the four implemented computations of the transition duration (worst results in red. best results in green).

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud ubs

Titre : Capture, annotation et synthèse de mouvements pour l'animation basée données d'avatars de langue des signes française

Mot clefs : Capture de Mouvement, Avatar signeur, Langue des Signes Française, Annotation Automatique, Synthèse de Mouvements

Résumé : Cette thèse porte sur la capture, l'annotation, la synthèse et l'évaluation des mouvements des mains et des bras pour l'animation d'avatars communiquant en Langues des Signes (LS). Actuellement, la production et la diffusion de messages en LS dépendent souvent d'enregistrements vidéo qui manquent d'informations de profondeur et dont l'édition et l'analyse sont difficiles. Les avatars signeurs constituent une alternative prometteuse à la vidéo. Ils sont généralement animés soit à l'aide de techniques procédurales, soit par des techniques basées données. L'animation procédurale donne souvent lieu à des mouvements peu naturels, mais n'importe quel signe peut être produit avec précision. Avec l'animation basée données, les mouvements de l'avatar sont réalistes mais la variété des signes pouvant être synthétisés est limitée et/ou biaisée par la base de données initiale. Privilégiant l'acceptation de l'avatar, nous avons choisi l'approche basée sur les données mais, pour remédier à sa principale limitation, nous proposons d'utiliser les mouvements annotés présents dans une base de mouvements de LS capturés pour synthétiser de nouveaux signes et énoncés absents de cette base.

Pour atteindre cet objectif, notre première contribution est la conception, l'enregistrement et l'évaluation perceptuelle d'une base de données de capture de mouvements en Langue des Signes Française (LSF) composée de signes et d'énoncés réalisés par des enseignants sourds de LSF. Notre deuxième contribution est le développement de techniques d'annotation automatique pour différentes pistes d'annotation basées sur l'analyse des propriétés cinématiques de certaines articulations et des algorithmes d'apprentissage automatique existants. Notre dernière contribution est la mise en œuvre de différentes techniques de synthèse de mouvements basées sur la récupération de mouvements par composant phonologique et sur la reconstruction modulaire de nouveaux contenus de LSF avec l'utilisation de techniques de génération de mouvements, comme la cinématique inverse, paramétrées pour se conformer aux propriétés des mouvements réels.

Title: Capture, Annotation and Synthesis of Motions for the Data-Driven Animation of Sign Language Avatars

Keywords: Motion Capture, Signing Avatar, French Sign Language, Automatic Annotation, Motion Synthesis

Abstract: This thesis deals with the capture, annotation, synthesis and evaluation of arm and hand motions for the animation of avatars communicating in Sign Languages (SL). Currently, the production and dissemination of SL messages often depend on video recordings which lack depth information and for which editing and analysis are complex issues. Signing avatars constitute a powerful alternative to video. They are generally animated using either procedural or data-driven techniques. Procedural animation often results in robotic and unrealistic motions, but any sign can be precisely produced. With data-driven animation, the avatar's motions are realistic but the variety of the signs that can be synthesized is limited and/or biased by the initial database. As we considered the acceptance of the avatar to be a prime issue, we selected the data-driven approach but, to address its main limitation, we propose to use annotated motions present in an SL *Motion Capture* database to synthesize novel SL signs and utterances absent from this initial database.

To achieve this goal, our first contribution is the design, recording and perceptual evaluation of a French Sign Language (LSF) *Motion Capture* database composed of signs and utterances performed by deaf LSF teachers. Our second contribution is the development of automatic annotation techniques for different tracks based on the analysis of the kinematic properties of specific joints and existing machine learning algorithms. Our last contribution is the implementation of different motion synthesis techniques based on motion retrieval per phonological component and on the modular reconstruction of new SL content with the additional use of motion generation techniques such as inverse kinematics, parameterized to comply to the properties of real motions.