Example-based Multilinear Sign Language Generation from a Hierarchical Representation

Boris Dauriac, Annelies Braffort, Élise Bertin-Lemée

Proceedings of the 7th International Workshop on Sign Language Translation and Avatar Technology (SLTAT 7), pages 21–28
Language Resources and Evaluation Conference (LREC 2022), Marseille, 20-25 June 2022

Example-based Multilinear Sign Language Generation from a Hierarchical Representation

Boris Dauriac[1], Annelies Braffort[2], Elise Bertin-Lemée[3]
[1]R&D MocapLab, 70 rue du Landy, 93300 Aubervilliers, France, boris.dauriac@mocaplab.com
[2]Université Paris-Saclay, CNRS, LISN, Orsay, France, annelies.braffort@lisn.upsaclay.fr
[3]SYSTRAN, 5 rue Feydeau, Paris, France elise.bertinlemee@systrangroup.com

Abstract

This article presents an original method for automatic generation of sign language (SL) content by means of the animation of an avatar, with the aim of creating animations that respect as much as possible linguistic constraints while keeping bio-realistic properties. This method is based on the use of a domain-specific bilingual corpus richly annotated with timed alignments between SL motion capture data, text and hierarchical expressions from the framework called AZee at subsentential level. Animations representing new SL content are built from blocks of animations present in the corpus and adapted to the context if necessary. A smart blending approach has been designed that allows the concatenation, replacement and adaptation of original animation blocks. This approach has been tested on a tailored testset to show as a proof of concept its potential in comprehensibility and fluidity of the animation, as well as its current limits.

Keywords: Sign Language, avatar animation, motion capture, representation of Sign Language, AZee

1. Introduction

Rosetta[1] is a French project that aimed to study accessibility solutions for audiovisual content. One of the experiments consisted in designing an automatic translation system from text to Sign Language (SL) displayed through animation of a virtual signer.
The three main contributions concerning SL in this project were the constitution of Rosetta-LSF (Dauriac, 2022), an aligned corpus of text and SL captured using a mocap system, a translation system from text to AZee (Bertin-Lemée et al., 2022b), a representation of SL content, and a system allowing to generate virtual signer animations from AZee input.
This article describes the third contribution: the system of generation from AZee to virtual signer animations. After an overview of recent works in the field, we give some indications on the Rosetta-LSF corpus and the way it has been annotated in order to facilitate its use for generation, then we describe the main steps of the generation system. Finally, we give preliminary results and discuss the questions raised for evaluation.

2. Sign Language Generation

Sign language generation consists of creating animations that represent contents in SL, applied to a virtual character. These creations must be guided by a linguistic model of SL. The first section lists the concepts used in this article and the second one provides an overview of representative recent work in the field.

2.1. Avatar Animation

An avatar is made up of a complex 3D mesh that is given a humanoid shape, forming a virtual *skin*. It can be animated thanks to a virtual *skeleton* which is a tree structure composed of rigid segments called *bones* connected by joints. Each joint represent the six degrees of freedom (three rotations and tree translations) of a bone with respect to its parent, also called *3D pose*. A rig makes the link between the skeleton and the skin by defining the deformation of the latter depending on the bones' 3D pose.
An animation is a sequence of avatar poses displayed at a given frequency. Some poses, defined at a given timecode, are called *keyframes*. They act as control points in space and time, and may not be defined at each frame. From the main approaches listed by Naert et al. (2020), one can summarize three main approaches used for animation creation:

- *Hand-crafted*: The specification of keyframes is done manually, possibly assisted by computer and with techniques such as rotoscoping. The transitions between keyframes can be automatically computed using interpolation, resulting in a continuous movement. The quality of such animations relies on the skill level of the animator who select the keyframes. If they are not well chosen, this can result in movements that are robotic and perceived as not bio-realistic.

- *Automatic keyframing*: The principles are almost the same, except that the sequence of keyframes is provided by a representation of the sign structure rather than created by hand. Here also, the animation can be perceived as not good enough, because the computation relies on models that do not always take into account all the properties that allow the synthesis of a bio-realistic movement.

- *Data-driven*: The motion is captured on a human

[1]https://rosettaccess.fr/index.php/home-page-english/

	Generation of basic animation			Generation of final animation	
Name of the project	Hand crafted	Automatic keyframing	Mocap	Simple concat.	Edited concat.
JASigning		x		x	
EMBR		x		x	
Naert's project			x		x
Paula	x	x			x
Rosetta			x		x

Table 1: List of the most recent signing avatar systems.

using a motion capture (mocap) device. This allows for a high level of bio-realism but requires the use of a mocap system, and therefore a post-processing step on the recorded data.

To generate the final content, there are two main approaches:

- *Simple concatenation*: Blocks of animations are concatenated to form the final animation. These animations may have been created using any of the techniques outlined above. A process, called animation blending and described below, must then be implemented to link the blocks so that there is no break between the concatenated animations.
- *Edited concatenation*: There is still concatenation, but, in addition, edition of the blocks of animations is possible, in order to adapt the block to the context or to add realism to the whole animation.

One simple way to use existing data and combine them into new data is to use animation blending. This technique is implemented in different animation softwares like Blender or Motionbuilder. Video games industry relies a lot on blending, for example to generate a transition from running to walking. This is the same idea as a video or sound editing software. One can have several clips of animation on several tracks. Each track controls the 3D pose of a defined set of bones. On a given track, depending on the way clips overlap or not, two methods can be used:

- *Temporal interpolation*: When there is no overlapping between clips, temporal interpolation can be controlled between them.
- *Blend*: When there is overlapping of two clips, a blend is applied to transition from one clip to another. This blend is basically a weighted average of the set of 3D poses in the two clips. A "function of activation" is used to control the fading in and fading our of each clip across time.

2.2. Virtual Signer Animation

The generation of animations for virtual signers is a relatively new and underdeveloped field. Despite this, the different approaches listed above have been tested in research projects or even commercial products on SL. We propose here a synthesis of the most recent ones by positioning them according to these categories, grouped in Table 1.

A first generation of projects have been based on "Automatic keyframing / Simple concatenation" approaches: The first step consists of creating a collection of animations representing isolated lexical units (signs) which are stored in a database and identified by a gloss[2]. These sign animations are automatically keyframed, using a sign-level representation that describes the key poses. The signs are generally described in their citation form, i.e. not inflected by the linguistic context. Some procedural processes sometimes allow to inflect the signs so as to match with their surrounding linguistic context, or to add behaviour activity (e.g. breathing), but generally in a very limited manner. As a second step, SL utterances are built as a sequence of animation blocks. As such they are generally based on a simple concatenative approach. They are extracted from a database and concatenated to form a SL utterance. To date, the two platforms of this kind that have been most used are:

- JASigning: The Java Avatar Signing system is a platform tool for the synthesis of any sign language, freely available for research purposes (Elliott et al., 2008). It has been used for several projects with various SLs (Ebling and Glauert, 2013; Ebling and Glauert, 2016; Efthimiou et al., 2019; Roelofsen et al., 2021). The signs are represented in their citation form using SiGML, which is built on HamNoSys, a transcription system for signs. Sign inflection is possible in a limited manner and only at the sign level.
- EMBR (Embodied Agents Behavior Realizer) (Heloir and Kipp, 2009; Huenerfauth and Kacorri, 2015): The signs are represented in their citation form using k-pose-sequences called EMBRScript, coming with explicit timing information. Sign inflection is not possible.

These approaches have the same drawbacks: they use signs in their citation form with little or no sign inflection capabilities, they do not integrate linguistic structures such as classifiers, and they do not have a very advanced management of temporal aspects, either at the

[2]A gloss is a text label, generally a single word, reflecting the meaning of the sign it stands for.

level of signs or utterances. Moreover, as the animation is built from pure procedural synthesis, the rendering is rather robotic and far from being bio-realistic.
More recent projects aim to overcome these limitations, using edition approaches each with its own specificity:

- Naert's project: This project is based on the use of a mocap database in which movements have been annotated using a linguistic model. Several techniques are used to build new signs and to modify signs regarding the context. These processes are currently limited to phenomena involving the hand location and handshape (Naert, 2020).

- Paula: The DePaul University signing avatar project has been designed first for American Sign Language but is now being used for various SLs (McDonald et al., 2016). Initially designed to support professional animator's work by including a number of automation of current processes for the generation of content in SL, it is based on handcrafted animations. It relies on a multitrack animation engine, allowing for flexible and accurate synchronisation between the various parts of the body to be animated. Several procedural tools allow to increase naturalness, to modify or adapt signs to the context, or to create new ones, including classifiers, thanks to a formal linguistic representation of SL called AZee (McDonald and Filhol, 2021).

The approach we present here, used in the Rosetta project, is based on the use of gold standard motion capture for the constitution of a database of LSF extracts, AZee as the representation that drives the generation of the final animation, and on an edition approach, combining concatenation and procedural techniques.

First of all, we briefly present the corpus produced within the framework of this project.

3. Motion Capture Corpus

In our project we used the first task of the Rosetta-LSF corpus (Dauriac, 2022), downloadable from Ortolang[3]. This consists in richly annotated LSF translations of 194 news in French which are between three and 35 words in length, for instance: "*L'Everest menacé de réchauffement climatique*" (Everest threatened by global warming). More details on this corpus can be found in Bertin-Lemée et al. (2022a). After the motion capture, a 3D avatar with the same body proportions as the signer was created from the marker set. The avatar animations were then implemented into a 3D player to produce a video for each acquisition (see fig. 1), allowing to use an annotation software to annotate the SL content.
While AZee describes the structure and content of the SL utterance, the annotation scheme was designed to provide descriptions at the sign level. The annotations specify articulatory constraints and temporal information relevant for the generation process. Two tracks were used to annotate manual activity of right and left arms and hands. Annotation was carried out in a classical way, by segmenting and annotating manual units, but was not limited to assigning a simple gloss to them (*IdGloss* attribute). We added the different constraints to be applied on these units for any context of use, so as to inform the generation process about the possibilities or needs for modification in a new linguistic context.
For each segment, on each track, four attributes have been specifically defined to help the generation pro-

[3] https://www.ortolang.fr/market/corpora/rosetta-lsf

Figure 1: Avatar rendering

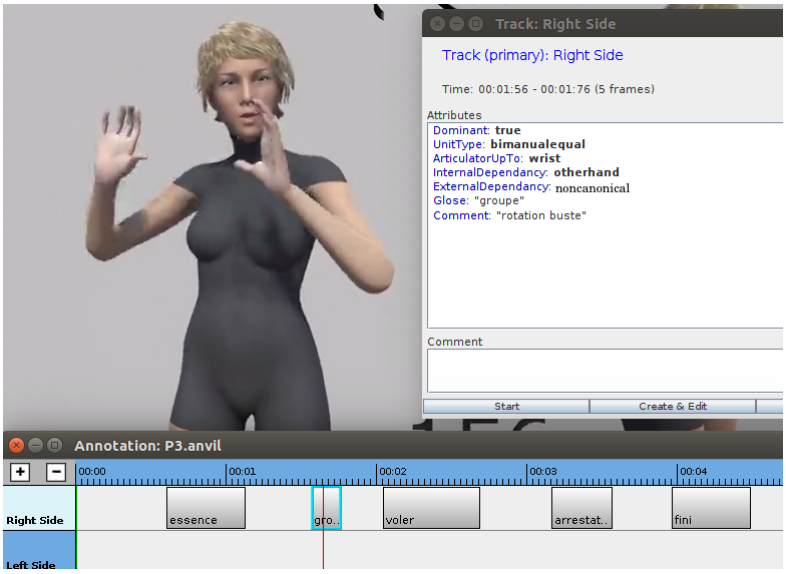

Figure 2: Example of annotation using ANVIL (Kipp, 2014)

cess:

- *UnitType*: This attribute allows to distinguish three categories of unit according to the number of hands involved and the nature of the relationship between the hands. It may have four values: monomanual (unit performed with one hand, this is the default value), bimanualequal (unit performed with both hands where there is no dominance of one hand over the other), bimanualdominant (unit performed with both hands for which there is a relationship of dominance of one hand over the other), and unknown (in case of doubt).

- *ArticulatoryUpTo*: This attribute identifies the articulatory constraints of the unit for the considered side. The aim is to indicate to the generation process the necessary and sufficient constraints, thus leaving the process free to modify the bending of certain joints if needed. This concerns the local constraints of all articulatory segments from the fingers to the shoulder. It is therefore more precise than what is usually called "handshape". This attribute may have six values: no (unconstrained posture), fingers (all the fingers are constrained and not the whole hand), wrist (the whole hand is constrained and not the forearm), elbow (the hand and forearm are constrained and not the arm), other (other cases to be detailed in the comments), and unknown (in case of doubt). No more indication is given (handshape, orientation and location), as this is directly retrievable from the mocap data.

- *InternalDependency*: This attribute describes the constraints between the hand and other parts of the body. The objective is to indicate the necessary and sufficient constraints to satisfy when modifications are applied to certain articulators (e.g. moving a hand, rotating the head, etc.). It may have six values: no (no constraints, default value), otherhand (constraint with respect to the other hand), head (constraint with respect to the head), body (constraint with respect to the torso), other (constraint with respect to another part of the body, to be specified in the comments), and unknown (in case of doubt).

- *ExternalDependency*: This attribute indicates the possibility or existence of constraints of the hand with respect to the signing space. The aim is to indicate if the articulation depends on a spatial context (e.g. modification of hand orientation or location, movement amplitude), so that the generation can be adapted to the spatial context. The possible values are notapplicable (for a sign that cannot be modified), canonical (when it is not modified), non canonical (when modified), and unknown (in case of doubt).

The fig. 2 shows an example of annotation for the sign *"GROUPE"* (GROUP) on the Right track: The *Dominant* attribute value is true (the signer is right-handed), the sign type is *bimanualequal* (no dominance of one hand over the other), and articulatory constraints up to the *wrist* (no constraints on other segments on the right side), with an *otherhand* internal dependency of one hand to the other, and an *noncanonical* external dependency as it is relocated.

To date, all 194 titles in task 1 have been annotated. This corpus was used to generate new utterances. The principles used to create these new animations are described below.

4. Generation Methodology

As for the Paula project, the description of the utterance to be generated is given by an AZee description. AZee is a formal approach to SL discourse representation (Hadjadj et al., 2018; Challant and Filhol, 2022). It allows to define *production rules* that associate forms to be articulated (to generate an animation in SL) and identified meaning. By combining them, one builds tree-structured expressions that generate signed utterances. Each node of the expression hierarchy therefore represents a portion of the utterance by itself, with the root node by definition covering the entire discourse. A "%t" pragma is appended on the AZee source line of nodes, followed by the corresponding text and the video frame numbers identifying the beginning and the end of aligned segment (see fig. 1, top right, second line: 7713), as illustrated in fig. 3. In this example, three nodes are defined: the first one is "*ont vendu leur vaisselle*" (sold their tableware) from frames 1739 to 1967, and it includes 2 sub-nodes: "*vaisselle*" (tableware) from frames 1767 to 1851, and "*vendu*" (sold) from frames 1855 to 1967. Each node with a "%t" is thus associated with a segment of mocap file forming an animation block, which we will call *AZee block* in the following. The smallest AZee block that can be found in the corpus is at the level of the sign.

```
:info-about %F ont vendu leur vaisselle %t 1739-1967
   'topic
   :là
   'info
   :info-about
      'topic
      :all-of %F vaisselle %t 1767-1851
         'items
         list
            :assiette
            :assiette
      'info
      :multiplicity %F vendu %t 1855-1967
         'elt
         :vendre
```

Figure 3: Excerpt from an AZee discourse expression.

Using our corpus, composed of the mocap files, associated annotations and AZee descriptions, we are able to generate a new sentence, by collecting blocks of mocap data, concatenating, and modifying them when needed, with the approach summarized in fig. 4.
The general principle of the smart blending methodology we designed in the Rosetta project is based on the fact that motion is managed synchronously over several animation tracks. Each track corresponds to a set of anatomical parts representing effectors such as the right arm, left arm, trunk, head, facial expressions, eye gaze, and the rest of the body. Thus, using a non linear animation blending tool on this hierarchical skeleton, it becomes then possible to assemble several blocs to generate new sign language sentences while keeping a multi-track approach. This is the main particularity of our proposed approach.

A new sentence to be generated is described within an AZee file: each necessary AZee block is extracted from the database, then, there are two blending cases:

- either a *%fallback* AZee block is mentioned, meaning that no higher-level block have been found to make the link between one sub-block and another ("Fallback" box fig. 4).
- Or a sub-block is replaced inside an AZee block ("Replacement" box fig. 4).

For each case, we have designed one blending methodology which corresponds to the two necessary operations for the creation of the new utterance.
In the first case, the *fallback blending*, one wants to transition from the end of one block to the beginning of the second one. As no information is known on how to put these two pieces together in a seamless sequence, this transition can occur simultaneously on all tracks (including facial expression, eye gaze, etc.) but require some precaution on the duration of such transition. In the corpus, the main end-effectors with the highest dynamics were found to be the wrists and the head. To compute the time allowed for transition between two blocks, i.e. the blending time, 3D position in the global 3D frame of these end-effectors have been used. To ensure the bio-realistic dynamics of the transition and predict the necessary time window, a simple proportional calculation have been used on the distance covered by each end-effector (wrist and head) in high dynamic movements between annotated AZee blocks from the corpora. The maximum of the predicted time windows for the three end-effectors has been used.
In the second case, the *replacement blending*, one wants to change an AZee sub-block inside an AZee block. In the simplest case, one wants to change one sign in a block, a city name for example. In many cases, animation from the arms have to be replaced while the rest of the body must be maintained to preserve the AZee block structure. This replacement may raise several problems:

- The block to be replaced and the one to be inserted don't have the same duration.
- The position of each segment in the global 3D frame are not the same between the replaced and inserted blocks, requiring a blending at the beginning and end.
- The inserted block may not have the same number of tracks as the replaced one.

For the duration offset between replaced and inserted blocks, it has been chosen to keep the inserted block duration. This means that each track where nothing is replaced (head and eye gaze tracks for example) has to be stretched or squeezed to match the inserted block duration. This choice has been made as the majority

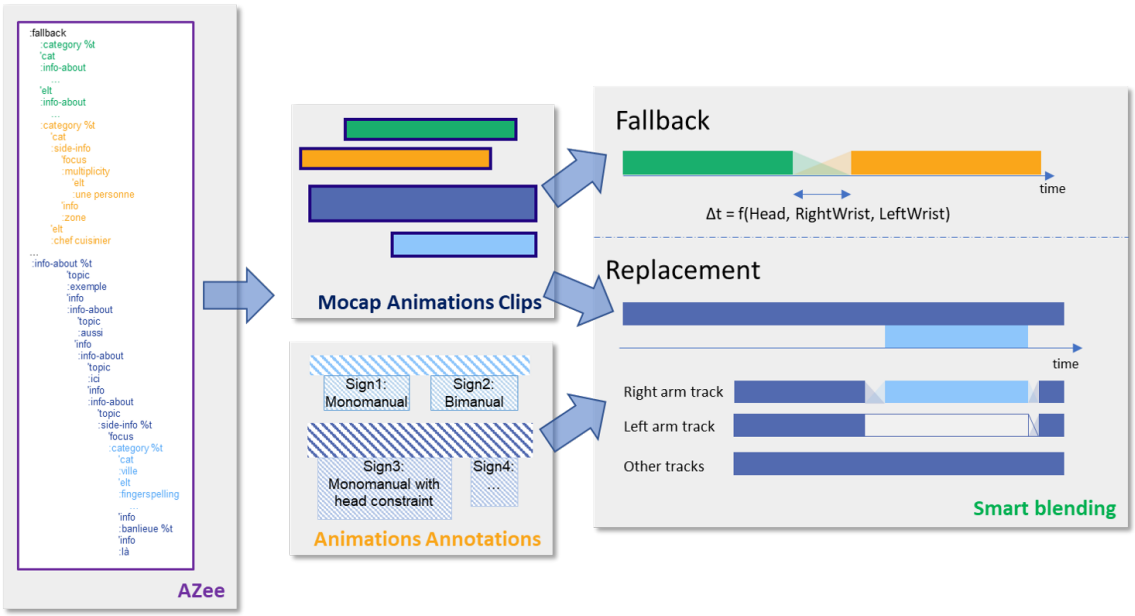

Figure 4: Smart Blending generation approach in Rosetta

of the inserted blocks were longer in duration than the replaced ones.

Blending time between the previous AZee sub-block and the next AZee sub-block is maintained on the replaced tracks.

For track replacements, for example if one takes the smallest AZee sub-block, i.e. a sign, it can be monomanual or bi-manual. When replacing a bimanual sign with a monomanual one, the non dominant arm track needs to be "emptied" and the monomanual animation of the non dominant arm is not used as it is not meaningful. Between the end of the previous AZee sub-block and the replaced sub-block end, the animation on the non dominant arm track is deleted and a blend is performed between the end of the sub-block and the beginning of the next AZee sub-block. When replacing a monomanual sign with a bimanual one, the non dominant arm track is replaced like the dominant one with the inserted sub-block. The same principle was applied to bigger AZee sub-blocks and other track conflicts by searching through the corpora annotations on the articulatory constraints (UnitType and InternalDependency Attributes).

At the end of the procedure, a video of the newly generated sentence has been created with a rendering engine (Unity[4]).

The approach aims at minimizing the edition of recorded movements to leverage the fine-grained precision of motion capture. For Fallback, motion edition only occurs during blending on all tracks. For Replacement, the methodology focused on the two arm tracks and their dependencies with other articulatory tracks. No edition was made to modify originally directional signs, nor on facial expressions as they were not annotated.

5. Tests and Discussion

In order to test the whole translation system from text to SL via the animation of a virtual signer, a testset was built by creating new sentences mixing segments from different newstitles of our corpus. 15 sentences were created and we retained the AZee translation of seven of them to test the functionality of our generation system. For example, we got the AZee description of the following sentence: "*Alsace : de grands chefs ont vendu leur vaisselle pour les plus modestes dans la banlieue de Gerstheim.*" (Alsace: top chefs sold their tableware for households in the lowest income group in the suburbs of Gerstheim.).

The corresponding animation was generated using mocap blocks extracted from the LSF translations of the following three sentences present in the corpus:

- "*Samedi 30 et dimanche 31 mars, de grands chefs ont vendu leur vaisselle en Alsace, à Gerstheim.* (On Saturday 30 and Sunday 31 March, top chefs sold their tableware in Alsace, in Gerstheim.)
- "*Moins de TVA pour les plus modestes: " Il ne faut pas traiter ça par le mépris ",lance Xavier Bertrand au gouvernement*" (Less VAT for households in the lowest income group: "We must not treat this with contempt", says Xavier Bertrand Bertrand to the government.)
- "*Le superéthanol n'est proposé que dans 1 000 stations-service en France, comme ici dans la banlieue de Bordeaux*" (Superethanol is only

[4] https://unity.com/

available at 1,000 service stations in France, like here in the suburbs of Bordeaux.)

From sentence animations, six AZee blocks have been extracted corresponding to "*Alsace*", "*Gerstheim*", "*grands chefs*", "*ont vendu leur vaisselle*", "*pour les plus modestes*" and "*comme ici dans la banlieue de Bordeaux*". Fallbacks where used to associate all the AZee blocks, apart from "*comme ici dans la banlieue de Bordeaux*" where "*Bordeaux*" needed to be replaced with "*Gerstheim*" inside the block. The annotation indicated that "*Gerstheim*" AZee block has constraints: both arms are used (UnitType attribute). The fallback methodology allowed to compute a blending time between each AZee block. They lied between 0.19 and 0.49 seconds. The duration of "*Bordeaux*" AZee block was 0.24 seconds whereas the "*Gerstheim*" one took 3.48 seconds (because this proper name is fingerspelled). "*Bordeaux*" AZee block has been slowed down to match "*Gerstheim*" AZee block duration. Then, on the right and left arm tracks, the animation of "*Bordeaux*" AZee block has been replaced with "*Gerstheim*" one. In "*comme ici dans la banlieue de Bordeaux*", the AZee block before "*Bordeaux*" was "*là*" and the one afterwards was "*banlieue*". A blend from the end of "*là*" AZee block and the beginning of "*Gerstheim*" AZee block as well as between "*Gerstheim*" AZee block end and "*banlieue*" AZee block beginning was applied according to the given annotation duration.

A video showing the result of the whole system (translation and generation) for the seven sentences can be seen on the project website[5]. The second sentence of the video is the one described here above.

Although a real evaluation could not be carried out on such a limited number of examples, we were able to show them to the advisory board of the project which gave us some qualitative feedback. There were few comments on the multi-track methodology itself, with remarks focusing more on possible translation problems, contextualisation problems with the image added to the left of the avatar, or signing speed problems, as the person we recorded signs quickly. A few negative points were noted related to the appearance of the avatar (we only had a very simplified avatar in this project), the presence of a very local sign and therefore not necessarily known by everyone (the sign representing the Parisian urban transport company: RATP), and an error in the choice of a variant for the sign "là", probably due to a lack of precision during the annotation process. The positive points that were identified concern the fluidity of the animation. A comparison between an animation generated with a classical concatenation method and the method presented here was shown to the advisory board members, who preferred the rendering of the second one. There was no difference in the perception of smoothness between the animations with our method and the animations generated by simply replaying the mocap.

Of course, there are still a number of aspects to be addressed. For example, we have not yet annotated the non-manual elements (mouthing, facial expressions, eye gaze) in the corpus. Once done, there will be no particular difficulty in taking them into account during the animation process because the methodology already allows for this. Another important aspect concerns the management of the signing space. The annotation already provides an indication of whether a sign is in its canonical form or not regarding the spatial context (ExternalDependency attribute). Several strategies can be explored. For example, for a given sign performed in a canonical way, one could generate a non canonical relocated form by combining the handshape(s) with the location of another sign, while respecting the possible internal dependencies.

6. Conclusion and Prospects

We have presented here a new system of automatic generation from AZee (a hierarchical representation of SL) to French Sign Language (LSF), by means of the animation of an avatar, based on smart blending approaches and the use of an aligned corpus of AZee descriptions and mocap data, the Rosetta-LSF corpus. An implementation of the system has been made and has allowed to test its functioning on some examples, thus providing a proof of concept.

The capacities of this system and the size of the corpus still need to be extended before real evaluations can be carried out. But we can already stress that the evaluation of such a system will not be easy.

Metrics for evaluating the quality of translations, such as the ones proposed in the European QT21 project[6], provide a scoring grid for the types of errors produced by the translation system, which makes it possible to highlight the shortcomings of the systems and subsequently prioritise the areas for improvement. This project has proposed Multidimensional Quality Metrics (MQM), which is a framework for describing and defining custom translation quality metrics. It provides a flexible vocabulary of quality issue types, a mechanism for applying them to generate quality scores, and mappings to other metrics.

Some of the error categories, linked to the translation process itself, are called "Accuracy". There is an accuracy error when the target does not accurately reflect the source message. Our generation system does not handle the translation process, which role is to translate between French text into AZee description, and so we cannot use this type of error to analyse the quality of the animation. Another category called "Fluency" allows us to evaluate the quality of an utterance, whether it is the result of a translation or not. These errors can

[5] `https://rosettaccess.fr/index.php/rosettas-final-demonstrator/` - Note that the subtitles were also automatically produced, and therefore may contain errors compared to the spoken French version.

[6] `https://www.qt21.eu/`

be related to grammar, spelling, typography, inconsistency, opacity. In our case, the target is not a text, but an avatar animation, thus some of these categories cannot be used at all, and other should be adapted. For example, it is not necessarily easy to define the types of grammatical errors for SL. Anyway, it would be interesting to study if this kind of evaluation could be adapted to our system. To these categories, we will certainly have to add a category related to "Body Fluency", allowing to evaluate all the aspects linked to the naturalness of the movement and its bio-realistic aspect, making a distinction between linguistic fluency and body fluency.

The establishment of a robust and comprehensive evaluation protocol is clearly a subject of study in its own that needs to be pursued in the near future.

7. Acknowledgements

This work has been funded by the Bpifrance investment project "Grands défis du numérique", as part of the ROSETTA project (RObot for Subtitling and intElligent adapTed TranslAtion).

We thank Noémie Churlet, Raphaël Bouton and Media'Pi! for their commitment to this project, which would not have had the same validity and impact without them.

8. Bibliographical References

Bertin-Lemée, E., Braffort, A., Challant, C., Danet, C., Dauriac, B., Filhol, M., Martinod, E., and Segouat, J. (2022a). Rosetta-LSF: an Aligned Corpus of French Sign Language and French for Text-to-Sign Translation. In *13th International Conference on Language Resources and Evaluation (LREC)*.

Bertin-Lemée, E., Braffort, A., Challant, C., Danet, C., and Filhol, M. (2022b). Example-Based Machine Translation from Text to a Hierarchical Representation of Sign Language. *arXiv preprint arXiv:2205.03314*.

Challant, C. and Filhol, M. (2022). A First Corpus of AZee Discourse Expressions. In *Language Resources and Evaluation Conference (LREC), Representation and Processing of Sign Languages, Marseille, France*.

Ebling, S. and Glauert, J. (2013). Exploiting the full potential of JASigning to build an avatar signing train announcements. In *Third International Symposium on Sign Language Translation and Avatar Technology*.

Ebling, S. and Glauert, J. (2016). Building a Swiss German Sign Language avatar with JASigning and evaluating it among the Deaf community. *Universal Access in the Information Society*, 15:577—587.

Efthimiou, E., Fotinea, S.-E., Goulas, T., Vacalopoulou, A., Vasilaki, K., and Dimou, A.-L. (2019). Sign Language Technologies and the Critical Role of SL Resources in View of Future Internet Accessibility Services. *Technologies*, 7(1).

Elliott, R., Glauert, J., Kennaway, R., Marshall, I., and Safar, E. (2008). Linguistic modelling and language-processing technologies for avatar-based sign language presentation. *Universal Access in the Information Society*, 6:375—391.

Hadjadj, M., Filhol, M., and Braffort, A. (2018). Modeling French Sign Language: a Proposal for a Semantically Compositional System. In *International Conference on Language Resources and Evaluation*.

Heloir, A. and Kipp, M. (2009). EMBR: A realtime animation engine for interactive embodied agents. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*.

Huenerfauth, M. and Kacorri, H. (2015). Augmenting EMBR Virtual Human Animation System with MPEG-4 Controls for Producing ASL Facial Expressions. In *5th International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*.

Kipp, M. (2014). ANVIL: A Universal Video Research Tool. pages 420–436.

McDonald, J. and Filhol, M. (2021). Natural synthesis of productive forms from structured descriptions of sign language. *Machine Translation*, 35(3):363–386.

McDonald, J., Wolfe, R., Schnepp, J., Hochgesang, J., Gorman Jamrozik, D., Stumbo, M., Berke, L., Bialek, M., and Thomas, F. (2016). An automated technique for real-time production of lifelike animations of American Sign Language. *Universal Access in the Information Society*, 15:551—-566.

Naert, L., Larboulette, C., and Gibet, S. (2020). A survey on the animation of signing avatars: From sign representation to utterance synthesis. *Computers Graphics*, 92:76–98.

Naert, L. (2020). *Capture, annotation and synthesis of motions for the data-driven animation of sign language avatars*. Phd thesis in computer science, Université de Bretagne Sud.

Roelofsen, F., Esselink1, L., Mende-Gillings, S., and Smeijers, A. (2021). Sign Language Translation in a Healthcare Setting. In *Translation and Interpreting Technology Online (TRITON)*, pages 110—-124.

9. Language Resource References

Dauriac, B. et al. (2022). *ROSETTA-LSF corpus*. distributed via ORTOLANG: `https://hdl.handle.net/11403/rosetta-lsf/v1`, v1.