Thesis Follow-up Abstract

Sign Language synthesis by a decreasing granularity system from AZee

Submitted by
Paritosh Sharma
Email:
paritosh.sharma@universite-paris-saclay.fr

for the thesis follow-up on

26/06/2023

Thesis Director:
Michael Filhol
Email:
michael.filhol@cnrs.fr

External Member:
Alexis Heloir
Email:
alexis.heloir@uphf.fr

Internal Member or Référent:
Tobias Isenberg
Email:
tobias.isenberg@inria.fr

Contents

Chapter 1

Introduction

In human communication, the sign languages are the main languages used by deaf people around the world. Synthesis of these sign languages is a promising method for deaf communication, allowing us to customize and create new sign language content and preserve the artist's anonymity.

Triggering pre-recorded animations from a gloss-based database is a common method for synthesizing sign language [11]. Here, a gloss represents the sign and is mapped to a clip of an avatar performing the sign. However, this technique requires a lot of time and manpower to create these pre-recorded clips. Therefore this method does not scale up well with large sign language utterances and cannot be applied to avatars in virtual worlds where maintaining large databases of animations is not possible.

These problems of scaling and data management have motivated research into the synthesis of sign language with procedural methods [5]. Here, a gloss is mapped to a sequence of motion constraints to be evaluated and synthesized on the avatar. Nonetheless, synthesizing realistic motion with such systems remains a difficult problem, and addressing the signer's prosody, expressivity, and identity by providing control over style is even more challenging.

Glosses have traditionally been used as a formalization of sign language utterances. Yet this imposes the problem of synchronisation and reusability of those signs, which vary with a change in context.

To solve this, the AZee model [3] allows us to write parameterised signed forms for semantic functions. Given a description, it generates a timeline that specifies every aspect of the utterance that the avatar should produce, resolving the problems with timing, sign concurrency, and non-manual features synchronization. Additionally, interpolation information is contained in AZee's temporal specifications, which is crucial for synthesizing the utterance. This has motivated research for data-driven synthesis from AZee [4].

Chapter 2

Subject of the Thesis

The theme of my PhD is the development of a synthesis system which can synthesize AZee descriptions of a sign language discourse through a descending order of granularity. This opens up the following research questions:

- Since AZee defines sign language utterance as a set of blocks or scores, can we synthesize the blocks individually on a multi-track timeline?
- What are the best measures that allow a linguist to constrain a signing avatar for a better low-level synthesis?
- How to better use the pre-animated actions to cover more use cases? For example, parameterizing the motion data for the relevant specification given in the AZee expression.
- How to increase the quality of the low-level synthesis by increasing its naturalness using noise functions and better management of the f-curves using Bézier handles [1].
- How to integrate the two techniques? Since the blocks generated using the low level synthesis would look more robotic than those that used a pre-animated action. Here, applying the motion manifold from the pre-animated action data to solve the posture constraints can be a path to consider for a more seamless utterance generation [6].
- Since AZee itself can be seen as a format to define avatar motion [10] can sign language synthesis be seen as a linguistic motion re-targeting problem?

Chapter 3

Progress

3.1 Multi-Track Representation

The old method to synthesize AZee flattened all the constraints on a timeline. The first contribution of this PhD was to create an algorithm to improve the pre-existing low level 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 low level synthesis 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.

3.2 Layers in Signing Avatars

Using a multi track representation, the blocks of a synthesized utterance can be independently generated using evaluated low-level posture constraints or pre-recorded animations. However, all low-level constraints were generalized as a set of Inverse Kinematics(IK) Problems to solve. For specific scenarios, relying on the IK and joint limits to constrain movement of the posture is not enough. Thus, we introduce a layer-based approach to solving constraints and show how it can be used for a complete data-driven sign language synthesis model.

3.3 Morphs and Facial Expressions

Enhancements in hand shapes and addition of facial expressions at low-level. was done using a new methodology for creating a set of *morphs* inside the AZee language. We encapsulated sets of human movements and map their respective pose space deformations within our morphs. This allows us to capture the rigid as well as the non-rigid shape changes of the human anatomy and also addresses the stretching and contracting of the skin at its extremities. We create our pose space deformations based on the study of local avatar movements and a popular cognitive facial model for facial expressions. We integrate our set of morphs in our existing blender add-on implementation for AZee with a standard parameterized 3D avatar model, resulting in a fully articulated avatar that can produce more realistic movements with a faster real-time synthesis. The proposed

methodology has the potential to enhance the realism of signing avatars and contributes to the development of a more intuitive toolkit for AZee linguists.

3.4 Motion Templates

We proposed a novel method for generating intermediate poses in a multi-track representation of a sign language discourse. The proposed method uses procedural generation with artistic techniques to prioritize certain aspects of the generated poses while sacrificing others to improve the overall consistency of the representation.

Chapter 4

Perspectives

To address the remaining research questions of this work, the following ideas can be incorporated.

- Improvements in the avatar model using SMPL avatar[8].
- Creation of motion templates from motion capture data directly.
- For now, we use morphs for defining hand shapes and facial expressions only. It would be interesting to study more use cases such as for spine-extension, head movements, etcetra.
- We want to improve our facial animation system to have a larger coverage like the FLAME model [7] or Paula [9].
- less robotic low-level synthesis using ambient noise analysis and style transfer techniques [6].

All the work is implemented as an add-on in blender [2] the Figure 4.1 displays a blender window instantiated with the AZee add-on. Its main components include:

(a) Shape Key properties
Modify, add and debug shape keys

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

(c) Non-linear Editor
To place all the animated blocks from the utterance.

(d) Action Editor
Allows us to modify and visualize the generated actions as well as the pre-recorded animations.

(e) AZee editor
An editor to write AZee expressions and change additional settings.

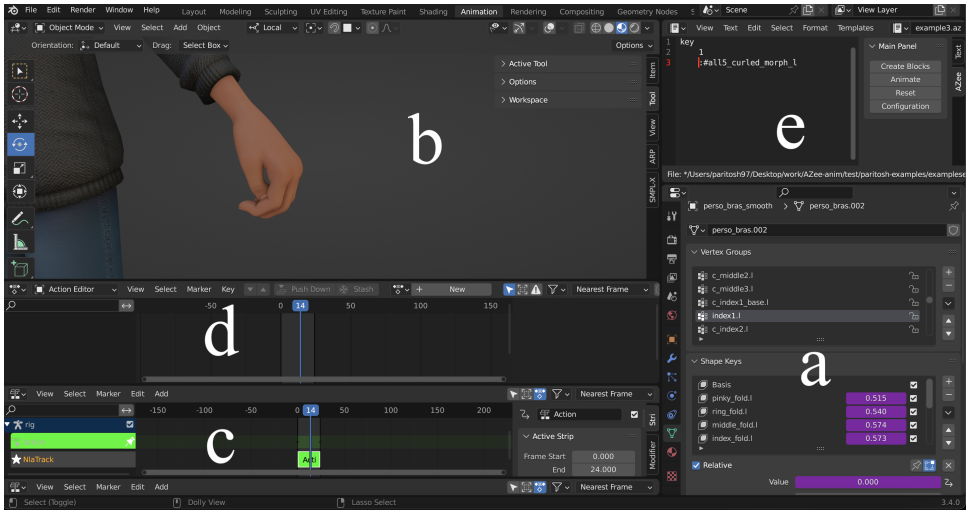


Figure 4.1: Blender interface. (a) Shape Key properties (b) 3D Viewport (c) Non-linear Editor (d) Action Editor (e) AZee editor

Chapter 5

Schedule of Activities

Time Period	**Activities**
Year 1, Semester 1	Implementation improvements, including action labeling and solver enhancements. Fixes were made for IK targets, orientations, and spine inclination. Sync rules and block transitions were added, along with avatar skin mesh. Separate IK and FK mode was added too. The development process involved debugging, meetings, and refining the interface.
Year 1, Semester 2	Key accomplishments included completing and addressing symmetry and transition-related bugs, refining constraints, and resolving orientation issues. Collaborative meetings, literature reviews, and the multi-track paper was presented in SLTAT 2022. Enhancements were made in animation techniques, armature functionality, and facial expression integration. The implementation also saw progress in refining block timings, synchronization scores, and the existing IK was optimised
Year 2, Semester 1	Notable milestones included conducting literature reviews, conference paper acceptances for VISIGRAPP (won best PhD paper award), SLTAT, and ACM SCA. Participation in 3MT competition. Poster presentation on Université Paris-Saclay PhD day. Teaching responsibilities were also successfully managed alongside the research work.

Table 5.1: Schedule of Activities

Bibliography

[1] Dominique Bechmann and Mehdi Elkouhen. "Animating with the "Multidimensional deformation tool"". In: *Computer Animation and Simulation 2001*. Ed. by Nadia Magnenat-Thalmann and Daniel Thalmann. Vienna: Springer Vienna, 2001, pp. 29–35. ISBN: 978-3-7091-6240-8.

[2] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[3] Michael Filhol, Mohamed Hadjadj, and Annick Choisier. "Non-manual features: the right to indifference". In: *International Conference on Language Resources and Evaluation*. Reykjavik, Iceland, 2014. URL: https://hal.archives-ouvertes.fr/hal-01849040.

[4] Michael Filhol, John Mcdonald, and Rosalee Wolfe. "Synthesizing Sign Language by connecting linguistically structured descriptions to a multi-track animation system". In: *11th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2017) held as Part of HCI International 2017*. Ed. by Constantine Stephanidis Margherita Antona. Vol. 10278. Universal Access in Human–Computer Interaction. Designing Novel Interactions. Vancouver, Canada: Springer, July 2017. DOI: 10.1007/978-3-319-58703-5_3. URL: https://hal.archives-ouvertes.fr/hal-01849419.

[5] SYLVIE GIBET, THIERRY LEBOURQUE, and PIERRE-FRANCOIS MARTEAU. "High-level Specification and Animation of Communicative Gestures". In: *Journal of Visual Languages & Computing* 12.6 (2001), pp. 657–687. ISSN: 1045-926X. DOI: https://doi.org/10.1006/jvlc.2001.0202. URL: https://www.sciencedirect.com/science/article/pii/S1045926X01902022.

[6] Daniel Holden et al. "Fast Neural Style Transfer for Motion Data". In: *IEEE Computer Graphics and Applications* 37.4 (2017), pp. 42–49. DOI: 10.1109/MCG.2017.3271464.

[7] Tianye Li et al. "Learning a model of facial shape and expression from 4D scans". In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36.6 (2017), 194:1–194:17. URL: https://doi.org/10.1145/3130800.3130813.

[8] Matthew Loper et al. "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[9] John Mcdonald et al. "An automated technique for real-time production of lifelike animations of American Sign Language". In: *Universal Access in the Information Society* 15 (May 2015). DOI: 10.1007/s10209-015-0407-2.

[10] Fabrizio Nunnari, Michael Filhol, and Alexis Heloir. "Animating AZee Descriptions Using Off-the-Shelf IK Solvers". In: *Workshop on the Representation and Processing of Sign Languages*. Miyazaki, Japan, May 2018, pp. 155–162. URL: https://hal.archives-ouvertes.fr/hal-01836486.

[11] F. Pezeshkpour et al. "Development of a legible deaf-signing virtual human". In: *Proceedings IEEE International Conference on Multimedia Computing and Systems*. Vol. 1. 1999, 333–338 vol.1. DOI: 10.1109/MMCS.1999.779226.

Appendix: Published Papers

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

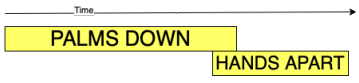


Figure 3: Timely evaluation

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.

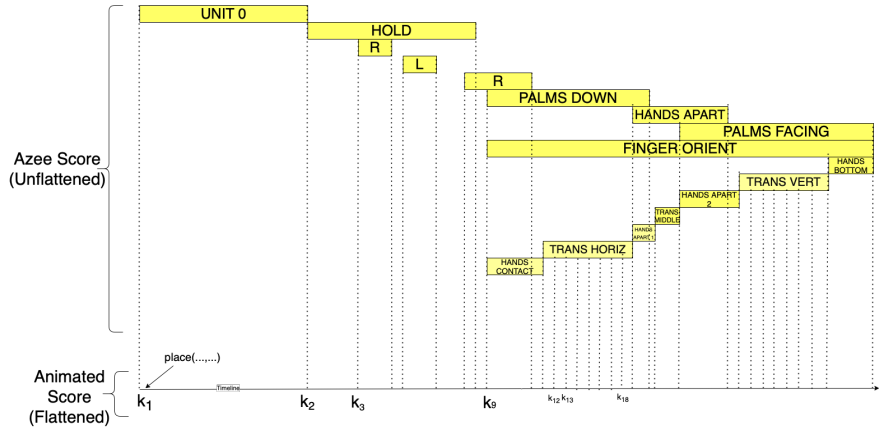


Figure 1: Arrangement of blocks in an AZee score(top) and the equivalent flattened score(bottom)

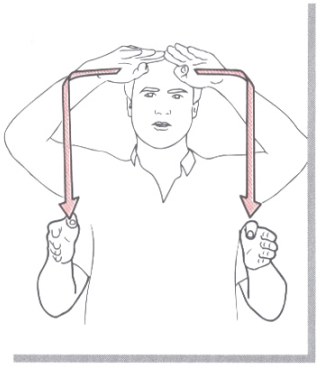


Figure 2: HANDS CONTACT and PALMS DOWN in :armoire (Moody, 1997)

3.1.2. Rule 2: Constraint Precedence

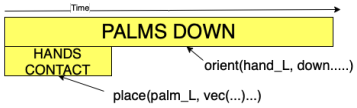


Figure 4: Constraint Precedence

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.

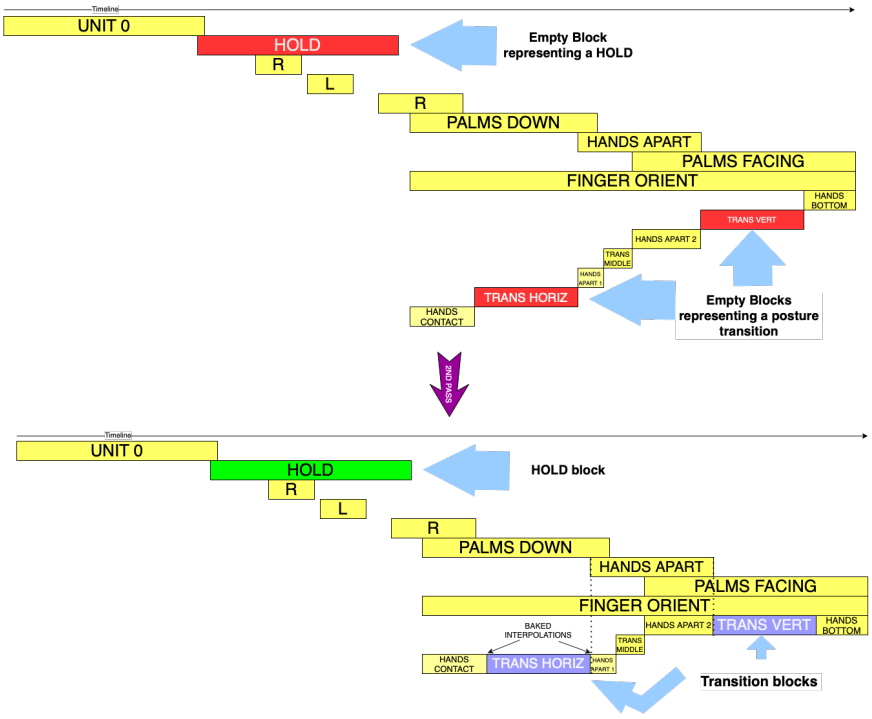


Figure 5: Second Pass to resolve transpaths and holds for *:nicht-sondern(:arbre, :armoire)*

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

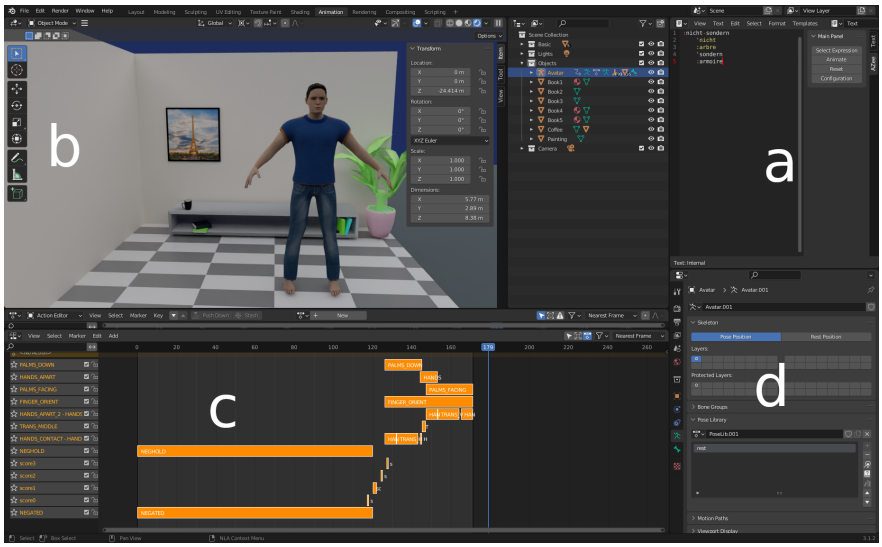
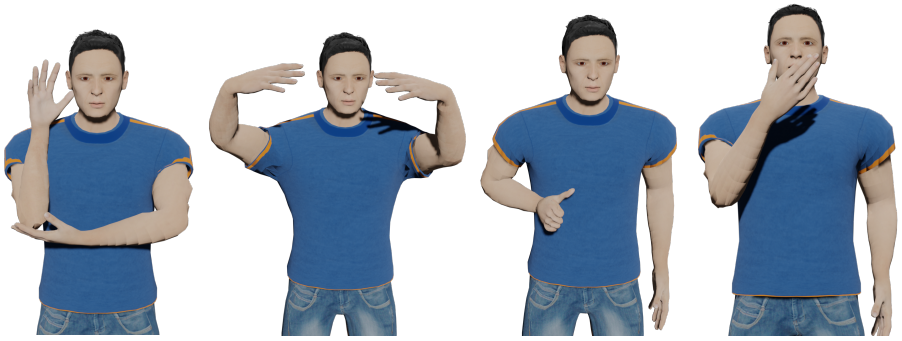


Figure 6: Main Blender interface. (a) AZee editor. (b) 3D Viewport. (c) Non-linear Editor. (d) Properties panel.



(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.

A Layered Approach to Constrain Signing Avatars

Paritosh Sharma[1][a]
[1]*LISN, CNRS, Université Paris–Saclay, Orsay, France*
paritosh.sharma@lisn.upsaclay.fr

1 RESEARCH PROBLEM

In human communication, the sign languages are the main languages used by deaf people around the world. Synthesis of these sign languages is a promising method for deaf communication, allowing us to customize and create new sign language content and preserve the artist's anonymity.

Triggering pre-recorded animations from a gloss-based database is a common method for synthesizing sign language (Pezeshkpour et al., 1999). Here, a gloss represents the sign and is mapped to a clip of an avatar performing the sign. However, this technique requires a lot of time and manpower to create these pre-recorded clips. Therefore this method does not scale up well with large sign language utterances and cannot be applied to avatars in virtual worlds where maintaining large databases of animations is not possible.

These problems of scaling and data management have motivated research into the synthesis of sign language with procedural methods (GIBET et al., 2001). Here, a gloss is mapped to a sequence of motion constraints to be evaluated and synthesized on the avatar. Nonetheless, synthesizing realistic motion with such systems remains a difficult problem, and addressing the signer's prosody, expressivity, and identity by providing control over style is even more challenging.

Glosses have traditionally been used as a formalization of sign language utterances. Yet this imposes the problem of synchronisation and reusability of those signs, which vary with a change in context.

To solve this, the AZee model (Filhol et al., 2014) allows us to write parameterised signed forms for semantic functions. Given a description, it generates a timeline that specifies every aspect of the utterance that the avatar should produce, resolving the problems with timing, sign concurrency, and non-manual features synchronization. Additionally, interpolation information is contained in AZee's temporal specifications, which is crucial for synthesizing the utterance.

This has motivated research for data-driven synthesis from AZee (Filhol et al., 2017). A synthesized utterance can be depicted as a set of blocks on a muti-track timeline (Sharma and Filhol, 2022). These blocks can be generated using evaluated low-level posture constraints or pre-recorded animations. However, all low-level constraints were generalized as a set of Inverse Kinematics(IK) Problems to solve. For specific scenarios, relying on the IK and joint limits to constrain movement of the posture is not enough. Thus, we introduce a layer-based approach to solving constraints and show how it can be used for a complete data-driven sign language synthesis model.

[a] https://orcid.org/0000-0001-9938-008X

2 OUTLINE OF OBJECTIVES

The objectives of this work can be summarized as follows:

- Show problems with only IK-based constraining.
- Present a new layered-based approach to better synthesize AZee constraints.

3 STATE OF THE ART

Animation from AZee descriptions can be divided into two categories: pre-animated and bottom-up synthesis(building from minimal constraints). Pre-animated methods use explicit, often manually created, mappings of utterance description to motion data. (Filhol and Mcdonald, 2018) also uses template utterance descriptions and facilitates the generation of utterances with parameterized motion sequences. However, the diversity of these generated utterances is limited to the number of designed animations and database content. Moreover, the required manual labour hinders scalability.

The Bottom-Up synthesis creates motion from minimalist constraints rather than relying on pre-animated mappings. Despite recent research efforts, the naturalness of generated motion still falls significantly behind that of a pre-animated motion. However, it provides a broader coverage since it doesn't

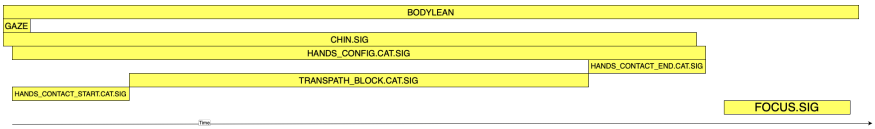


Figure 1: Arrangement of the utterance for expression in section 3 on the timeline

rely on pre-recorded motion data and hence, can produce all motion specified by the utterance description.

The AZee native level defines several basic types to constrain an armature posture. To understand it better, let's consider the following AZee expression from the corpus *40 brèves* (Challant and Filhol, 2022) (Filhol and Tannier, 2014) (LIMSI and LISN, 2022)

```
:about-point
    'pt
    ^Rssp
    'locsig
    :category
        'cat
        :pays
        'elt
        :Irak
```

The above AZee expression is a representation of an SL production assigning "Iraq" to a point on the right-hand side of the signing space, a way of creating a reference the signer can later point to refer to the country. Let's assume we have pre-recorded action for the rule *:Irak*. fig. 1 represents this utterance on a multi-track timeline. The blocks constrain the posture in the following ways:

- **HANDS_CONTACT_START.CAT.SIG**: Placement constraints to keep the fingertips in contact in the beginning.
- **HANDS_CONTACT_END.CAT.SIG**: Placement constraints to keep the fingertips in contact at the end.
- **TRANSPATH_BLOCK.CAT.SIG**: Transpath constraint specifying the movement of hands along an arc
- **HAND_CONFIG.CAT.SIG**: Constraints for finger configuration.
- **CHIN.SIG**: Constrain the chin up
- **GAZE**: Constrain the gaze to the right signing space
- **BODYLEAN**: Orient the back so it leans towards the right.
- **CHIN.SIG**: Constrain the chin up
- **FOCUS.SIG**: pre-recorded action for the rule *:Irak*

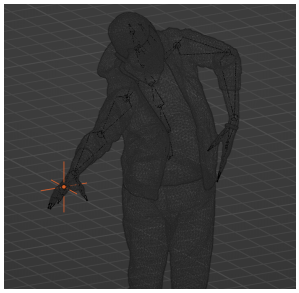


Figure 2: Hand IK chain pulling the spine and the shoulder

During block application, the IK chain to be chosen is based on the joint dependencies to evaluate the *placement* constraints. Though this system does allow for the extension of the IK chain through the spine, it invokes the solver with a new chain every time it applies a placement. This results in a slower evaluation of placements; moreover, the arms and torso have different purposes in the human body (fig. 2). The torso movement could have its own meaning irrespective of the movement of the arm. Lastly, whenever we switch the chain for the next constraint, we lose the IK information required by the other constraints, such as *transpaths*(movement specified along a path) or other placements.

To fix this, some systems define the spine, and arm IK separately (Baerlocher and Boulic, 2004) (Elliott et al., 2008). (McDonald et al., 2016) present the use of spine and shoulder extensions with an analytical hand IK model to address the timing of spine and shoulder movement separately. This allows for more natural bust movement. Avatar layers based on behaviour, skeleton and muscles were initially introduced by (Chadwick et al., 1989). This interests us since we aim to define our posture based on its behaviour w.r.t. linguistic constraints. Thus, in the following section, we propose a layer-based posture configuration to solve and apply the native AZee constraints.

Figure 3: Use of IK behaviour in HANDS_CONTACT_START.CAT.SIG block from fig. 1

4 METHODOLOGY

The goal of our method is to constrain our posture using layers and specify the relationship between the layers through the native AZee constraints. (Chadwick et al., 1989) define a layer as a conceptual simulation model which maps higher-level parametric input into lower-level outputs. Thus, using layers of the character related through the AZee constraints, we aim to add higher-level control over the low-level skeleton specification.

We define our posture using the following three behaviour layers:

4.1 IK Behaviour layer

The IK behaviour layer represents the motion specifications for the skeleton's arms and fingers. Having pre-defined chains in a layer also allows for better evaluation of *transpath* constraints. This layer encapsulates the chain movements for each arm bone and finger phalanges. When the IK is applied, a numerical IK solver generates the joint rotations for each bone in the chain. The mesh deformations are generated by weight painting (Mohr et al., 2003) based on a given skeletal state.

4.2 Forward Kinematics(FK) Behaviour layer

The FK behaviour layer constitutes the motion specifications for all FK bones. It is used for all the local or global bone rotation changes in the skeleton. The constraints *orient*, *rotate*, *trill*, *look* and *lookat* use the FK layer.

Figure 4: Use of FK behaviour in CHIN.SIG block from fig. 1

4.3 Morph Behaviour layer

Certain linguistic constraints are more suitable to be evaluated using pre-defined shape keys rather than as other constraints in the IK or FK behaviour layers. For example, the following AZee expression constrains the posture to close its little finger.

```
L_closed
azop
    'param$0
    'nodefault
    orient
        'dir
        !little
            ^param$0
            2
        'along
        oppvect
            dir
            !palm
                ^param$0
```

However, this method is slower since it generates vectors for constraining each of the joints in the finger. When used for each finger, a rule like fist_closed will generate 20 vectors for each joint. Thus, morphs make it more suitable to define a part of the sign language motion space. The following conditions can be considered when defining morphs,

- The morph action has a linguistic meaning(fist_closed, brow_raised, etc.)
- The morph action evaluates to local rotations on the skeleton or to some shape of the mesh independent of the skeleton layer.

Thus, the above expression can be rewritten as,

Figure 5: Evaluated morph for $little\_closed(w) = 0.5$

```
L_closed
azop
    'param$0
    'nodefault
    morph
        'little_closed
    ^param$0
        1.0
```

A set of morphs is pre-defined on the morph layer and then applied to the posture with the specified weight during block evaluation.

4.4 Ordering the constraints

Once the Score is generated, for each block, the constraints have to be sorted based on their dependencies which can be represented as a dependency graph. The topological sorting of this graph gives us a set of sorted constraints. These sorted constraints are then used to create the dependency graph of the blocks, which determines the order of block evaluation. fig. 6 shows us the blocks for fig. 1 with their first constraint and edges representing dependencies.

4.5 Updating the layers

Since each layer affects the low-level skeleton specification, the other layers have to be updated with the skeleton as well during the constraint application.

5 IMPLEMENTATION

5.1 Pre-animated Dataset and Data Preparation

Based on our defined behaviour layers, we created simple animations for rules such as *:Irak*, *:cuisine*, etcetera. To use this action dataset effectively, we map the varying behaviour layer control node to the parameters from the AZee expression. fig. 7 shows the rule *:Irak* while the body leans towards right signing space.

5.2 Blender add-on

We implement our animator as an add-on in Blender(v3.4) (Community, 2018). fig. 8 shows the Blender interface configured with the AZee animator addon. Its main components include:

(a) Properties
Modify inverse kinematics (IK) settings and animation layers.

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

(c) Non-linear Editor
To place all the baked blocks from the utterance.

(d) Action Editor
The third etc allows us to modify and visualize the generated actions as well as the pre-recorded animations.

(e) AZee editor
An editor to evaluate AZee expressions. It also includes settings for armature configuration, toggling constraints, managing body sites and defining global signing space and camera position.

We use the AutoRigPro (Artell, 2023) add-on to implement the posture layers since it has pre-defined IK and FK switching mechanisms important for updating our layers.

6 EXPECTED OUTCOME

Our implementation is still under development. However, the current implementation allows us to visualise the AZee block timeline and generate simple utterances using both pre-recorded animations and minimal constraints. Synthesis for the utterance in fig. 1 can be seen at the following link.

https://github.com/Paritosh97/phd/raw/master/grapp-2023/outcome.mp4

We see that the animator can synthesise AZee expressions using both bottom-up and pre-animated techniques.

7 STAGE OF THE RESEARCH

The theme of my PhD is the development of a synthesis system which can synthesize AZee descriptions of

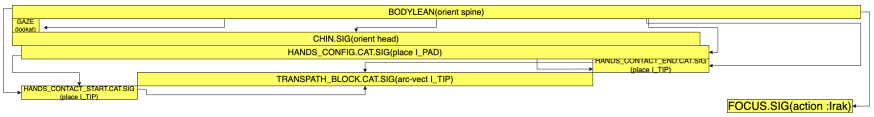


Figure 6: Block DAG for expression in section 3

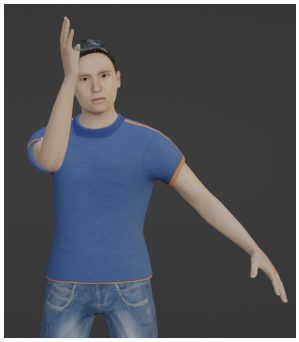


Figure 7: *:lrak* with body leaning towards the right signing space

a sign language discourse through a descending order of granularity. As explained earlier, we use pre-animated and bottom-up constraint synthesis for utterance generation. Thus, we can summarize the future goals of the PhD as follows.

- Improve usage of the pre-animated actions by parameterizing the motion data for the relevant specification given in the AZee expression.
- Increasing the quality of our bottom-up synthesis by increasing its naturalness using noise functions and better management of the f-curves using Bézier handles (Bechmann and Elkouhen, 2001).
- Integrating the above two techniques since the blocks generated using the bottom-up synthesis would look more robotic than those that used a pre-animated action. Here, applying the motion manifold from the pre-animated action data to solve the posture constraints can be a path to consider for a more seamless utterance generation (Holden et al., 2017).
- Testing and debugging our blender implementation for more complex utterances.

ACKNOWLEDGEMENTS

This work has been funded by the Bpifrance investment “Structuring Projects for Competitiveness” (PSPC), as part of the Serveur Gestuel project. Special thanks to my PhD director, Dr Michael Filhol, for providing guidance and feedback throughout this research.

REFERENCES

Artell (2023). Auto-rig pro.

Baerlocher, P. and Boulic, R. (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20:402–417.

Bechmann, D. and Elkouhen, M. (2001). Animating with the “multidimensional deformation tool”. In Magnenat-Thalmann, N. and Thalmann, D., editors, *Computer Animation and Simulation 2001*, pages 29–35, Vienna. Springer Vienna.

Chadwick, J. E., Haumann, D. R., and Parent, R. E. (1989). Layered construction for deformable animated characters. *ACM Siggraph Computer Graphics*, 23(3):243–252.

Challant, C. and Filhol, M. (2022). A First Corpus of AZee Discourse Expressions. In *Language Resources and Evaluation Conference*, Proceedings of the 13th Language Resources and Evaluation Conference, Marseille, France.

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

Elliott, R., Glauert, J. R., Kennaway, J., 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(4):375–391.

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

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*, Miyazaki, Japan.

Filhol, M., Mcdonald, J., and Wolfe, R. (2017). Synthesizing Sign Language by connecting linguistically structured descriptions to a multi-track animation system. In Margherita Antona, C. S., editor, *11th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2017) held as Part of HCI International 2017*, volume 10278 of *Universal Access in Human–Computer Interaction. Designing Novel Interactions*, Vancouver, Canada. Springer.

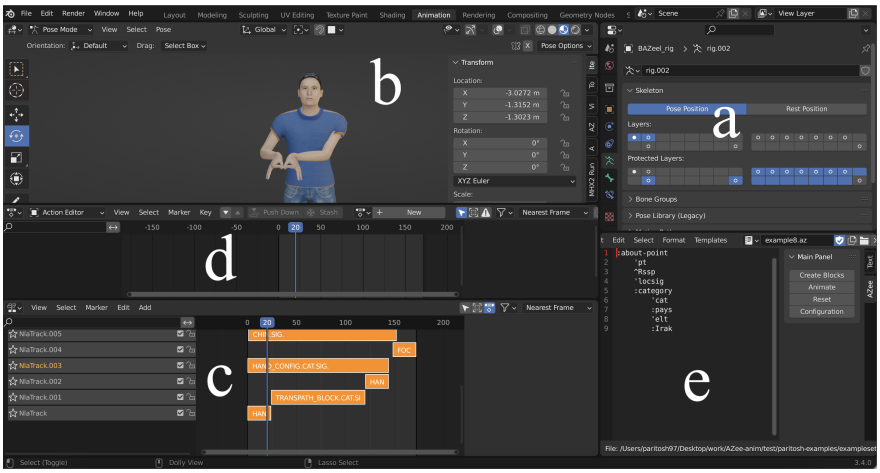


Figure 8: Blender interface. (a) Properties. (b) 3D Viewport. (c) Non-linear Editor. (d) Action Editor. (e) AZee editor

Filhol, M. and Tannier, X. (2014). Construction of a French-LSF corpus. In *Workshop on Building and Using Comparable Corpora*, Reykjavík, Iceland.

GIBET, S., LEBOURQUE, T., and MARTEAU, P.-F. (2001). High-level specification and animation of communicative gestures. *Journal of Visual Languages & Computing*, 12(6):657–687.

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.

LIMSI and LISN (2022). 40 brèves. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

McDonald, J., Wolfe, R., Schnepp, J., Hochgesang, J., Jamrozik, D. G., 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(4):551–566.

Mohr, A., Tokheim, L., and Gleicher, M. (2003). Direct manipulation of interactive character skins. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 27–30.

Pezeshkpour, F., Marshall, I., Elliott, R., and Bangham, J. (1999). Development of a legible deaf-signing virtual human. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 333–338 vol.1.

Sharma, P. and Filhol, M. (2022). Multi-Track Bottom-Up Synthesis from Non-Flattened AZee Scores. In *7th Workshop on Sign Language Translation and Avatar Technology: The Junction of the Visual & the Textual Challenges and Perspectives (SLTAT 7)*, Marseille, France.

EXTENDING MORPHS IN AZEE USING POSE SPACE DEFORMATIONS

Paritosh Sharma, Michael Filhol

LISN, CNRS, Université Paris-Saclay, Orsay, France

ABSTRACT

Signing avatars have become increasingly important for sign language synthesis. However, to behave realistically, they must be able to replicate the coordinated activity of human hand movements and facial expressions. Most methods currently evaluate such motion using just kinematic techniques, which can limit the realism of the virtual characters. We propose a new methodology for creating a set of *morphs* in the AZee language to address this issue. We encapsulate a set of human movements and map their respective pose space deformations within our morphs. This allows us to capture the rigid as well as the non-rigid shape changes of the human anatomy and also addresses the stretching and contracting of the skin at its extremities. We create our pose space deformations based on the study of local avatar movements and a popular cognitive facial model for facial expressions. We integrate our set of morphs in our existing blender add-on implementation for AZee with a standard parameterized 3D avatar model, resulting in a fully articulated avatar that can produce more realistic movements with a faster real-time synthesis. The proposed methodology has the potential to enhance the realism of signing avatars and contributes to the development of a more intuitive toolkit for AZee linguists.

Index Terms— Sign Language, Avatar, AZee, Morphs

1. INTRODUCTION

Procedural synthesis of sign language is a technique for creating the animation of a signing avatar based on a list of low-level motion constraints to be evaluated and synthesized on the avatar. The AZee model [1] allows us to synthesize a multi-track animation timeline specifying all parts of the utterance to render with the avatar [2]. This allows customisation and creation of new Sign Language content without requiring pre-animated data for the corresponding elements in the description.

One of the challenges in procedural synthesis is generating correct shape configurations inside the avatar's motion space. Using Inverse Kinematics(IK) or Forward Kinematics(FK) to calculate the joint rotations for shape configuration often leads to poses that either look unnatural or are incorrect since the description doesn't generalize to all avatars. Additionally, writing low-level descriptions to represent these shapes is often difficult for a linguist.

To address these issues, we propose a methodology for defining new sets of morphs that can be used to synthesize hand shapes and facial expressions, improving the synthesis of sign language gestures using the AZee model. Morph target animation, a computer animation technique that involves blending between different pre-defined shapes or *morph targets*, can be used to overcome the problems with just kinematic-based models as it only requires a single pre-defined morph to obtain intermediate poses. We aim to embed these morphs into our animation system, map them to their poses on our avatar, and add them to our existing Blender add-on for simpler, faster, and better shape synthesis.

2. RELATED WORK

We begin our discussion by first reviewing procedural Sign Language synthesis. Then we explore the use of morph target animation methods in computer graphics. Finally, we discuss the use of morph targets in signing avatars.

2.1. Procedural Sign Language Synthesis

Procedural Sign Language synthesis is an emerging area of research that aims to generate sign language animations from linguistic descriptions. The idea is to generate animations on the avatar directly from the description without the need for pre-recorded motion data. This has several applications, especially when the goal is to minimize the amount of pre-recorded motion data.

To do this, a set of constraints (provided by the linguistic description) act on the posture for a certain time [3]. The animation is generated by applying these constraints to the avatar's anatomy. To do this, constraints are evaluated using IK and FK techniques to generate postures. Finally, interpolations are applied to generate motion.

For example, we can define the shape of a closed index finger in AZee as follows,

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 GipsaLab — Grenoble Alpes University).

```
orient
 'dir
 !index
  ^side
  2
 'along
 oppvect
  dir
   !palm
    ^side
```

The above expression instructs the posture to close the index finger of *side* by orienting the index FK bone in the direction opposite to the palm.

Since the technique uses the skeleton space of the avatar, it often results in unnatural synthesized shapes, as shown in figure 1. Furthermore, with complex expressions, it is difficult for a linguist to write such code for a variety of hand shapes.

Fig. 1: Unnatural shape synthesis of the hand shapes using the kinematic model to generate hand shapes with AZee

2.2. Morph Target Animation in Signing Avatars

Morph target animation is a type of computer animation technique that involves changing the shape of a 3D model by blending between different pre-defined shapes or *morph targets*. Morph target animation is especially useful in character animation scenarios when the goal is to have more control over the movements because it allows us to key-frame the composite geometry of the mesh. A popular application of morph targets is facial animation [4].

A set of morph targets is required to synthesize facial animations [5] and was used to further extend the JASigning system [6]. Similarly, the EMBR system [7], the Paula animation system [8] [9], and the SIGNCom system [10] use morphs to synthesize facial animations as well.

Along with facial animation, morphing can also be used as a Pose Space Deformation(PSD) [11]. A PSD is a hybrid method that combines skeleton space deformations(SSD) with morphing and employs scattered data interpolation to compute non-linear skin corrections in pose space. This gives us a kinematic model that also has poses which can be pre-sculpted by artists.

A similar approach to control deformable material using Dynamic Morph Targets derived from these PSDs was used by [12].

Thus, defining pose space for specific skeleton joints allows us to use morph targets for not just the geometry of the mesh but also the skeletal representation of the avatar.

3. METHODOLOGY

The AZee language tentatively defines morph constraints as a morph target with some weight. Hence, implementing a set of motion space using morphs is appealing as it simplifies the work of the linguist. However, AZee doesn't have definitions of the motion space, which could be covered with these morphs. The syntax of a morph expression can be defined as follows:

```
morph
    'morph_id
    weight[0, 1]
```

where $morph\_id$ is the name of the morph with which the language is extended, and $weight[0, 1]$ represents the amount of weight applied to the given morph.

Additionally, in previous works, a *morph* was referred to as a non-skeletal articulatory constraint. Here, we redefine morph as a constraint which can constrain both the skeleton and the mesh. The target motion space of a morph can be mapped to a single parameter, its weight. This makes morph a local constraint on the avatar.

3.1. Skeletal morphs

In this section, we discuss the parts of morph motion space which depend on the skeleton.

3.1.1. Adduction and Abduction

An *Adduction* refers to the movement of a limb or other part towards the mid-line of the body. On the contrary, an *Abduction* is a limb's movement away from the body's mid-line. Figure 2 shows the adduction and abduction of the palm.

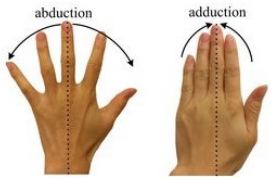


Fig. 2: Adduction and Abduction of the palm [13]

Although adduction and abduction generalise to various joints, we are concerned only by the scenarios representing a local movement.

3.1.2. Extension and Flexion

Extension refers to an extension or bending movement of a joint that increases the angle between two bones. *Hyper-extension* refers to an extension beyond its normal range of motion, typically in a backward direction. Lastly, *Flexion* is the opposite of Extension and refers to a bending movement of a joint that decreases the angle between two bones. Figure 3 shows the index finger's extension, hyper-extension, and flexion.

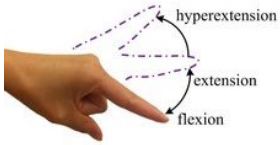


Fig. 3: Hyper-extension, Extension and Flexion of the index finger [13]

Just like adduction and abduction, we are only concerned by scenarios with local hyper-extension and flexion of the joints.

3.2. Non-Skeletal morphs

Apart from the skeletal morphs, we also want to use morphs for detailing facial characteristics. For this, we use a subset of the FACS model [14] for our facial morphs. We chose the model for its simplicity and as a baseline and aim to have a bigger set in the future.

We choose not to include the action units which correspond to global dependencies. These include Action Units(AUs) 51 to 60, which correspond to head movements AUs 61 to 69, which correspond to eye movements.

Fig. 4: AU 61(eye turn left) not implemented as morph since we already have *look* and *lookat* constraints in AZee

Fig. 5: Facial expression in rule *:inter-subjectivity* by combining AU18 and AU22

morph_id	Movement
I_closed	Hyperextension and Flexion of index fingers
M_closed	Hyperextension and Flexion of middle fingers
R_closed	Hyperextension and Flexion of ring fingers
L_closed	Hyperextension and Flexion of little fingers
T_closed	Hyperextension and Flexion of thumbs
palm_extended	Adduction and Abduction of the palms

Table 1: First Set of AZee Morphs

4. RESULTS

Based on the above methodology, we get a new list of morphs and the respective skeletal movements as shown in table 1, which can be used to extend the low-level constraints of AZee. We use this list to redefine the AZee vocabulary of morph IDs. Which can be used to create poses which can be synced independently.

We also get another set of facial morphs based on FACS, which can be combined for facial expressions such as in *:inter-subjectivity*(figure 5).

5. IMPLEMENTATION

We implement our set of morphs as shape keys in blender [15]. We also use FACSHuman [16] to extract the relevant set of FACS shape keys for our MakeHuman [17] based blender avatar. To use this shape key dataset, we map the AZee morph definitions with our defined shape key names in a *skeleton.morphmap* which is initialized with our avatar posture.

We extend our AZee animator add-on in Blender for

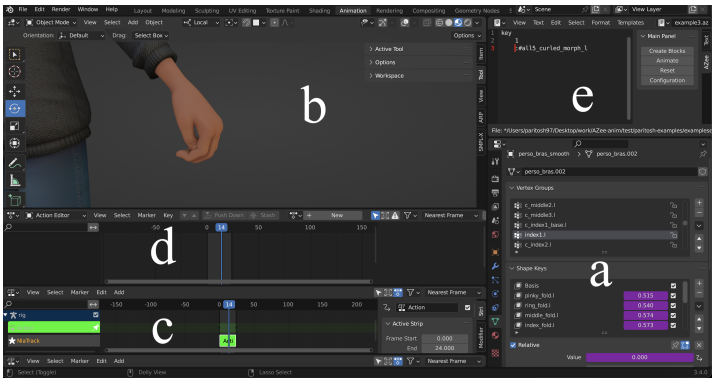


Fig. 6: Blender interface. (a) Shape Key properties (b) 3D Viewport (c) Non-linear Editor (d) Action Editor (e) AZee editor

morph support. Figure 6 shows the Blender interface configured with the new shape keys. Its main components include:

(a) Shape Key properties
Modify, add and debug shape keys

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

(c) Non-linear Editor
To place all the animated blocks from the utterance.

(d) Action Editor
Allows us to modify and visualize the generated actions as well as the pre-recorded animations.

(e) AZee editor
An editor to write AZee expressions and change additional settings.

5.1. Outcome

Our blender add-on can be used with AZee morphs. We observe significant improvements in the synthesis of hand shapes compared to the previous approach, which can be seen in figure 7. Figure 5 shows the facial expression for the rule *:inter-subjectivity*, which was synthesized using the morphs linked to the FACSHuman extracted shape keys.

6. CONCLUSION AND FUTURE WORK

We presented a methodology to extend the low-level of AZee with a new set of morphs. This allowed us to map low-dimensional pose space deformations to AZee morphs and produces better shapes, is easier to pose for the linguist and is faster at run-time since it is based on pre-recorded animation data.

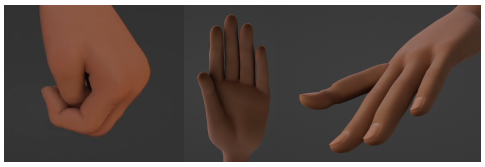


Fig. 7: Improvements in the shape synthesis compared to the kinematic approach in figure 1

Our system still has some limitations which we want to address in the future:

Larger Use Cases For now, we use morphs for defining hand shapes and facial expressions only. I would be interesting to study more use cases such as for spine-extension, head movements, etcetra.

Low Coverage We want to improve our facial animation system to have a larger coverage like the FLAME model [18] or Paula [8].

Naturalness Our morph interpolations look robotic. One way to improve this could be to modify the bezier handles of the underlying f-curves between morph-based poses based on the morph.

Universality We introduce an additional step i.e. creation and mapping of shape keys which could get cumbersome when re-targeting because the same shape keys may not work on different types of avatars. A potential solution to this could be to explore other models for avatars like the SMPL model [19].

7. REFERENCES

[1] Michael Filhol, Mohamed Hadjadj, and Annick Choisier, "Non-manual features: the right to indifference," in *International Conference on Language Resources and Evaluation*, 2014.

[2] Paritosh Sharma and Michael Filhol, "Multi-Track Bottom-Up Synthesis from Non-Flattened AZee Scores," in *7th Workshop on Sign Language Translation and Avatar Technology: The Junction of the Visual & the Textual Challenges and Perspectives (SLTAT 7)*, Marseille, France, June 2022.

[3] Alexis Heloir and Michael Kipp, "Embr–a realtime animation engine for interactive embodied agents," in *Intelligent Virtual Agents: 9th International Conference, IVA 2009 Amsterdam, The Netherlands, September 14-16, 2009 Proceedings 9*. Springer, 2009, pp. 393–404.

[4] Zhigang Deng and Junyong Noh, "Computer facial animation: A survey," *Data-driven 3D facial animation*, pp. 1–28, 2008.

[5] Vince Jennings, Ralph Elliott, Richard Kennaway, and John Glauert, "Requirements for a signing avatar," in *Proceedings of the LREC2010 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Philippe Dreuw, Eleni Efthimiou, Thomas Hanke, Trevor Johnston, Gregorio Martínez Ruiz, and Adam Schembri, Eds., Valletta, Malta, May 2010, pp. 133–136, European Language Resources Association (ELRA).

[6] Ralph Elliott, F. Bueno, Richard Kennaway, and John Glauert, "Towards the integration of synthetic sl animation with avatars into corpus annotation tools," 01 2010.

[7] Alexis Heloir and Michael Kipp, "Real-time animation of interactive agents: Specification and realization," *Applied Artificial Intelligence*, vol. 24, no. 6, pp. 510–529, 2010.

[8] John Mcdonald, Rosalee Wolfe, Julie Hochgesang, Diana Jamrozik, Marie Stumbo, Larwan Berke, Melissa Bialek, and Farah Thomas, "An automated technique for real-time production of lifelike animations of american sign language," *Universal Access in the Information Society*, vol. 15, 05 2015.

[9] Ronan Johnson, "Towards enhanced visual clarity of sign language avatars through recreation of fine facial detail," *Machine Translation*, vol. 35, no. 3, pp. 431–445, sep 2021.

[10] Sylvie Gibet, Nicolas Courty, Kyle Duarte, and Thibaut Le Naour, "The signcom system for data-driven animation of interactive virtual signers: Methodology and evaluation," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 1, no. 1, pp. 1–23, 2011.

[11] J. P. Lewis, Matt Cordner, and Nickson Fong, "Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, USA, 2000, SIGGRAPH '00, p. 165–172, ACM Press/Addison-Wesley Publishing Co.

[12] Nico Galoppo, Miguel A. Otaduy, William Moss, Jason Sewall, Sean Curtis, and Ming C. Lin, "Controlling deformable material with dynamic morph targets," in *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2009, I3D '09, p. 39–47, Association for Computing Machinery.

[13] Lefan Wang, T. Meydan, and Paul Williams, "A two-axis goniometric sensor for tracking finger motion," *Sensors*, vol. 17, 04 2017.

[14] Paul Ekman and Wallace V Friesen, "Facial action coding system," *Environmental Psychology & Nonverbal Behavior*, 1978.

[15] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[16] Michaël Gilbert, Samuel Demarchi, and Isabel Urdapilleta, "Facshuman, a software program for creating experimental material by modeling 3d facial expressions," *Behavior Research Methods*, vol. 53, no. 5, pp. 2252–2272, 2021.

[17] MakeHuman Online Community, *MakeHuman*, MakeHuman Community, 2023.

[18] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17, 2017.

[19] Sergey Prokudin, Michael J Black, and Javier Romero, "Smplpix: Neural avatars from 3d human models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1810–1819.

Appendix: Submitted Papers

Intermediate Block Generation for Multi-Track Sign Language Synthesis

Paritosh Sharma
Michael Filhol
paritosh.sharma@universite-paris-saclay.fr
michael.filhol@cnrs.fr
LISN, CNRS
Orsay, France

Figure 1: Example results for the AZee expression presented. the first two poses are snapshots of *TOPIC* and *INFO*, while the following represent the intermediate pose and the evaluated motion curves

ABSTRACT

Generating realistic Sign Language using signing avatars is a challenging task that typically involves synthesis using either procedural or pre-animated techniques like motion capture or artistic editing of signs. However, combining these two approaches is difficult. In this work, we propose a novel method for generating intermediate poses in a multi-track representation of a sign language discourse. The proposed method uses procedural generation with artistic techniques to prioritize certain aspects of the generated poses while sacrificing others to improve the overall consistency of the representation. The system is implemented as an add-on in Blender, an open-source 3D toolkit.

CCS CONCEPTS

• **Procedural animation**; • **Motion processing**;

KEYWORDS

sign language, animation, motion retargeting

Conference acronym 'SCA, June 03–05, 2023, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Paritosh Sharma and Michael Filhol. 2023. Intermediate Block Generation for Multi-Track Sign Language Synthesis. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'SCA)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The field of simulating realistic Sign Language using signing avatars has been a topic of growing interest in recent years. Procedural generation techniques are commonly used on a skeletal representation[6] of the avatar to provide broad coverage of synthesized signs, while artistic and motion capture techniques are employed to achieve a more natural coverage of signs, albeit providing a smaller coverage. While both techniques have shown promising results individually, their combination requires careful tuning to produce accurate and realistic sign language generation.

Existing works for Sign Language synthesis use AZee[4] - a model which facilitates writing expressions to represent Sign Language discourses and determine the forms to be articulated on a multi-track timeline which automatically maps directly to a non-linear editor[9][2]. In this work, we propose a novel method to effectively generate intermediate blocks which contain interpolation information. To do this, we filter the most important components of the poses in the multi-track representation and generate intermediate poses which preserve these components while allowing for some deviation for naturalness. The algorithm is based on the use of artistically created motion templates. This allows us to generate motion curves that look more natural and accurately reflect the

intended meaning of the respective track while also preserving the original constraints.

2 OVERVIEW

Filhol et al.[5] proposed a fundamental guiding principle for animating AZee, according to which the coarser the basic animation blocks, the more natural the final animation. To understand this better, let's consider the following AZee expression representing signed utterance meaning "A cat is cute" or "Cats are cute".

```
:info-about
    'topic
    :cat
    'info
    :cute
```

Evaluating this expression with the AZee interpreter generates a recursive representation of blocks to be animated.

2.1 Block Generation

While generating the underlying blocks on the multi-track timeline, we match each block with a motion template. A block placed on the timeline can either be a *pre-animated block* with re-targeted animation data in our library(for this example, let's assume "cat") or a *minimally constrained block* with IK(Inverse Kinematics) and morph constraints to be satisfied on the avatar posture(example "cute").

Thus, figure 2 represents the corresponding block structure for the above expression. The blocks can be summarized as follows:

- **TOPIC**: Re-targeted sign for "cat".
- **INFO**: Contains more blocks with posture constraints for the sign "cute" such as HAND_PLACE_TRILL.INFO for placing the hand and trilling it on an axis and HAND_CONFIG.INFO for creating hand configuration.
- **HOLD1**: Constraints to hold the ending of the sign "cat".
- **HOLD2**: Constraints to hold the ending of the sign "cute".
- **BLINK**: Constraints for eye-blink.

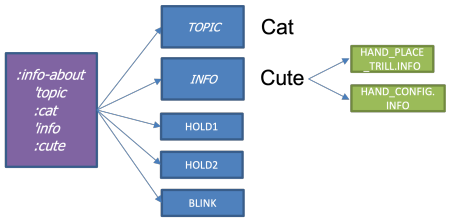


Figure 2: Blocks generated from the example AZee expression

Here, a coarser animation block such as **TOPIC** will generate more natural animation than **INFO** which is a composite block made up of **HAND_PLACE_TRILL.INFO** and **HAND_CONFIG.INFO**.

2.2 Intermediate Block Generation

After baking the minimally constrained and pre-animated blocks on the timeline, intermediate blocks are generated between pairs of blocks by mapping the f-curves to each other, as shown in figure 3. The resultant f-curves are evaluated based on the motion template.

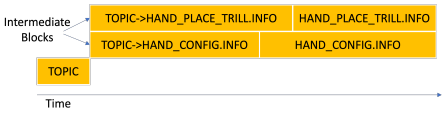


Figure 3: Intermediate blocks generated for blocks of *TOPIC* and *INFO*

2.3 Motion Template

A motion template contains information regarding how the f-curves in the intermediate blocks should be baked for pairs of blocks. Given a set of blocks $B = B_1, B_2, ..., B_n$ representing either poses or motions on a multi-track timeline, the motion template is used to generate f-curves inside the intermediate blocks that represent the interpolation between any two blocks B_i and B_j, where $1 \leq i < j \leq n$. This can be achieved using a function $f(B_i, B_j)$ that accepts two input blocks and generates an intermediate block between them. The function can be implemented using various interpolation techniques, such as linear, cubic, or spline interpolation. The specific choice of interpolation technique depends on the artist.

3 IMPLEMENTATION AND RESULTS

Our synthesis model is implemented in Blender[1] as an add-on. We also have a library of re-targeted signs for synthesizing our pre-animated blocks as well as skeletal and facial[3] morphs, which are used along with an IK solver for synthesizing the minimally constrained blocks.

We extend this synthesis model by adding motion templates and extend our algorithm by adding template checks for these motion templates. Figure 1 shows intermediate motion curves for the above example.

In future, we aim to improve our synthesis model by addressing the following areas:

- Improvements in the avatar model using SMPL avatar[8].
- Better AZee facial morphs using the FLAME facial model[7].
- Creation of motion templates from motion capture.

ACKNOWLEDGMENTS

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).

REFERENCES

[1] Blender Online Community. 2018. *Blender - a 3D modelling and rendering package.* Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

[2] Boris Dauriac, Annelies Braffort, and Elise Bertin-Lemée. 2022. Example-based Multilinear Sign Language Generation from a Hierarchical Representation. In *Proceedings of the 7th International Workshop on Sign Language Translation and Avatar Technology: The Junction of the Visual and the Textual: Challenges and Perspectives.* 21–28.

[3] Paul Ekman and Wallace V Friesen. 1978. Facial action coding system. *Environmental Psychology & Nonverbal Behavior* (1978).
[4] Michael Filhol, Mohamed Hadjadj, and Annick Choisier. 2014. Non-manual features: the right to indifference. In *International Conference on Language Resources and Evaluation*. Reykjavik, Iceland. https://hal.archives-ouvertes.fr/hal-01849040
[5] Michael Filhol, John Mcdonald, and Rosalee Wolfe. 2017. Synthesizing Sign Language by Connecting Linguistically Structured Descriptions to a Multi-track Animation System. 27–40. https://doi.org/10.1007/978-3-319-58703-5_3
[6] Michael Kipp, Alexis Heloir, and Quan Nguyen. 2011. Sign language avatars: Animation and comprehensibility. In *Intelligent Virtual Agents: 10th International Conference, IVA 2011, Reykjavik, Iceland, September 15-17, 2011. Proceedings 11*. Springer, 113–126.
[7] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (2017), 194:1–194:17. https://doi.org/10.1145/3130800.3130813
[8] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.
[9] Paritosh Sharma and Michael Filhol. 2022. Multi-Track Bottom-Up Synthesis from Non-Flattened AZee Scores. In *7th Workshop on Sign Language Translation and Avatar Technology: The Junction of the Visual & the Textual Challenges and Perspectives (SLTAT 7)*. Marseille, France. https://hal.archives-ouvertes.fr/hal-03721720

Received 12 April 2023; revised 12 March 2009; accepted 5 June 2009