Multi-Track Bottom-Up Synthesis from Non-Flattened AZee Scores

Paritosh Sharma, Michael Filhol

Multi-Track Bottom-Up Synthesis from Non-Flattened AZee Scores

Paritosh Sharma, Michael Filhol
Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), CNRS, Université Paris–Saclay, Orsay, France
paritosh.sharma@lisn.upsaclay.fr, michael.filhol@cnrs.fr

Abstract

We present an algorithm to improve the pre-existing bottom-up animation system for AZee descriptions to synthesize sign language utterances. Our algorithm allows us to synthesize AZee descriptions by preserving the dynamics of underlying blocks. This bottom-up approach aims to deliver procedurally generated animations capable of generating any sign language utterance if an equivalent AZee description exists. The proposed algorithm is built upon the modules of an open-source animation toolkit and takes advantage of the integrated inverse kinematics solver and a non-linear editor.

Keywords: AZee, sign language, avatar

1. Introduction

Sign language synthesis is a technique for converting a sign language utterance description into an avatar animation. Such avatars are commonly referred to as signing avatars. Automating this process can provide a flexible way to generate sign language content while preserving the signer's anonymity. This also provides means to customize the sign language content more conveniently than fixed video recordings of signers.

Various systems for sign language synthesis have been developed over the years. Most of them relied on descriptions that modeled sign language utterances as sequences of glosses. This approach has several limitations ranging from synchronisation to contextual variations of signs. Hence, various utterance representations have been developed over these years to address one or more of these problems. EMBRScript (Kipp et al., 2011) added timing information to these sequences of glosses. The P/C model (Huenerfauth, 2006) solves the problem of synchronisation and concurrency of signs by allowing for partitions in utterance descriptions. The ATLAS project (Lombardo et al., 2010; Bertoldi et al., 2010) addresses the issue of sign variations using modifiers. Finally, the HLSML model (López-Colino and Pasamontes, 2011; López-Colino and Pasamontes, 2012) addresses the issue of timing information and sign variations.

Unlike those mentioned above, the AZee model (Filhol et al., 2014) allows us to write parameterised signed forms for semantic functions. A sign language utterance is encoded in the form of a hierarchy of applied production rules instead of a sequence. Given a description, it produces a timeline specifying all parts of the utterance to render with the avatar, thereby addressing the issues of non-manual features synchronisation, sign concurrency, and timing. Furthermore, AZee's timeline specifications also carry interpolation information and are essential for synthesising the utterance.

These features of AZee are essential for our work since modern animation systems use a multi-track timeline and allow for non-linear editing of animation blocks. This paper aims at synthesising AZee input with such type of software, namely Blender in our case.

We first present two prior approaches complementing each other that worked on animating from AZee, and explain a fundamental limitation found in one of them. We then propose a novel algorithm to animate AZee descriptions that allow for better synthesis. Lastly, we present our implementation in Blender and snapshots of output results we were able to generate.

2. State of the Art

To animate AZee, Filhol et al. (2017) follow a fundamental guiding principle, according to which the coarser the basic animation blocks, the more natural the final animation. To apply this principle, we should try to work from coarse AZee blocks as much as possible and fall back on synthesising from lower levels of AZee specification only if necessary. If this top-down search in the hierarchy of the AZee expression is not attempted, or indeed if it reaches the bottom of the hierarchy, the animation needs to be built from the bottom-up, i.e., work from the minimal articulatory constraints provided by AZee in its block specifications. In this section, we first review the Paula system, the only one attempting a top-down search for synthesis from an AZee description. Then we look into a Blender implementation, the only one proposed for a bottom-up synthesis of AZee.

2.1. Top-Down Approach

The Paula sign synthesis system provides a multi-track animation system close to how AZee describes sign language utterances. The system uses multiple animation techniques, capitalising on their strengths. Currently, it principally relies on pre-animated clips made by artists whose work is made simpler by using procedural techniques such as spine-assisted computation (Mcdonald et al., 2015); hence they do not have to be an expert in keyframe animation or armatures. These clips, representing coarse animation blocks, are essential in encapsulating the natural motions (McDonald et al., 2016) which are vital to improving sign language generation. Furthermore, the system has been extended to enable natural proform synthesis (Filhol and Mcdonald, 2018). Various extensions have been made for better facial model synthesis (Wolfe et al., 2021). Overall, this gives a more natural animation since it encapsulates movements that would be natural to a human signer. All of this is done on a multi-track animation timeline.

Using coarse blocks improves natural synthesis. However, it relies on a large set of shortcut clips, and does not address solving minimal constraints in the case none exists for a given segment.

2.2. Bottom-Up Synthesis: Building from Minimal Constraints

In contrast, a bottom-up approach proposes working from small articulation constraints and then combining and evaluating them to generate an animated utterance on a timeline. Thus, while it generates motion that looks more robotic, it can generate any sign language utterance description, and therefore give complete coverage of the AZee language description. This method of synthesis from AZee was most recently attempted by (Nunnari et al., 2018).

To understand it better, let's consider the AZee expression *nicht-sondern(arbre, armoire)* from their work, which means "not a tree but a wardrobe." Evaluating this expression with the AZee interpreter yields a set of time-bounded intervals arranged on a timeline. These intervals can be represented as blocks on a horizontal axis such as those shown in Figure 1. This arrangement is called an AZee score. Each of these intervals contains articulatory constraints such as, *placements* (e.g. place fingertip at forehead), *orientations* (e.g. orient forearm along upward vector), *transpaths* (e.g. fingertip must transition on a circular path) and *holds* (e.g. hold block UNIT0 for a duration).

In such a score, we notice that these constraints can apply simultaneously Figure 2. For example, PALMS DOWN, which refers to the orientation of palms downwards, while HANDS CONTACT, which refers to the contact of palms. Since both these blocks affect common bones of a bone chain, animating them separately is a problem.

To avoid this problem, Nunnari et al. chose to *flatten* the AZee score to create a linear sequence of keyframes comprising of,

- the constraints corresponding to the boundaries of the original blocks (example k_1, k_2 in Figure 1)
- additional keyframes to control interpolations as specified by transpaths (example k_{12}, k_{13}, ... k_{18} in Figure 1)

Each of the former kinds contains all of the articulatory constraints applied at that time, collecting from any block starting, ending, or crossing over that keyframe. A keyframe of the latter kind contains the same, plus the additional place constraints generated by the transpaths.

When flattening, all the underlying constraints within the blocks are projected on a single timeline. For example, in Figure 1, the constraints in PALMS DOWN and HANDS CONTACT are combined to make one single set of constraints for the keyframe k_9.

This *flattened* score is then used to animate the posture. This is done by resolving the sets of constraints associated with each keyframe in chronological order on the timeline. The constraints are then eventually resolved into the rotation of bone joints. Thus, a posture with n bones can be represented as the following:

$$X(i) = \{bone\_rot_1(i), bone\_rot_2(i), ..., bone\_rot_n(i)\}$$

where X is the state of the posture for the i-th frame.

A problem with this approach is that, even though the system fixes the issue of concurrent constraints depending on each other, it loses the information brought by the parallelism of the blocks while flattening. This means that the only information we have for interpolation are the constraints present on k_1, k_2, ..., and so on. Moreover, every interpolation between each pair of successive keyframes will be distributed on all the bones, including those that should not be affected. Thus, we lose the dynamics of the present blocks, and there is no information on how the system should interpolate amongst these flattened constraints. Also, even if the concurrent blocks comprised of constraints not affecting the same bone chains, there was never a need to flatten in the first place.

In the following section, we propose to fix this using an algorithm that does not flatten by presenting a multi-track bottom-up synthesis of an AZee description.

3. Algorithm for Multi-Track Synthesis

We aim to build a multi-track system without flattening the AZee score. Our work focuses on synthesising the *non-flattened* AZee score in Figure 1. Since the score is constructed based on linguistic descriptions which can be non-linear, we need to impose a certain set of rules while constructing the multi-track timeline, which were previously resolved by flattening the score. Similar to the previous work, we focus on placements and orientation constraints. However, since we are not *flattening*, the *transpath* and *hold* constraints will not be resolved, and we have to deal with them separately.

3.1. Resolving Conflicting Cases

We chose to resolve the conflicting cases by applying the following rules.

3.1.1. Rule 1: Timely Evaluation

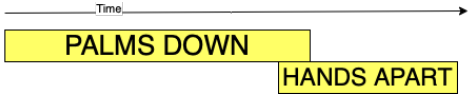

Figure 3: Timely evaluation

Problem: Time overlapping blocks containing constraints that act on the same bone chain but do not start at the same time. For example, in Figure 3, HANDS APART shouldn't be evaluated before PALMS DOWN.

Response: In this scenario(Figure 3), the evaluation of HANDS APART has to account for the fact that the palms are already facing downwards since both blocks act on the same kinematic chain. Thus, to fix this, time overlapping blocks acting on the same bone chains have to be evaluated chronologically.

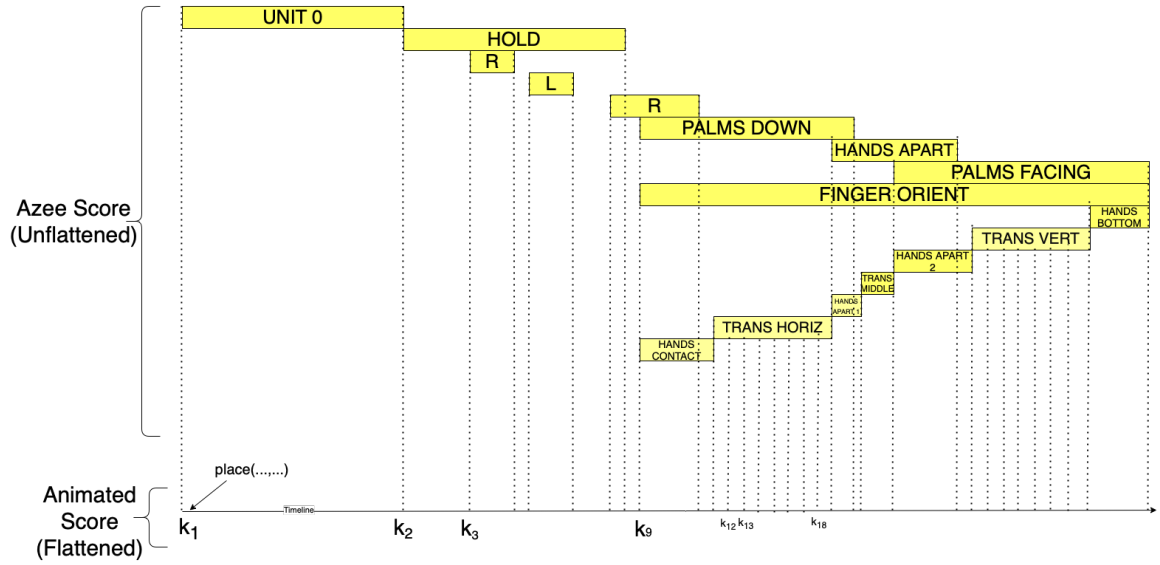

Figure 1: Arrangement of blocks in an AZee score(top) and the equivalent flattened score(bottom)

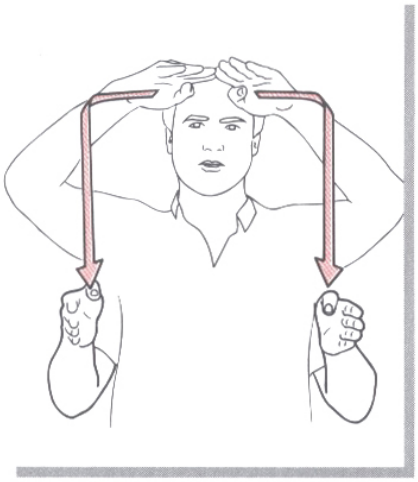

Figure 2: HANDS CONTACT and PALMS DOWN in :armoire (Moody, 1997)

3.1.2. Rule 2: Constraint Precedence

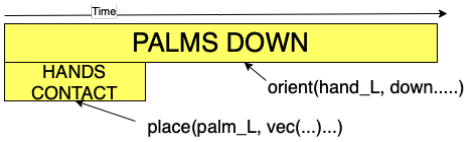

Figure 4: Constraint Precedence

Problem: Time overlapping blocks containing constraints that act on the same bone chain but start at the exact same time. For example, in Figure 4, HANDS CONTACT contains placements while PALMS DOWN contains orientations.

Response: In this scenario(Figure 4), the evaluation of PALMS DOWN has to account for the fact that the hands are already in contact. Thus, to fix this, precedence has to be given to the block containing placement constraints over those with orientation constraints.

3.1.3. Rule 3: Second Pass for Transpaths

Problem: A block contains a transpath constraint.

Response: The transpaths represent transitioning of the posture along some path for an effector of the body. It depends on the evaluation of surrounding blocks and all subsequent interpolations. The solution is, therefore, to evaluate blocks containing transpaths in a Second Pass(Figure 5) after all other blocks have been animated.

3.1.4. Rule 4: Second Pass for Holds

Problem: A block contains a hold constraint.

Response: A block containing the hold constraint specifies that constraints of some other block have to be held for a duration. It, therefore, depends on the animation of that reference block. Hence, blocks containing holds have to be evaluated in a Second Pass (Figure 5) as well.

3.2. Non-Conflicting Cases

Any case not mentioned above will be clear of conflicts and can be evaluated independently. These include:

- all blocks not overlapping each other on the timeline;
- overlapping blocks that act on different bone chains;
- other constraints such as morph and look act independently from the others.

4. Implementation and Experimental Results

The above system has been implemented as an add-on in Blender(v3.1). The interface (Figure 6) shows the Blender interface configured for AZee synthesis. Its main components include:

AZee editor (a) An editor to evaluate AZee expressions. It also includes settings for armature configuration, toggling constraints, and managing body sites.

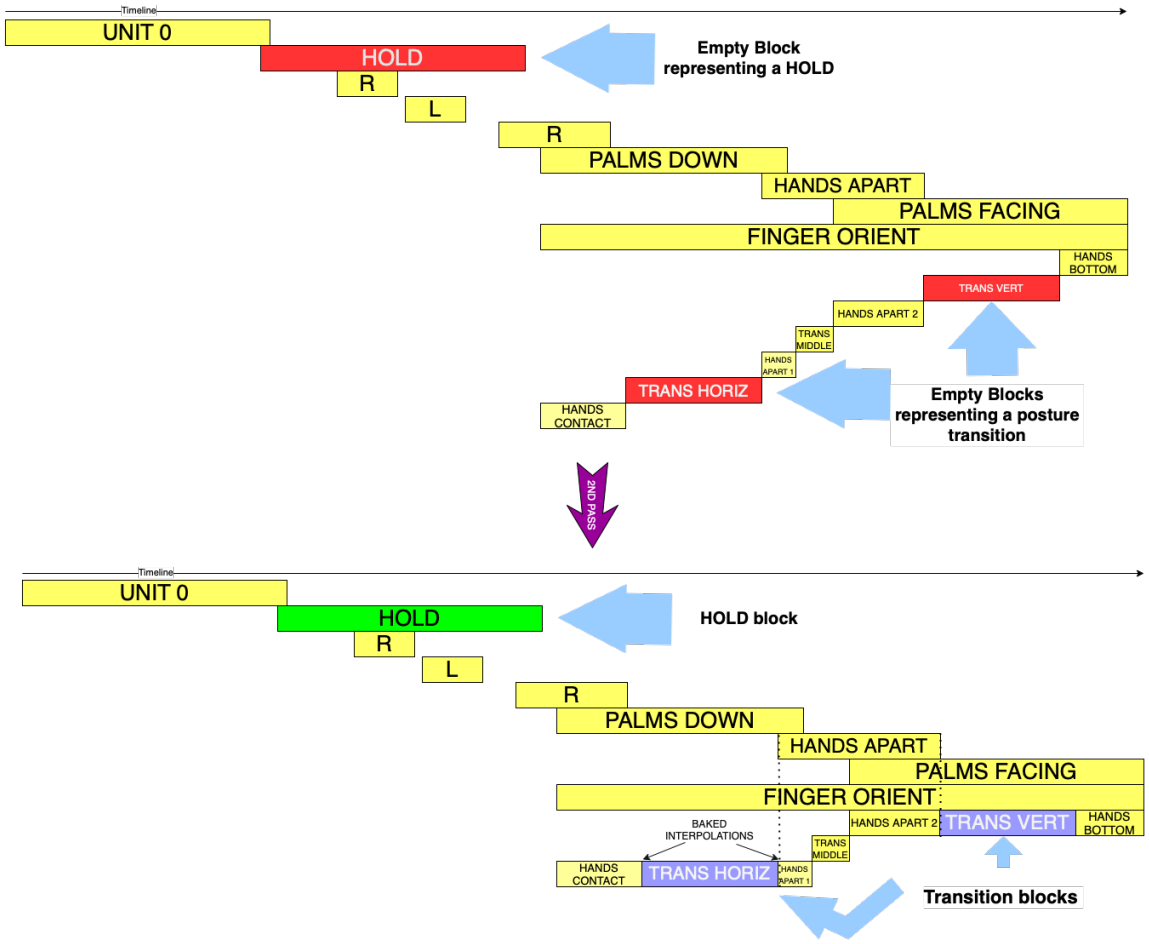

Figure 5: Second Pass to resolve transpaths and holds for *:nicht-sondern(:arbre, :armoire)*

Viewport (b) Shows the 3D scene with the avatar

Non-linear Editor (c) To place all the baked AZee blocks after evaluation.

Properties (d) Modify inverse kinematics (IK) settings, access pose library, and animation layers.

To implement IK solving, we chose to use the iTaSc IK solver (Smits et al., 2008). The reason for that choice is its popularity and that it is already integrated into Blender.
Our implementation is still under development, but the current state of progress already allows to visualise timelines and extract renders, as shown in Figure 7. Here, we present various synthesised AZee descriptions such as *:bien*, *:armoire*, *:arbre*, *:bonjour*.
The current implementation produces satisfactory utterances of simple descriptions but needs more testing and debugging for complex utterances. These occur mainly when there joint orientations get close to the rotation limits. This can be observed in *:armoire* in Figure 7 where the hand rotation limits are reached to satisfy the orientations and placements. But we see that the linguistic constraints on the forearm, hand, and finger orientations, for example, are well satisfied.
As a result of not flattening the score, we preserve the dynamics of individual blocks. This can be seen in *armoire_comparison.mp4* available at `https://doi.org/10.5281/zenodo.6563373` where (A) shows an *:armoire* synthesised using a flattened score while (B) shows the one synthesised using our approach. For (A) we observe that the interpolations are distributed on all bones while for (B) they distribute only over the relevant bones of the blocks shown in the Non-linear Editor.

5. Conclusion and Future Prospects

In this work, we proposed an algorithm that allows for developing the first multi-track animation system for AZee bottom-up synthesis. This proposed algorithm is a step forward in sign language synthesis, allowing for individual AZee blocks to be synthesised independently and ensuring that the dynamics of these blocks are preserved by not flattening. We also integrate our algorithm as an add-on in the open-source Blender software.
Eventually, we want to integrate our work with a top-down technique to have a complete hybrid approach to animate AZee descriptions. The implementation should allow shortcuts using pre-animated clips, MoCap data, or processes that animate these blocks. This would create a system leveraging the advantages of both techniques, as proposed in the AZee–Paula effort.
The current system doesn't resolve AZee morph constraints. More research is needed to handle the bottom-up synthesis of morph constraints and integrate it with our current work. Furthermore, the AZee constraint dependencies

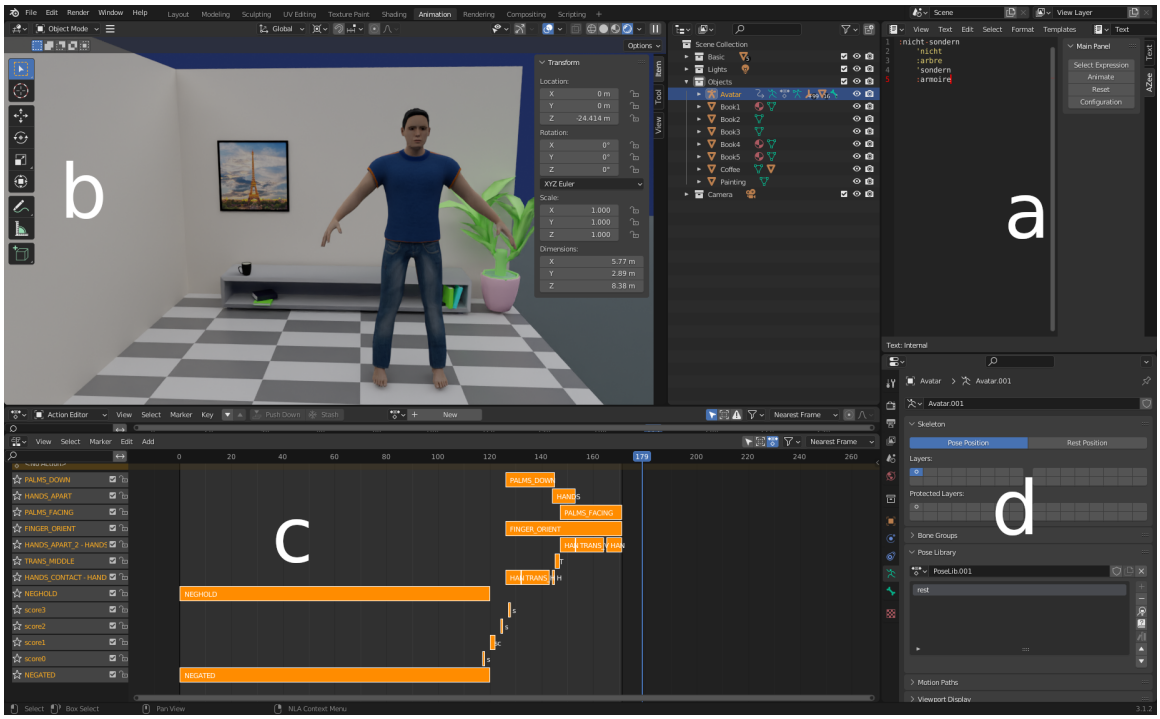

Figure 6: Main Blender interface. (a) AZee editor. (b) 3D Viewport. (c) Non-linear Editor. (d) Properties panel.

(a) *:arbre* (b) *:armoire* (c) *:bien* (d) *:bonjour*

Figure 7: Results

can eventually be mapped as a dependency graph (Zhang et al., 2021; Watt et al., 2012) which can be solved using a multi-pass system.

Lastly, this work can be extended to make the bottom-up synthesis less robotic using ambient noise analysis and style transfer techniques (Holden et al., 2017).

6. Acknowledgement

This work has been funded by the Bpifrance investment "Structuring Projects for Competitiveness" (PSPC), as part of the Serveur Gestuel project (IVès et 4Dviews Companies, LISN — University Paris-Saclay, and Gipsa-Lab — Grenoble Alpes University).

7. Bibliographical References

Bertoldi, N., Tiotto, G., Prinetto, P., Piccolo, E., Nunnari, F., Lombardo, V., Mazzei, A., Damiano, R., Lesmo, L., and Del Principe, A. (2010). On the creation and the annotation of a large-scale Italian-LIS parallel corpus. In Philippe Dreuw, et al., editors, *Proceedings of the LREC2010 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pages 19–22, Valletta, Malta, May. European Language Resources Association (ELRA).

Community, B. O., (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Filhol, M. and Mcdonald, J. (2018). Extending the azee-paula shortcuts to enable natural proform synthesis. In *Workshop on the Representation and Processing of Sign Languages*.

Filhol, M., Hadjadj, M., and Choisier, A. (2014). Non-manual features: the right to indifference. In *International Conference on Language Resources and Evaluation*.

Filhol, M., Mcdonald, J., and Wolfe, R. (2017). Synthesizing sign language by connecting linguistically structured descriptions to a multi-track animation system. pages 27–40, 05.

Holden, D., Habibie, I., Kusajima, I., and Komura, T. (2017). Fast neural style transfer for motion data. *IEEE computer graphics and applications*, 37(4):42–49.

Huenerfauth, M. (2006). *Generating American Sign Language classifier predicates for English-to-ASL machine translation*. Ph.D. thesis, Citeseer.

Kipp, M., Héloir, A., and Nguyen, Q. (2011). Sign language avatars: Animation and comprehensibility. pages 113–126, 09.

Lombardo, V., Nunnari, F., and Damiano, R. (2010). A virtual interpreter for the italian sign language. volume 6356, pages 201–207, 09.

López-Colino, F. J. and Pasamontes, J. C. (2011). The synthesis of lse classifiers: From representation to evaluation. *J. Univers. Comput. Sci.*, 17:399–425.

López-Colino, F. J. and Pasamontes, J. C. (2012). Spanish sign language synthesis system. *J. Vis. Lang. Comput.*, 23:121–136.

Mcdonald, J., Wolfe, R., Hochgesang, J., Jamrozik, D., Stumbo, M., Berke, L., Bialek, M., and Thomas, F. (2015). An automated technique for real-time production of lifelike animations of american sign language. *Universal Access in the Information Society*, 15, 05.

McDonald, J. C., Wolfe, R., Wilbur, R., Moncrief, R., Malaia, E., Fujimoto, S., Baowidan, S., and Stec, J. (2016). A new tool to facilitate prosodic analysis of motion capture data and a datadriven technique for the improvement of avatar motion. In *sign-lang@ LREC 2016*, pages 153–158. European Language Resources Association (ELRA).

Moody, B. (1997). *La langue des signes, dictionnaire bilingue élémentaire*.

Nunnari, F., Filhol, M., and Heloir, A. (2018). Animating azee descriptions using off-the-shelf ik solvers. In *Workshop on the Representation and Processing of Sign Languages*.

Smits, R., De Laet, T., Claes, K., Bruyninckx, H., and De Schutter, J. (2008). itasc: a tool for multi-sensor integration in robot manipulation. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 426–433.

Watt, M., Cutler, L. D., Powell, A., Duncan, B., Hutchinson, M., and Ochs, K. (2012). Libee: A multithreaded dependency graph for character animation. In *Proceedings of the Digital Production Symposium*, DigiPro '12, page 59–66, New York, NY, USA. Association for Computing Machinery.

Wolfe, R., McDonald, J., Johnson, R., Moncrief, R., Alexander, A., Sturr, B., Klinghoffer, S., Conneely, F., Saenz, M., and Choudhry, S. (2021). State of the art and future challenges of the portrayal of facial nonmanual signals by signing avatar. In *International Conference on Human-Computer Interaction*, pages 639–655. Springer.

Zhang, J.-Q., Xu, X., Shen, Z.-M., Huang, Z.-H., Zhao, Y., Cao, Y.-P., Wan, P., and Wang, M. (2021). Write-an-animation: High-level text-based animation editing with character-scene interaction. In *Computer Graphics Forum*, volume 40, pages 217–228. Wiley Online Library.