A survey on the animation of signing avatars: From sign representation to utterance synthesis

Lucie Naert, Caroline Larboulette, Sylvie Gibet

Computers & Graphics (2020)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

A Survey on the Animation of Signing Avatars: From Sign Representation to Utterance Synthesis

Lucie Naert[a,*], Caroline Larboulette[a], Sylvie Gibet[a]

[a]*IRISA, Université Bretagne Sud, Vannes, 56000, France*

ARTICLE INFO

Article history:
Received February 11, 2020

Keywords: Signing Avatar, Motion Synthesis, Sign Representation, Utterance Synthesis, Procedural Animation, Data–Driven Animation, Sign Language

ABSTRACT

Signing avatars make it possible for deaf people to access information in their preferred language. However, sign language synthesis represents a challenge for the computer animation community as the motions generated must be realistic and have a precise semantic meaning. In this article, we distinguish the synthesis of isolated signs deprived of any contextual inflections from the generation of full sign language utterances. In both cases, the animation engine takes as input a representation of the synthesis objective to create the final animation. Because of their spatiotemporal characteristics, signs and utterances cannot be described by a sequential representation like phonetics in spoken languages. For this reason, linguistic and gestural studies have aimed to capture the typical and special features of signs and sign language syntax to promote different sign language representations. Those sign representations can then be used to produce an avatar animation thanks to sign synthesis techniques based on keyframes, procedural means or data-driven approaches. Novel utterances can also be generated using concatenative or articulatory techniques.

This article constitutes a survey of (i) the challenges specific to sign languages avatars, (ii) the sign representations developed in order to synthesize isolated signs, (iii) the possible sign synthesis approaches, (iv) the different utterance specifications, and (v) the challenges and animation techniques for generating sign language utterances.

Authors and affiliation

First author (corresponding author):
Name: Lucie NAERT
Affiliation: IRISA, Université Bretagne Sud, 56000 Vannes
E-mail: lucie.naert@univ-ubs.fr

Second author:
Name: Caroline LARBOULETTE
Affiliation: IRISA, Université Bretagne Sud, 56000 Vannes
E-mail: caroline.larboulette@univ-ubs.fr

Third author:
Name: Sylvie GIBET
Affiliation: IRISA, Université Bretagne Sud, 56000 Vannes
E-mail: sylvie.gibet@univ-ubs.fr

*Corresponding author:
e-mail: lucie.naert@univ-ubs.fr (Lucie Naert)

Computers & Graphics (2020)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

A Survey on the Animation of Signing Avatars: From Sign Representation to Utterance Synthesis

ARTICLE INFO

Article history:
Received September 2, 2020

Keywords: Signing Avatar, Motion Synthesis, Sign Representation, Utterance Synthesis, Procedural Animation, Data–Driven Animation, Sign Language

ABSTRACT

Signing avatars make it possible for deaf people to access information in their preferred language. However, sign language synthesis represents a challenge for the computer animation community as the motions generated must be realistic and have a precise semantic meaning. In this article, we distinguish the synthesis of isolated signs deprived of any contextual inflections from the generation of full sign language utterances. In both cases, the animation engine takes as input a representation of the synthesis objective to create the final animation. Because of their spatiotemporal characteristics, signs and utterances cannot be described by a sequential representation like phonetics in spoken languages. For this reason, linguistic and gestural studies have aimed to capture the typical and special features of signs and sign language syntax to promote different sign language representations. Those sign representations can then be used to produce an avatar animation thanks to sign synthesis techniques based on keyframes, procedural means or data-driven approaches. Novel utterances can also be generated using concatenative or articulatory techniques.

This article constitutes a survey of (i) the challenges specific to sign languages avatars, (ii) the sign representations developed in order to synthesize isolated signs, (iii) the possible sign synthesis approaches, (iv) the different utterance specifications, and (v) the challenges and animation techniques for generating sign language utterances.

1. Introduction

Sign Languages (SL) are the primary means of communication for deaf people all around the world. They are natural languages with their own syntax and set of rules that drastically differ from oral languages.

Embodied conversational agents called *sign language avatars* or *signing avatars* are a promising way to present information to deaf people in their preferred language while preserving the anonymity of the signer. Avatars are also more flexible and interactive than 2D-video recordings of signers. Indeed, editing sign language content is simplified using avatar animations which can be customized to suit the needs of their users, for example by slowing down the stream of signs, changing their viewpoint, or even adapting the appearance of the agent. The great configurability of avatars allows them to be used in various applications, such as interfaces for bilingual dictionaries or for translators between a specific sign language and a specific oral language, or recreational learning instances.

Depending on the final application, the approaches to generate sign language animations will differ. The creation of an online bilingual dictionary will require the synthesis of isolated

signs in their citation form, i.e. not inflected by a sentence context. On the contrary, an application of translation into sign language will require a mastery of grammatical rules and the use of contextualized and coarticulated signs to generate correct utterances. In both cases, the isolated signs, like the utterances, must be as precise and natural as possible. This survey aims at presenting the different steps and various methods that are currently used to animate SL avatars whether at a sign or at an utterance level. This survey does not to cover machine translation issues: techniques for translating any oral language into any sign language are beyond the scope of this paper. Moreover, we focus on the animation of the manual features; a complementary survey, dedicated to the animation of facial expressions for sign language avatars, was proposed by Kacorri [1].

The remainder of the article is organized as follows: general information on sign languages and on avatars motion are given in Section 2. The scientific issues and challenges of signing avatar animation are detailed in Section 3. The process of isolated sign synthesis is described in Section 4. Utterance synthesis is presented in Section 5. Section 6 presents existing work on signing avatars. Finally, Section 7 compares and discusses the presented techniques.

2. General Information

2.1. Sign Languages

Sign languages are visual-gestural languages used by an important part of the deaf population. While oral languages mainly rely on the voice to convey a message and on the audio channel to receive it, sign languages use different sensory channels. The movements of the whole body and facial expressions are used to produce a message which is interpreted by the interlocutors via their visual channels.

Sign languages are languages in their own right that have been developed naturally over time by Deaf[1] communities to communicate. They have their own vocabulary, called *signary*, and precise grammatical rules. They are rich languages that, thanks to their visual and gestural aspects, can take advantage of the possibilities of space to tell stories, communicate information, describe situations with precision, date temporal events, make poetry, jokes, etc.

There is not just one universal sign language that would allow all deaf people around the world to communicate. Sign languages were born naturally from the need to communicate of the Deaf themselves [3]. As a result, there are as many sign languages as there are Deaf communities. Even if many countries present a single official sign language (e.g., French Sign Language, Brazilian Sign Language), there is no direct correspondence between countries and sign languages: several sign languages can be present in the same country and one sign language can extend beyond the borders of a country (e.g., Indo-Pakistani Sign Language). And, like with oral languages, there are regional variants. Moreover, while some sign languages like British Sign Language and New Zealand Sign Language can be considered as dialects of a specific sign language (BANZSL: British, Australian, and New Zealand Sign Language) as they share a common syntax and have a lot of lexical overlaps, others are mutually unintelligible such as, surprisingly, British Sign Language and American Sign Language [4]. Sign languages can differ in their signary (e.g., the sign for [DOG] in British and French Sign Language are very different: in British Sign Language, the signer uses both hands following the movement of the dog's front legs while, in French Sign Language, only one hand is used and is more akin to the movement of a dog's tail), in their grammar (e.g., some examples of the differences in the interrogative constructions between 35 different SL are described in [5]), or in the nature and frequency of use of their fingerspelling alphabet (British Sign Language relies on a two-handed alphabet while French Sign Language has a one-handed alphabet). Finally, an international signary exists called International Sign (IS) that can be used by deaf people who lack a common sign language.

2.2. Movement of Avatars

In $3D$ traditional animation, an avatar is represented by a complex $3D$ mesh in the shape of a virtual humanoid. It can

[1] We distinguish here the "deaf" and "Deaf" spellings: lower-case "deaf" refers to the pathological aspects of deafness while the capitalized "Deaf" designates its cultural dimension [2].

be animated thanks to a *skeleton* which is a tree structure composed of rigid segments (*bones*) connected by *joints*. A *pose* or *posture* is the state of the skeleton at a given time or frame, described by the position and orientation of each joint. The pose $X(i)$ of the skeleton at frame i is therefore defined as:

$$X(i) = \{(pos(i,1), orient(i,1)), orient(i,2), ..., orient(i,n)\} \quad (1)$$

where n is the total number of joints in the skeleton, $(pos(i,1), orient(i,1))$ is the absolute position and orientation of the root joint at frame i and $orient(i,j)$ with $j \geq 2$ is the relative orientation of the joint j at frame i.

A *motion* M is a sequence of k poses: $M = \{X(1), X(2), ..., X(k)\}$. Its duration is equal to $k * \Delta t$ where Δt is the timestep between two poses.

Forward and Inverse Kinematics (FK and IK resp.) are the main techniques used to synthesize the poses of the avatar. FK consists in computing a skeleton pose from the explicit specification of the position or orientation of all the joints. IK, the inverse problem, traditionally consists in computing the orientations of the joints of an articulated chain in order for some joints (often its end-effectors) to reach specific positions or orientations. Different kinds of constraints, among which joint limits, may be added to the system to eliminate physiologically impossible solutions. In order to have a constant Δt, in-between poses for a given frequency can be interpolated.

Standard file formats, like *BVH* or *FBX*, are designed to record motions. Generally, the skeleton's hierarchy is specified and the corresponding sequence of poses is stored in the form of a sequence of numerical values. The *FBX* format can, in addition, store the $3D$ mesh of the avatar as well as other features.

This paper focuses on the skeleton motion synthesis in the case of sign language generation. The skinning and rendering steps, which consist in computing the deformation of the $3D$ model and in displaying the resulting animation are beyond the scope of this paper.

3. Linguistic Background and Challenges

A sign is a lexical unit of sign languages as is a word to oral languages. An utterance is close to the concept of "sentence" in oral languages: it is composed of signs, performed sequentially or at the same time (*co-occurring* signs), and presents the statement of an idea.

Sign and utterance synthesis are two inherently different processes as sign synthesis is the process of generating the skeletal animation of an isolated sign while utterance synthesis consists in the animation of a whole sign language sentence and therefore requires a more extensive knowledge of sign language linguistics and involves different mechanisms than the generation of isolated signs. Sign language grammar must be taken into account in order to produce a consistent sign language utterance.

In this section, two linguistic approaches are presented. They integrate the main linguistic mechanisms of SL and can be used to animate signing avatars. The first one, the parametric approach, applies to a phonological level and is mainly used for the synthesis of isolated signs. The second one characterizes the sign inflection mechanisms. It addresses the more variable aspects of the language at a lexical level and should be particularly taken into account during utterance synthesis.

3.1. The Parametric Approach for Isolated Sign Synthesis

3.1.1. Sign Language Phonology

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

Moreover, the existence of a phonological system leads to the validation of the *double articulation* principle [6]. This principle claims that human languages can be segmented on two levels: a first level linking an element with a meaning (in oral lan-

guages, a word is the element of minimal size for the first level) and a second level composed of distinctive units without meanings (the phonemes). Here again, SL phonology constitutes a way to create a bridge between oral and sign language linguistics. In the case of SL, the parametric approach states that a sign is a sequence of discrete values taken by SL phonological components. These components often refer to five elements, three of which (the *hand configuration*, the *hand placement* and the *hand motion*) have been described in 1960 by Stokoe [7], one, the *hand orientation*, was specified in 1978 in the work of Battison [8] while the *non manual features* were later added to the definition:

1 - Hand configuration corresponds to the overall shape of the hand characterized by the disposition of the fingers. Each configuration corresponds to a discriminating and meaningful posture of the hand (see Fig. 1). Researchers do not agree on the number and nature of hand configurations. For French Sign Language (LSF), Cuxac lists 39 configurations [9][2] while Boutora identifies 77 configurations [10] and Millet 41 [3].

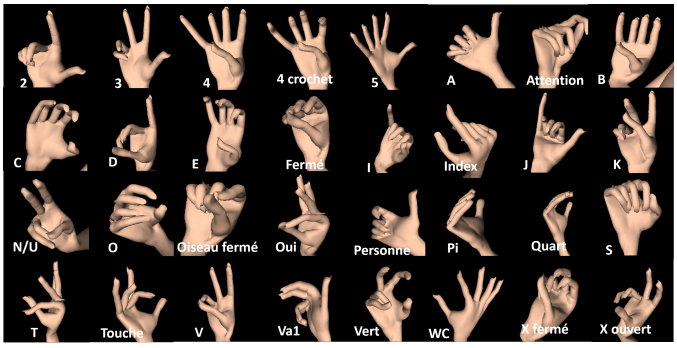

Fig. 1. Hand configurations: the 32 configurations of the *Sign3D* project [11].

2 - Hand placement is the location of the hand in the signing space or on the body of the signer. Depending on the field of study, it is defined differently. In linguistic studies, hand placement, also called *anchoring* [3, 12], is often the global area where the sign is produced (neutral space in front of the signer, eyes, hand palm, etc.) in its *citation form* [13, 14, 15] (also called "uninflected form" [16], i.e. deprived of any syntactic context). For the computer animation community, it can either designate the precise Cartesian coordinates, or the discrete area where the hand is positioned at a precise time. When Cartesian coordinates are used, the hand placement often changes during the realization of a sign. When discrete areas are used, the hand placement may vary depending on the sign and the granularity of the partition of the signing space (see Fig. 2).

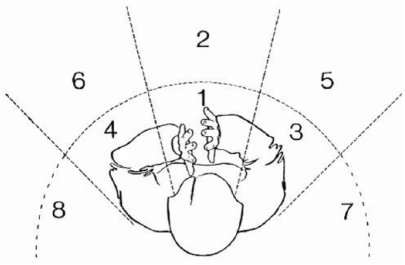

Fig. 2. Hand placement: example of a discretization of the signing space (extracted from [17]).

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

4 - Hand orientation is defined by the direction of the hand palm and of the palm normal (see Fig. 3). It is strongly constrained by the hand movement and the human physiological limits. However some minimal pairs distinguished only by the orientation exists (e.g., in LSF, [MAISON] *(house)* and [DEMANDER] *(to ask)* are such a minimal pair; their configuration, movement and placement are similar while the hand orientations are different: in [MAISON], the direction of the palms is upwards while in [DEMANDER], the palms are oriented towards the interlocutor in front of the signer.).

Fig. 3. Hand Orientation: definition of the axes.

5 - Non-manual features (NMFs) include the facial expressions, the mouthing, the gaze and torso direction. NMFs and,

[2]His list identifies up to 41 hand configuration counting the hand configuration alternatives. He specifies that it is not a "closed inventory".

particularly, the facial expressions, have many functions in SL: (i) they are often considered (but not always [3]) as a phonological component ([GAGNER] *(win)* and [DOMMAGE] *(too bad)* in LSF are a minimal pair only distinguished by the facial expression: the first is done with a happy expression while the second is done with a sad expression), (ii) they can also express the affect (surprise, fear or anger hues can be added to the message with the adequate facial expression), or (iii) have syntactic roles (e.g., a raise of the eyebrow can indicate a question while a swelling of the cheek can add information about the width of an object). Those functions can co-occur in signs and utterances and must be managed in synthesis systems [18].

A sign is therefore a sequence of values taken in parallel by each of these components in finite sets (hand configuration, hand placement) or infinite sets (hand movement, hand orientation). However, while in vocal languages, words are formed with a simple sequence of phonemes, in sign languages, components take on values simultaneously in addition to having a sequential aspect. To designate this particularity, we refer to sign languages as *multilinear* languages [19].

3.1.2. Challenges for Sign Synthesis

The construction of signs following the parametric approach consists in creating signs from scratch by combining, on one virtual character, the phonological parameters whose values have been fixed. In this case, we consider that a sign is the exclusive result of the values taken by its parameters.

This approach mainly aims at creating signs with relatively stable parameters. We are therefore more interested in signs in their citation form, i.e. without a syntactic context. This context can be added in a second step, as described in Section 3.2.

The parametric approach is beneficial for movement synthesis and signing avatar animation because the decomposition of language into atomic elements allows each component to be treated independently before synchronizing the different channels, and this theory offers a way to represent a language production in the form of a sequence of targets to be reached.

However, the implementation of a phonological recombination system raises some challenges:

Definition of a corporal mapping between the phonological elements and the set of joints (i.e. the *channel*) of the model to be animated. Indeed, if we want to assign the value taken by a phonological parameter to an avatar, it is necessary to define a mapping between the parameter in question (e.g., the hand configuration) and the articulations affected by this parameter (e.g., the fingers joints).

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

Intra-channel coarticulation. During the realization of a sign, the change from one parameter value to another (e.g., a hand configuration change) must be managed in order to have a smooth and realistic motion.

Synchronization of the body channels. To obtain the desired meaning, the channels must be synchronized precisely. On a given channel, the determination of the precise timing to reach the different values is important. In addition, relative synchronization between the different channels is also essential to obtain both a semantically correct sign and a realistic movement.

3.2. Sign Inflections for Utterance Synthesis

When synthesizing utterances, the form of some signs will vary to take the context into consideration. This is called the *sign inflection*. Two types of inflections are distinguished: inflections due to the illustrative nature of SL often referred to as iconic mechanisms (Section 3.2.1) and inflections using spatial referencing (Section 3.2.2).

3.2.1. Iconic Mechanisms

Sign language iconicity refers to the similarity between the sign and what it designates (resp. the *signifier* and *signified* of Saussure [20]).

In his work, Cuxac [9] states that LSF[3] has two modes of production: (i) a non-illustrative one based on signs in their citation form (called *standard signs* by Cuxac) whose gestural execution is relatively invariant and (ii) an illustrative one, called *structure of great iconicity*, using different spatiotemporal mechanisms to describe a scene, an object or an animal.

We describe hereinafter three illustrative mechanisms of SL:

Size and Shape Specifiers consists in using the hand configuration, wrist orientation and amplitude of motion to describe the shape and size of an object. For example, [TO GIVE] will not be performed with the same configuration and orientation in the sentence "I give you a book" (configuration of the *'duck's beak'* representing thick flat objects) and in "I give you a glass" (configuration of the *'C'* representing cylindrical objects) (see Fig. 5). In addition, the amplitude of the motion is often used as a size specifier, as shown on Fig. 4.

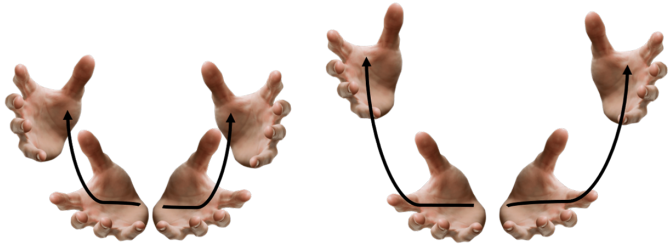

Fig. 4. Iconicity on the LSF sign *[BOL]* (bowl): the size of the bowl corresponds to the amplitude of the motion.

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

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

3.2.2. Spatial Referencing

The placement of entities inside a scene, their referencing, or the creation of interactions between those entities can be achieved through the variation of the hand placement or of the motion trajectory. We present below two of such spatial referencing inflections:

In **indicating verbs**, the trajectory or *motion path*, of some signs can change according to the relation between the described entities. The hand movement corresponding to the verb [TO GIVE] in the sentence "I give him" will not be performed in the same direction as the same verb [TO GIVE] in the sentence "you give me" (see Fig. 5).

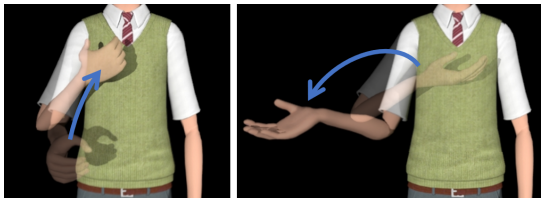

Fig. 5. Indicating verbs. Hand movements and configurations for the sentences "you give me a glass" (left) and "I give him a book" (right).

Pointing gestures consist in using the hand (often, the tip of the index) to designate an entity or a location. It can be used to indicate the subject(s) or object(s) of an action ([I], [YOU], [THIS ONE]) or to associate virtual objects to $3D$ locations in the signing space to give a relative placement of one object with respect to the other (in the context of a description for example) or for a future referencing of these objects [16].

[3] Cuxac research on LSF can be applied to many SL.

These inflected structures, whether iconic or using spatial referencing, are consistently present in SL discourses and are essential in situations of scene description or storytelling.

3.2.3. Challenges for Utterance Synthesis

To highlight the difference between sign and utterance synthesis, we can do a parallel with oral languages. In many languages, words have a different form when they are isolated and uncontextualised than when they are used in a sentence: verbs are conjugated and some words are put to the singular/plural form, for instance. Moreover, when pronouncing words, speakers make a *liaison* between the end of some words and the beginning of the following. More generally speaking, the pronunciation of a word will slightly impact the pronunciation of the neighbouring words: it is the coarticulation mechanism. The same mechanisms exist for sign languages.

Isolated sign synthesis mainly aims at building signs for bilingual dictionaries or educational applications. The signs are uncontextualised and need just to be in their citation form. Utterance synthesis is mainly used for machine translation systems to accurately express an oral language sentence in a given sign language (e.g., between German and Swiss German Sign Language [21] or between English and Irish Sign Language [22]) and for storytelling. Whether for translation or storytelling, the form of some signs is influenced by the context: both signs in their citation form and inflected signs should be used. Indeed, the signs in their citation form are not enough for utterance synthesis: a simple concatenation of those signs to create an utterance does not do justice to the richness of SL and can generate incorrect utterances. The targets that must be reached in space will vary with respect to the context even if the signs have similar gloss denomination. The multiple variations of illustrative signs must be taken into account just as well as the fixed citation form of signs.

Many challenges of the synthesis of inflection mechanisms are specific to the mechanisms involved. However, we list here some issues that are common to all iconic mechanisms.

Context awareness: the context of the signs must be taken into account to build inflected signs. For instance, as mentioned above, indicating verbs take a different form when put into an utterance. Another example are proforms or classifier predicates which take advantage of the hand configuration and movement to embody an object or a person. They are used to describe an infinite number of situations and are not suited for isolated sign synthesis but are very interesting in utterance synthesis.

Space representation: the signation space is an area in front of and around the signer in which he/she can place the entities of his speech. In this space physically limited by the signer's reaching capabilities, he/she will be able to describe an infinite and constantly changing space. Interestingly, space will also be the carrier of temporal information: a movement of the torso backwards or forwards makes it possible to place events in the past or the future. When doing motion synthesis, the appropriation of the signing space requires the definition and naming of $3D$ areas around the signer. The creation of sign language content for avatar animation requires this discretization of the signing space.

Coarticulation: in SL, coarticulation effects are characterized by the influence of one sign on the adjacent signs [23]. Coarticulation is expressed both in the transitions between the signs and in the inflection of each sign to take into account its previous and following signs. To have a more natural sign language flow, the transition motion between signs must be generated carefully [24]. As coarticulation results in a smoother motion with some spatial targets not reached, some synthesis systems loosens their trajectory constraints to obtain more realistic motions by taking coarticulation into account [25].

4. Isolated Sign Synthesis

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

In the field of sign synthesis, work on the synthesis of signs in their citation form is more common than work on the synthesis of inflected signs for several reasons: signs in their citation

form exist in a limited number and are listed in dictionaries, their description in a notation system often already exists (e.g., *HamNoSys* [26], see Section 4.1) and they have an almost direct equivalent with words in oral languages. These signs can be defined by a set of fixed values taken by phonological components of SL following the parametric approach.

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

We first present, in Section 4.1, visual representations and parametric notations of signs that can be used to manually specify keyframes. Then, in Section 4.2, we survey the different computer-friendly representations of signs that are directly used in procedural animation and, potentially, in data-driven processes. All the different sign representations are compared in Table 1. Finally, Section 4.3 presents the existing synthesis techniques.

4.1. Linguistic Representation of Signs

In order to synthesize a sign, a representation of the synthesis objective is needed. For isolated signs which are mainly synthesized following a parametric approach, the representation should highlight the structure and, possibly, the values taken by the phonological elements during the sign production. The spatiotemporal aspect of sign languages makes exhaustive representation of signs a complex problem involving researchers both in the linguistic and in the computer animation fields. We present here the representations mainly used by the linguistic communities to study signs.

4.1.1. Visual Representations

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

Signs are motions specified both in the $3D$ space and in time: in a **drawing**, the use of arrows and of different types of contour (e.g., dotted line or fine line) allow for a partial representation of those dimensions on a $2D$ paper. However, this schematic representation depends on the interpretation of the user and on the skill of the artist. Drawings are an ambiguous representation that often needs to be clarified with annotations. Fig. 6 is extracted from a French Sign Language (LSF) textbook. It shows a drawn representation of the sign [HELLO] complemented with annotations about the hand motion, the facial expression and the hand configuration. As a consequence of this ambiguity and of the impractical aspects of $2D$ drawing for computerization, this type of representation cannot be directly used as a computer formalization for animating an avatar.

Fig. 6. The sign for [HELLO] in LSF. To remove any ambiguity of the 2D drawing, some annotations are added to describe the motion, the facial expression and the hand configuration.

Videos recordings are another, more precise and exact representation of signs that is very popular in the linguistic community to store and study SL signs and utterances[4]. Videos do not allow for signer anonymity but are very efficient to record the dynamics of signs. Nevertheless, a video recording alone eliminates the depth information and imposes a point of view on his viewer. Like drawings, it lacks flexibility and its format is not suited for automatic computer synthesis.

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

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

4.1.2. Gloss Representation

Another straightforward way to describe sign language is to use a gloss representation or "glossing". Glossing consists in associating one or more words of an oral language to a sign. It is used by linguists to annotate SL videos. No gloss standard exists (which is a recurrent impediment to obtain consistent annotated corpora [27]) but, as a convention for this paper, we will designate glosses using brackets and uppercase letters (i.e. [GLOSS]). A sequence of glosses is not a translation but the oral language description of a sign language utterance: for example, a sentence glossed as "[YOU][LIVE][WHERE?]" will be translated as "Where do you live?". Furthermore, a sign may not have a one-word equivalent in an oral language. In this case, the corresponding gloss can contain more than one word (e.g., [TO GIVE] may be used in different contexts, such as in the gloss description: "[TO-GIVE-A-PAPER]" or "[TO-GIVE-A-GLASS]".

The way to execute a sign is not indicated in the gloss description. Therefore, few animation systems can choose to rely exclusively on a glossing description of the desired SL production since it implies the presence of a motion database annotated on the same gloss-level as the specification like in [28] for motion capture data or in [29] for hand-crafted animation[5]. As a consequence, isolated sign animation systems need another, lower-level sign representation to achieve the actual motion synthesis.

4.1.3. Parametric Notation and Writing Systems

The most obvious way to transcribe a language is writing. For oral languages, alphabets or syllabaries are common ways to describe the sounds that can be produced in a spoken utterance. Each character is assigned a sound and the concatenation of characters forms new sounds that result in words and sentences when spoken out loud. The transcription of an entire language is possible using a finite number of letters (using the native writing system or other transcription alphabet such as the Pīnyīn notation for Chinese language).

[5]Glosses can however be used to sequentially specify a full utterance (see more detail in Section 5.1.1).

The conception of a similar transcription system for signed languages is the focus of many studies. The written representation of the linguistic production is called a *notation system*. As mentioned in Section 3.1, signs can be decomposed into linguistic components that can be seen as phonemes (the work of Friedman is an example of a phonological analysis of American Sign Language [30]). Parametric notations propose to decompose signs into a combination of those components and to assign a value for each one from a finite set of possible values. However, due to the spatiotemporal aspect of the language, the assignment of a finite number of characters to the description of the whole set of possibilities of the language is not an easy task. The parametric notations that are presented in this section aim at discretizing the continuous concepts that are space and time in order to find the optimal transcription of sign language. Some questions are often raised, such as the partitioning of the signing space, the number of possible hand configurations, or the way to represent kinematic behaviours (acceleration, deceleration, etc.).

In 1825, Auguste Bebian, looking for a notation system for sign language, is the first to define the signs as a combination of elementary components [31]. However, due to "The Milan Conference" of 1880, which promoted the oral education of deaf people in Europe at the expense of sign languages, research concerning sign language writing was stopped until the important work of Stokoe in 1960. Stokoe [7] defined three linguistic components: *hand configuration*, *hand placement* and *hand motion* (see Section 3.1). His work on a sign language transcription system resulted in the **Stokoe's notation** [32] which describes the ASL signs using a combination of those three components: the sign location is called *tabula* or *TAB*, the hand configuration is *designator* or *DEZ* and the hand motion is *signation* or *SIG*. Each component is specified using a limited set of symbols. *Hand orientation* and *non-manual features* (NMF) (mainly facial expressions) were later added to complete the definition [8] but only hand orientation was incorporated to the notation. The notation merely describes the signs without tak-

ing into account the intent of the signer: even if the two hands perform a symmetric gesture, the hand configuration of each hand will be described (in Fig. 7, we can see the symbols B˖ repeated twice, one for each hand).

Later, the **Hamburg Notation System** or **HamNoSys** [26] was developed to palliate those limitations. It includes the four components of the *Stokoe's notation* and some facial expressions. The variety of possible values for each of the components makes *HamNoSys* a much more complete notation system. It was designed to be language-independent and extensible to new linguistic mechanisms. It can better handle some SL mechanisms like symmetry (the symbol ¨ at the beginning of the transcription in Fig. 7) and can be transcribed in a linear way using computer Unicode symbols. However, it is still limited in terms of non-manual features.

SignWriting [33] is based on a dancing gestures transcription and constitutes a more graphical and intuitive notation of SL. It integrates some facial expressions, body movements and even iconic signs (see Fig. 7). Hand configurations and orientations can be defined using a limited set of symbols but the placement of the hands with respect to the body is indicated only in a relative manner thus leaving some ambiguities. It differs from the two previous notations in the sense that it was designed to be used by deaf people as a writing system and not by linguists or researchers as a notation system. It is SL independent and is taught in deaf classes around the world [34]. Other writing systems exist for different sign languages (like "SEL" for Brazilian Sign Language [35]) but none as popular as *SignWriting*.

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

The temporal aspects of SL were thoroughly studied in the work of Johnson and Liddell. In 1989, they introduced the **Movement/Hold model** [36] where signs were defined as an alternation of static poses (*Hold*) and dynamic transitions (*Movement*) between two consecutive poses. Then, between 2010 and 2012, they defined the **Sign Language Phonetic Annotation**

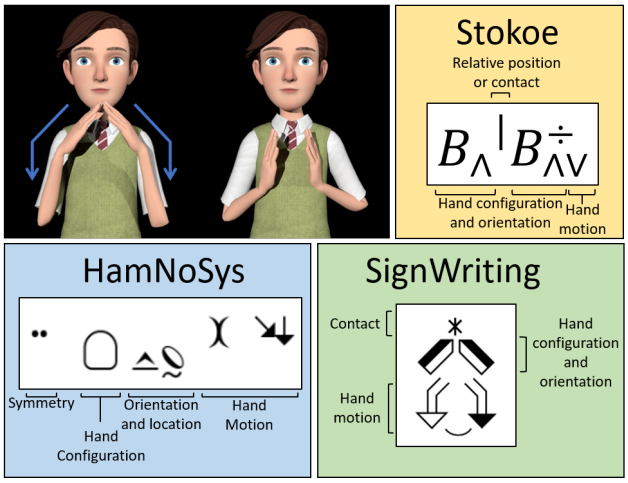

Fig. 7. The sign for [HOUSE] in American Sign Language using the *Stokoe*, *HamNoSys* and *SignWriting* notation systems. The sign is shown on the avatar on the top-left corner.

(SLPA) [37] which relies on the Posture-Detention-Transition-Shift (PDTS) classification. This classification uses two timing features to distinguish between the four classes, called *segments* or *timing units*. The first timing feature is the *static/dynamic* nature of the segment. In static segments, one or more SL component(s) (hand configuration HC, orientation FA, placement PL, non-manual features NM) is stable during a finite amount of time while *dynamic* segments are transitions from one static segment to the following. The second timing feature is the *transient/deliberate* quality of the motion during the segment. This feature mainly impacts the duration of the segment. A deliberate segment will have a significantly higher duration than a transient segment. More precisely, the four types of timing units are the *Posture* (static and transient), the *Detention* (static and deliberate), the *Transform* (dynamic and transient) and the *Shift* (dynamic and deliberate) segments. The *SLPA* has the particularity of transcribing a sign using a table: the timing units are represented in the columns whereas the articulatory components are described on the lines. An "∞" symbol designates a change in the articulatory features. Fig. 8 shows the *SLPA* transcription of the word [CHICAGO] in ASL.

4.1.4. Phonetic Codings of Hand Configurations

To describe the hand configuration, the phonetic and the phonological systems must be distinguished. In phonological systems such as *Stokoe's notation*, *SignWriting* or *HamNoSys*,

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

Fig. 8. The sign for [CHICAGO] in American Sign Language using the SLPA notation system (table based on [37]). The sign [CHICAGO] draws a "7" in the signing space with a *'C'* hand configuration. The three placements PL_i correspond to the three inflection points of the "7".

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

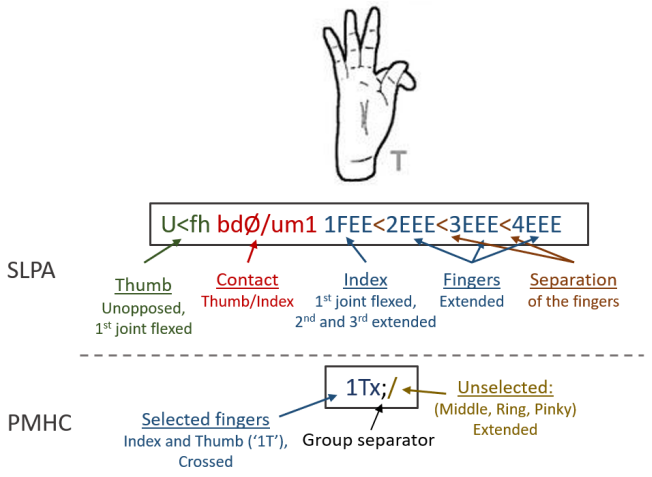

Fig. 9. The coding of the 'T' hand configurations with the *SLPA* and the *PMHC* phonetic codings.

In addition to the temporal aspects, the **SLPA** introduces a phonetic coding for the hand configurations [38, 39]. Indeed, in Fig. 8, *HC1* represents a hand configuration that stays stable during the execution of the sign but it gives no indication on the nature of the hand configuration. The *SLPA* coding of the hand configuration was defined in order to precisely describe those hand configurations. The position of each finger is described by indicating if each joint (3 joints of each finger + 2 joints of the thumb) is flexed (*f* or *F* with respect to the intensity of the flexion) or extended (*e* or *E*). The position of the fingers with respect to each other are noted with a particular symbol (e.g., "=" if the fingers touch each other, "<" if they are separated). The description of the thumb position with respect to the other fingers is placed at the beginning of the specification (*U* for "unopposed", *O* for "opposed" and *L* for "lateral"). Additional notations describe the type of contact between the fingers. The *SLPA* phonetic coding is thus exhaustive but its exhaustiveness makes it a complex system, hard to use in practice. A more user-friendly version of the *SLPA* could be defined by reducing the number of degrees of freedom and introducing anatomical knowledge to eliminate anatomically impossible hand configurations [40].

Another phonetic coding of the hand configurations, the **Prosodic Model Handshape Coding (PMHC)**, was proposed by Eccarius and Brentari [41]. Each configuration is coded based on (i) the detection of sets of *selected fingers* – groups of fingers that are the most relevant for the configuration and that share a state (or *joint configuration*) – more than one group can be identified (separated by ";"), (ii) the determination of the state of those sets of fingers (e.g., curved "@", bent "[", crossed "x" or extended by default), (iii) the determination of the state of the non-relevant fingers (extended "/" or flexed "#") and, (iv) the specification of the thumb position. It uses the standard ASCII characters to represent the hand features and different sign languages were studied to design the system making it a generic and extendable system. Fig. 9 shows the coding of two hand configurations with the *SLPA* and the *PMHC*. *PMHC* is more compact than *SLPA* but it lacks its precision.

While the visual representations can be taken advantage of by graphic designers to manually define the key poses of a skeleton, the notation and writing systems are rarely used as such by the computer animation community. They are useful systems for SL linguistic analyses but lack the precision needed to synthesize motions. Phonetic codings of hand configurations, also defined in the linguistic fields, provide detailed descriptions of

the SL hand configurations and could potentially be integrated in a specification language. In order to animate SL avatars, SL notation systems have therefore been used as a basis to design SL specification languages and scripts that can directly be exploited to create signs automatically.

4.2. Scripting Languages for Sign Representation

Scripting languages have been designed to specify the signs in a way directly understandable by a computer. Those languages have been developed by the computer animation community but are often based on linguistic sign representations (see Table 1 for a comparison of the different sign language representations).

4.2.1. Descriptive Languages Based on Existing Notation Systems

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

The **SignWriting Markup Language (SWML)** [46] is an XML version of *SignWriting* and has been designed to allow storage and processing of sign language files. The structure corresponding to a sign, called a *signbox*, contains the exact same information as the *SignWriting* transcription of the sign. Consequently, it does not overcome the limitations of *SignWriting*, namely the ambiguities of the hand placement and the time management.

Similarly, **SiGML** was initially an XML version of the *HamNoSys* transcription system [47]. This language was developed within the European projects *ViSiCAST* and *eSIGN* that promote Deaf access to information [48]. Contrary to *SWML*, it was designed with the prospect of animating virtual signers, this is why some aspects of SL, like timing or very precise orientations, can be specified in *SiGML* and not with *HamNoSys*. Moreover, the PDTS classification of Johnson & Liddell was later added to *SiGML* in an extended version of the language [42] additionally providing an explicit timing control, synchronization between elementary motions and a direction specification in various contexts. It is one of the most advanced existing sign language specification for SL synthesis.

A parametric description of signs based on an XML version of the *Movement/Hold model* of Johnson & Liddell studies was also designed by Amaral et al. [49] in order to animate an avatar using Brazilian Sign Language to translate textbooks for educational purposes [50].

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

4.2.2. Programming Languages for Sign Synthesis

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

In an early work, Lebourque et al. [51] defined **QualGest**, a high-level specification language dedicated to LSF that takes into account the four manual parameters (hand configuration, placement, motion and orientation), called *gestems*.
To specify the hand placement, they use a discretization of the signed space, including the definition of: (i) a set of directions (defined from the three main planes – sagittal, frontal and horizontal–, plus two intermediate planes), (ii) of amplitudes (proximal, medial, distal and extended), and, (iii) of body positions. Movements are defined using a finite set of predefined primitives (pointing, straight-line, curve, ellipse, wave, or zigzag), parameterized by a set of starting, ending and, if needed, intermediary locations. Hand configurations are defined, using 5 basic hand configurations (angle, hook, spread, fist, stretched), completed by *modifiers*. Hand orientation can be specified in a relative or absolute manner from two hand directions (palm and metacarpus). All the different values taken by the parameters can be specified by meaningful terms

Table 1. Comparison of the sign representation.

Category	Name	Fidelity	Temporal aspects, synchronization	Non manual features	Flexibility	Understandable by a computer
Visual Representation	Drawings	✓	✗	✓	✗	✗
	Video recordings	✓✓	✓✓	✓✓	✗	✗
Parametric Notation	Stokoe [32]	(✓)	✗	✗	(✓)	✗
	HamNoSys [26]	✓	(✓)	(✓)	(✓)	(✓)
	SLPA [37, 39]	✓	✓	✗	(✓)	(✓)
	SignWriting [33]	✓	(✓)	(✓)	(✓)	✗
Scripting Language	SiGML (extended) [42]	✓	✓	(✓)	(✓)	✓✓
	QualGest [25]	✓	✓	✗	✓	✓✓
	Losson [43]	✓	✓	(✓)	✓	✓✓
	Zebedee [44]	✓	✓	✗	✓✓	✓✓
	EMBRScript [45]	✓	✓	✓	(✓)	✓✓

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

or numerical values. Even if non-manual channels are not taken into account in this description, other interesting information about the symmetry of the two arms, the synchronization between the dominated/non-dominated arm or the number of repetitions of elementary gestures can be added, thus indicating the intent of the signer. Moreover, coarticulation mechanisms can be added to the synthesized motion.

Similarly, Losson defines a sign description where signs are divided into atomic gestures called *shifts* [43]. Again, the four Stokoe's parameters are used to determine the characteristics of the *shifts*: initial and final configurations of the hand, orientation and placement, and nature of the movement. Hand motions are specified using displacement primitives (straight line, arc or circle) and the location of the targeted destination of the hand (plus the equation of a plane in which the displacement takes place for arc and circle trajectories). The primitives can be augmented with secondary movements, contact zones or modifiers. To describe the hand configurations, the behavior of the thumb is considered separately from the other fingers (similarly to the hand configuration coding of *SLPA*). Repetitions in the movement, synchronicity properties of the hands, symmetry or anti-symmetry characteristics, or relative placement of the hands can be specified. In addition, Losson's representation uses a parametric computer language that allow the specification of sign inflection mechanisms (size and shape specifiers, spatial referencing). Fig. 10 shows an example of Losson's specification for the sign [CHIMNEY] in LSF using two *shifts*.

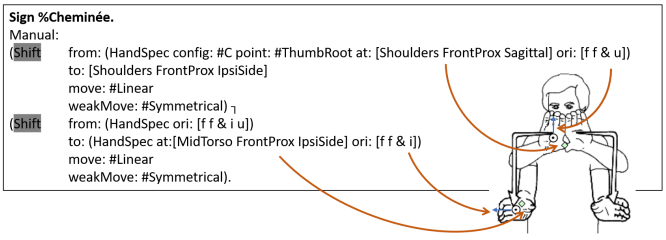

Fig. 10. Example of Losson's description for the sign [CHIMNEY] in LSF (avatar and code extracted from [43]).

In *QualGest* and Losson's approaches, the time does not explicitly appear in the SL specifications, whereas the **EMBRScript** is based on the explicit specification of key poses in absolute time, allowing a fine temporal control of the animation [52]. It was originally designed to describe the motions of Embodied Conversational Agents (ECA) and was extended to be applied on SL avatars by adding new hand configurations, facial expressions and gaze directions [45]. It specifies the low-level animation data of the *k-pose-sequence*, a sequence of key postures corresponding to a sign.

While the previous specification used a phonological definition of signs, **Zebedee** is a sign specification language based on a geometrical definition of signs [44]. Geometric constraints on points, vectors or surfaces replace the set of parameters usually used (hand configuration, orientation, placement). The *Zebedee* model separates signs into two types of temporal units, following the Movement/Hold model of Liddell and Johnson [36]: the *key postures* when the parameters of the motion reach a stable state and *transitions* in-between two consecutive key postures. One of the advantages of this description model is that it can also represent signs with inflections by modifying the geometrical constraints describing the sign (e.g., the difference between [BIG BALLOON] and [SMALL BALLOON] in *Zebedee* will be done by changing the radius parameter). *Zebedee* captures both the temporal and the spatial constraint of sign languages. Furthermore, the geometric nature of the language can highlight the structure of signs. Fig. 11 shows the representation of the sign for [BALLOON] in LSF with the LIMSI avatar performing the sign.

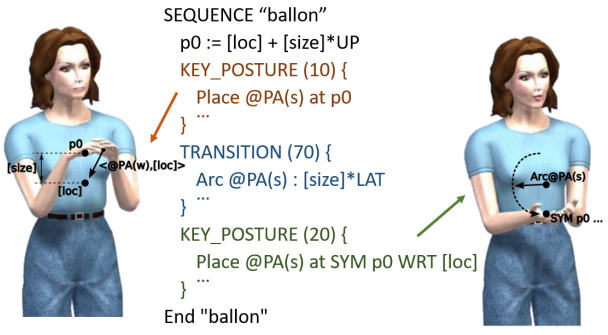

Fig. 11. Example of the representation of the sign for [BALLOON] in LSF with the *Zebedee* language. The left (resp. right) image is the initial (resp. final) position of the LIMSI avatar (avatar and code extracted from [44]).

4.3. Synthesis Techniques

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

4.3.1. Keyframe Techniques

Keyframe synthesis is the most straightforward technique to generate the animation of an isolated sign. An animation is a sequence of avatar poses displayed at a given frequency. Some poses may be more relevant than others – e.g., the state of the avatar at the beginning and end of a sign, or the pose describing an inflection point in the hand configuration. Those key poses, associated with a time tag, are called *keyframes* and a special attention should be paid to their description. Keyframe animation consists in describing the pose of an avatar for each keyframe, the transitions between those keyframes is then automatically computed by interpolation.

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

3D traditional animation consists in setting the avatar in a specific pose at different frames of a timeline using a specialized animation software such as Autodesk Maya [53] or Blender [54] and the knowledge of human anatomy and motion. Different joint angles values are tested and the best values are selected for a particular pose of a particular sign. The process is fastidious and cannot be generalized to other signs. The early work of Shantz [55] in 1982 which is considered to be the first work on sign language avatar animation [56], used this technique.

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

Examples of avatars relying on hand-crafted keyframes:

- *Paula* of DePaul University is an American Sign Language avatar that partially relies on traditional animation and on the PDTS classification [57, 58].
- The Italian Sign Language avatar of the university of Torino is animated using hand-crafted keyframes [59, 60].
- *Elsi*, a French Sign Language avatar designed by the

LIMSI laboratory to be exhibited in French train stations, is based on 3*D* rotoscoping [29, 61].

- The work of Irving et al. [62] proposes to synthesize signs by defining its keyframes in a parametric way according to the Stokoe parameters. Each hand location, configuration, movement and orientation can be described very precisely using sliders in a graphical interface.

- The Turkish Sign Language avatar of Yorganci et al. [63] uses an original approach: the torso and arm motions, the facial expressions and the hand configurations of the avatar are manually modelled separately and are combined in order to generate signs, leading to a parametric and somewhat generic creation of signs.

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

Automatic Keyframing. The tiresome 3*D*-modeling work can be avoided by using the sign specification of key postures of Section 4.2 to automatically compute the key poses of the avatar. Indeed, a lot of those specifications define signs as a sequence of static and dynamic segments intuitively leading to a key postures/interpolation definition of signs. They can be the basis for the definition of spatiotemporal targets for the avatar joints leading to keyframe animation.

A typical approach is to define the hand configuration(s) of a key pose using forward kinematics (FK) by referring to a look-up table where the joint angles of the hand corresponding to each hand configuration are listed. The position of the wrist or the palm of the hand is determined using the placement descriptors of the chosen sign representation and the angles of the arms are computed using the result of an inverse kinematics (IK) algorithm (see Fig. 12 and Fig. 13). The motion between two keyframes is synthesized using different interpolation methods applied to the joints angles.

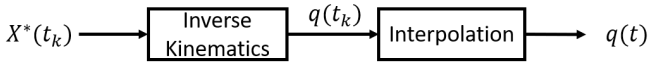

Fig. 12. The hand motion synthesis process for automatic keyframing techniques. An inverse kinematics system is used to compute $q(t_k)$, the state of the skeleton at discrete time t_k corresponding here to the timestamp of the keyframes, from $X^*(t_k)$, the desired position of the hands at t_k. To obtain $q(t)$, the state of the skeleton at time t, the intermediary poses of the skeleton between each t_k are interpolated.

Examples of avatars relying on automatic keyframing:

- Grieve [64] used an adaptation of the Stokoe notation for the ASCII characters (the *ASCII-Stokoe* notation) to place its targets.

- The work of Papadogiorgaki et al. [65] is based on the SignWriting Markup Language (*SWML*).

- The avatars of Krnoul et al. [66] and Fotinea et al. [67] are based on the *HamNoSys* notation.

- Symbolic representation is also used by the VCom3D company to animate the commercial avatars of their Sign 4 Me application [68].

- The avatar of Delorme [69] relies on a segmentation of the motion in terms of key postures and transitions, as defined in the *Zebedee* specification system.

- In the EMBR avatar of Kipp et al. [52, 45], the skeleton key poses are described at a gloss level using the *EMBRscript*. The transition between two poses is smoothened by enhancing the interpolation with temporal modifiers.

- Losson & Vannobel [70, 56] used an analytical approach to compute the hand configuration, placement and movement of their avatar using Losson's specification of signs.

Keyframe animation creates a precisely controlled motion, both temporally and spatially, which is very important for sign language animation. Indeed, signs have to be generated carefully to keep their meaning. However, keyframe techniques,

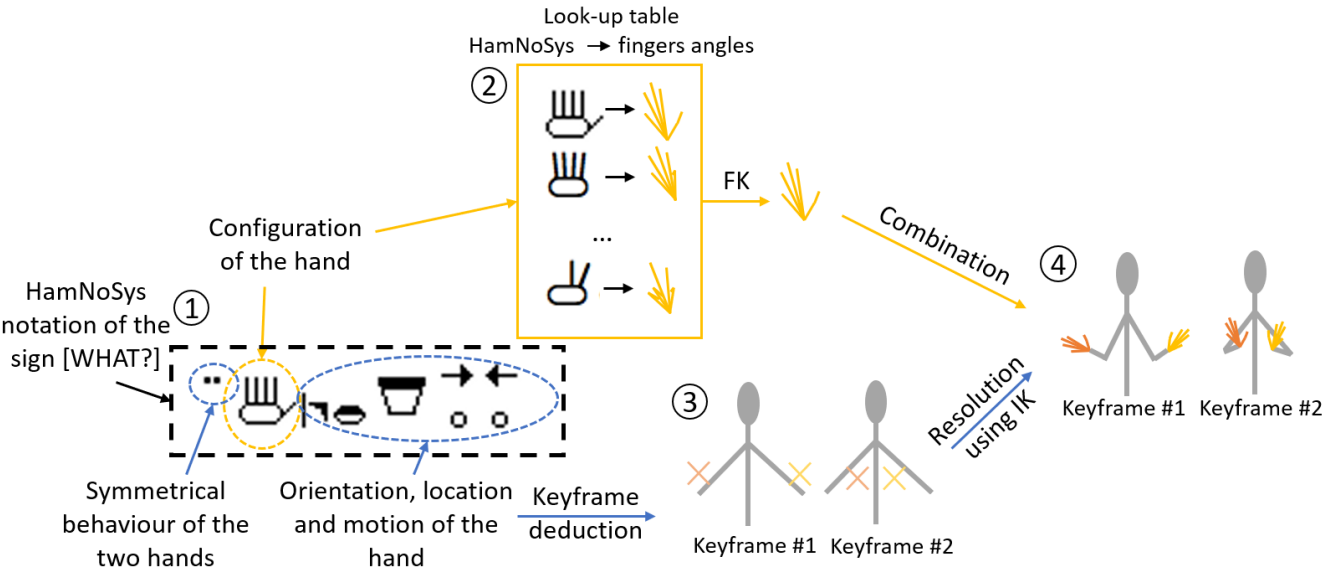

Fig. 13. Automatic keyframing typical process for the sign [WHAT?] in ASL. (1): *HamNoSys* notation of [WHAT?]. (2): Resolution of the hand configuration with FK. (3): Computation of the keyframes and placement of the hands over time based on the *HamNoSys* notation. (4): Combination of the hand configurations and of the hand placement using IK.

whether manual or automatic, while providing consistent animations, are often characterized by robotic motions as interpolation between keyframes does not always convey the kinematic properties of natural motion.

4.3.2. Procedural Techniques

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

Continuous Motion Synthesis. To counter the limitations of keyframing/interpolation techniques which do not guarantee fluid and human-like motions, procedural approaches use kinematics or dynamics control loops in order to generate continuous motion. Those approaches are still based on sign representations which define discrete spatio-temporal targets at the task level. A continuous motion is created in order to reach those targets.

Automatic keyframe animation uses IK algorithms as a way to obtain joint angles for a given keyframe without taking into account the possible intermediary solutions of the IK problem; only the output of the IK model is thus exploited. Conversely, in **procedural techniques based on IK**, the motion of the skeleton is generated by the intermediary postures obtained from the chosen iterative IK method through an IK-based control loop with optional biomechanical or neuromimetic constraints (see Fig. 14). The motion is thus fully generated by IK: no interpolation is used and, depending on the method, the automatically generated motion may appear smoother and less robotic than the motions resulting from automatic keyframe animation. Furthermore, constraints on the trajectory of the joints can be directly integrated in the IK-based control loop to include coarticulation effects [71, 72], or biomechanical synergies [73]. However, if the animation is computed automatically, the quality of the successive postures cannot be guaranteed (possibility of instabilities or unrealistic trajectories), and it is difficult to temporally control the process. Therefore, the resulting animation may be less precise than automatic keyframe animation using IK.

While kinematic animation focuses on the motion and

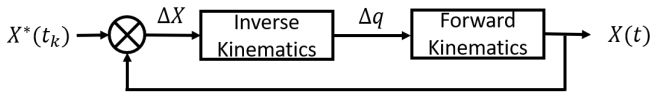

Fig. 14. The hand motion synthesis process for IK-based procedural techniques. At each time step, the difference ΔX between the targeted $X^*(t_k)$ and the actual position and orientation of the hand $X(t)$ is computed to serve as input to an inverse kinematics system. A small displacement of all the skeleton joints Δq is computed, thus estimating the state $q(t)$ of the system at time t, and applied to the skeleton, thanks to a forward kinematics system.

trajectories themselves, dynamic animation concentrates on forces. Instead of guiding the gesture like in the kinematic case, it consists in modeling the forces that lead to the motion. Therefore, it is a powerful tool to obtain realistic reaction when interacting with the environment. Although the physically based simulation of avatars in which the forces of interaction between different parts of the body are simulated would be relevant for signing avatars, this raises issues of precision, interaction, and real-time. In particular the physically based animation of hand movements is still an open problem in the field of computer animation. The degree of accuracy required for sign languages, both in terms of complex manual configurations, simulation time, and multi-segment interactions is not yet reached. Therefore, most techniques developed for signing avatars have focused on kinematic animation which allows to reach both the precision of the animations essential for the hands and the fluidity of the movements. To our knowledge, no work has been dedicated to exclusive dynamic animation for the application of sign language synthesis. Still, some work takes advantage of the realism provided by **dynamic controllers** combined with IK (see Fig. 15).

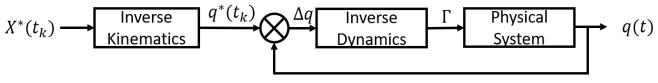

Fig. 15. The hand motion synthesis process for dynamics-based procedural techniques. First, $q^*(t_k)$, the desired state of the skeleton at discrete times t_k is computed from $X^*(t_k)$, the desired position and orientation of the hands, using an inverse kinematics system. Then, at each time step, the difference Δq between the desired state of the skeleton $q^*(t_k)$ and the actual state of the skeleton $q(t)$ is computed to serve as input to an inverse dynamics system. A force Γ is computed and applied to the skeleton, thanks to a physical system, to obtain $q(t)$.

Examples of avatars relying on continuous motion synthesis:

- *GessyCA* [25, 72], the synthesis engine based on *QualGest*, takes as input a discrete sequence of weighted spatio-temporal targets to produce the motion. Both the hand movements and the hand configurations are generated with a sensorimotor **IK-based** control loop including the modeling of biomechanical synergies and a neuromimetic sigmoid function. Furthermore, by taking into account coarticulation effects in the optimization process, phonetic targets implicitly contain information from the past and future context, leading to the generation of a more human-like motion.

- In the *VISICAST* project [74], a **dynamic** model coupled with IK targets is defined: each joint is represented as a control system for which the controlled variable is the angle of the joint. Joints are virtual masses whose acceleration is proportional to the force computed to reduce the error between the current angle and the desired angle of the joint. The trajectory of the arm motion is computed using IK and the angles deduced by IK are then fed to the controllers as reference angles.

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

In order to add some realism to the animation, some mechanisms can be implemented. One of the preferred techniques is to add some small human imperfections to the avatar animation to avoid too stiff postures or too perfect performances. Adding noise, ambient motions, autonomous behaviors ensuring breathing motions, or modifying the timing of the motion on the different channels are recurrent methods used to improve the realism of avatar motions. Signal processing techniques based on motion capture data analysis can also be used to improve the realism of avatar motions [58].

Examples of avatars using procedural means to add realism:

- In the ViSiCAST project [74], noise, damping and ambient motions (small random eye / head / torso motions) have been added to reduce the robotic and unnaturally stiff movement inherent to their dynamics-based model.
- A small de-synchronization of the hands during the performance of symmetrical signs has been added to the Irish Sign Language avatar of Smith et al. [75].
- In the EMBR project of Héloir & Kipp [52], an autonomous behaviour ensuring breathing and some natural movements like blinking as well as blushing effects has been implemented.
- To liven the avatar *Paula*, noise has been added to its movements [76].

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

4.3.3. Data-Driven Sign Synthesis

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

Motion Capture (MoCap) is used to record the position of markers placed on a human being performing a motion. The position and orientation of the human's joints is then deduced from the MoCap data during the post-processing step.

Fig. 16 describes the offline motion database creation process. In this section, we are only interested in the synthesis methods based on a *MoCap* database. However, the annotation step which consists in dividing the continuous flow of motion into smaller segments (*segmentation*) and *labeling* those segments, directly impacts the motion synthesis. This step is necessary and fundamental for the editing process. For SL gestures, it aims both to identify linguistic features, and to find precise temporal boundaries between linguistic units. The challenges raised by the annotation issues are presented in [77] and [28]. Manual annotation is a fastidious task whose automation has been widely investigated [78, 79, 80].

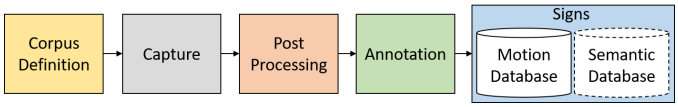

Fig. 16. The offline sign database creation process. The motion database contains the MoCap data while the annotations constitute the semantic database.

Currently, MoCap is used by research labs more for analysis (MOCAP1 and Signaire 3D [81], two corpora of the LIMSI laboratory in France or Lu & Huenerfauth's work [82] for example) than for synthesis purposes (SignCom [83] and Sign3D [28] projects) [84]. Indeed, MoCap does not only supply motion sequences to be played back on an avatar, it also provides material to be analyzed and from which motion features and motion invariants can be extracted and potentially reinjected into the animation process.

It is technically impossible to capture all the signs in all the possible contexts due the size of the SL vocabulary, the multiple inflection mechanisms of signs and sentences in SL, the need for various Deaf participants, and time and memory constraints. Therefore, synthesizing signs using a limited set of pre-recorded signing sequences is the major challenge of data-driven techniques. Three main approaches exist to create isolated signs using motion capture data:

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

Regardless of the chosen synthesis technique, data-driven synthesis involves motion retrieval. It consists in choosing and extracting the best motion(s) for a particular application

among a set of motions. It can be done by (i) directly querying the motion features (e.g., see Kapadia et. al. [85] for motion retrieval of non linguistic motion using Laban movement analysis), or (ii) using a textual search in a semantic database containing the annotation of the motion [83, 86]. In this second case, the retrieval process relies on a management system database, separating the two levels of representation (semantic symbols and raw motion data) (see Fig. 17). This is the most straightforward technique as the motions of sign language are semantically meaningful. However, motion feature query remains interesting to decide between several motions with the same annotation label [87], or to extract motions at a finer level than gloss.

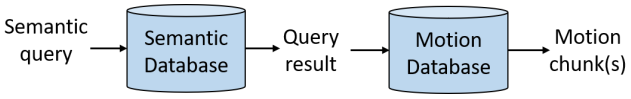

Fig. 17. Semantic motion retrieval.

Assuming that the quality of the captured database and post-processing is correct, the **play-back of a pre-recorded sign** will give an accurate and realistic motion. Indeed, the main advantage of MoCap technologies is that it preserves the human qualities of motion. The motion and dynamics of the generated sign will therefore be perceived as natural and credible, thus increasing the acceptance of an avatar driven by such a method [88]. However, while this technique can be used to generate new utterances by concatenating existing signs as we will see in Section 5.2.1, the generation of new signs, not existing in the original database, or the editing of a sign in order to adapt it to a new context are not the objectives of the play-back approach. The utterances produced using this technique are therefore limited by the number and variety of the recorded data.

The existing database can also be augmented by **editing** the pre-recorded motion using synthetic animation techniques. For example, in [83], some signs are created by inverting the direction of the hand motion or by modifying the hand configuration of an existing sign. The sign [DARK] in French Sign

Fig. 18. The Sign360 application of MoCapLab [89, 90].

Language can be created by inverting the hand trajectory of the sign [LIGHT] (see Fig. 19) whereas the object of the sign [TO GIVE] can be modified by changing the hand configuration (see Fig. 5).

Motion warping is also a motion editing technique that can be applied to add variability in sign language generation. It consists in altering a motion of the database by changing its trajectory while keeping the kinematic properties of the motion [91]. This is a promising technique for data-driven synthesis of SL, but so far little work has been dedicated to the development of this technique. Another approach, Dynamic Time Warping (DTW), can be used to introduce temporal variability in the sign language production. This technique can be used: i) as an elastic distance to retrieve motions in the database; ii) as a synthesis process to stress/compress the edited motion; iii) as a process to generate style transfer whether at a phonological level [92], or at an utterance level [93]. In terms of style transfer, Gerard, the avatar of [94], could generate stylized sign sequences in LSF by learning the time warping of other recorded sequences. As data-driven techniques, motion warping and DTW ensure natural-looking kinematics of the resulting movements. However, the realism of the transformed and edited movements has still to be verified with respect to the generated postures and the meaning of the synthesized sentences.

Machine learning methods are another way to reuse and

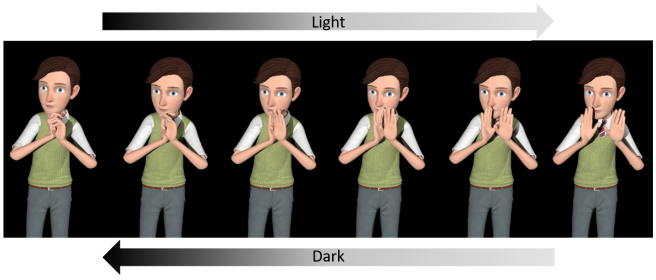

Fig. 19. The inversion of the hand trajectory transforms the sign [LIGHT] into the sign [DARK].

generalize motion data. It relies on the captured data to extract knowledge and models that can provide new plausible and contextualized movements. Carreno et al. [95] present a very thorough survey on motion synthesis based on machine learning methods. For the particular case of sign language synthesis, Huenerfauth et al. [16] studied and synthesized directional verbs like [ASK], [MEET] or [SEND] using MoCap data of SL performances from fluent signers as a training data set to learn their synthesis models.

Examples of data-driven avatars:

- *Tessa* [96], a British Sign Language avatar, and *Simon* [97], a Sign Supported English[6] avatar both take advantage of the play-back technique.
- Play-back of signs in Swedish Sign Language was also done by Alexanderson et al. [98], but the main concern of their work was the study of motion capture and skeleton reconstruction.
- The *Sign3D* project combines both play-back and sign synthesis editing techniques [28].
- For commercial applications, *Sign360* of the French company MoCapLab presents a French Sign Language avatar driven by pre-recorded gestures [90, 89] (see Fig. 18).

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

5. Utterance Synthesis

In the previous section, we reviewed the processes and techniques for isolated sign synthesis. This section is dedicated to the synthesis of full utterances. An utterance is a set of co-occuring and/or sequential signs, that represent the statement of an idea. It is close to the concept of "sentences" in oral languages. Utterances are composed of signs in their citation forms and of inflected signs (see Section 3.2).

5.1. Utterance Representation

Since an utterance is composed of a set of signs, it is possible to take advantage of the representations of the signs seen in Section 4.1 and in Section 4.2 to specify an utterance. However, a simple concatenation of these signs in their citation form would be incorrect in the same way that a sentence without conjugating the verbs would be incorrect in oral language. It is important to take into account the grammar and semantics of SL in order to inflect the signs that need to be inflected and to coordinate the different body channels of the avatar. First, we describe four challenges of the representation of an SL utterance and show, using examples, how the representation by a sequence of glosses may be inadequate. Then, we describe and show the advantages and disadvantages of four utterance representations.

5.1.1. Limitation of the Representation with a Sequence of Glosses

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

However, utterance representation is a complex problem for which a sequence of glosses is not the appropriate solution as co-occurrences of signs are common phenomena. The main

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

challenges of utterance representation are listed here. We illustrate each challenge with an example that highlights the limitations of the gloss representation.

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

The direction of the torso, shoulders and gaze, for example, are often used in **role-shift** cases (see Section 3) and are not synchronized to a particular sign. In this case, the signer takes the role of the person, animal or object he/she is describing or whose words he/she is telling. If he/she repeats a dialogue he/she has heard between a person A and a person B, A's statements can be transmitted with a movement of the shoulders, chest and gaze to the left, while B's statements will be reported with a movement to the right. These movements are associated with the whole statements and not with a particular sign. It is therefore an overlay at the statement level.

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

$$[\text{I}][\text{DANCE}][\text{NOT}] \quad (2)$$

A signer will often make the negative headshake from the beginning of the [DANCE] sign until the end of [NOT] (see Fig. 20). However, the representation by a sequence of glosses as shown in (2) does not provide this information. The [DANCE] sign is not affected by the negative aspect of the sentence, which can result in incorrect sentences in synthesis. One solution is the use of *parameterized glosses* in which contextual information is provided. Parameterized glosses are used by various researchers [99, 3, 100] but no standard exists.

Thus, (2) would become:

$$[\text{I}][\text{DANCE\_not}][\text{NOT}] \quad (3)$$

Where [DANCE_not] and [NOT] include the negative headshake. But even so, the movement of the head would be synchronized on each of the signs and not on the whole {[DANCE][NOT]} as it should be. Glosses, simple or parameterized, result in an *over-synchronization* (the synchronization is carried out on too many synchronization points) at the gloss level.

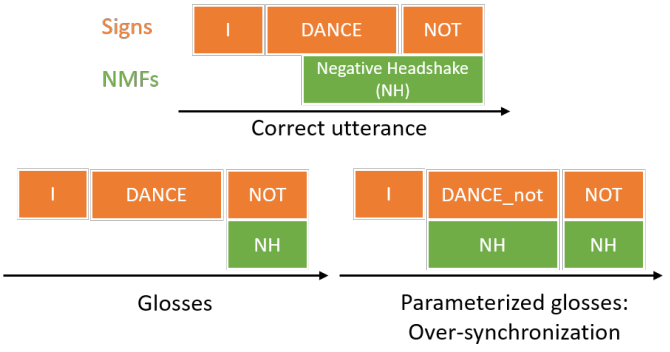

Fig. 20. Example of the the sentence "I did not dance". Top: the correct synchronization between the signs performed with the hands and the NMFs. Bottom left: the result of a simple sequence of glosses. [DANCE] does not include the negative headshake. Bottom right: an example of over-synchronization with parameterized glosses.

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

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

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

Let us take the example of the sentence "The frog jumps into the lake". This sentence should be signed with two proforms: one for the lake with the non dominating hand performing a wide C-shaped configuration to show the outlines of the lake, and one for the frog with the dominant hand. The latter jumps

into the lake, which results in a jumping movement of the dominant hand towards the space delimited by the non dominant hand.

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

[LAKE][LAKE_pr][FROG][FROG_pr][JUMP] (4)

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

Sign Inflection. The contextualization of the signs leads to modifications of the citation form of some signs in order to insert them into the utterance. Inflections can be of several types (see Section 3.2) and characterized by a change in hand configuration (e.g., proforms), trajectories (e.g., directional verbs) or amplitudes (e.g., size specifiers). Simple glosses do not provide such information.

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

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

[I][HE][SMALL][BALLOON][GIVE] (5)

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

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

[BALLOON_small][GIVE_*'O' config*_I → he] (6)

The representation (6) only contains two glosses, a first gloss naming the specified object and a second gloss corresponding to an inflected sign that can be retrieved from a database or generated on-the-fly. This parameterized representation will certainly achieve a result more similar to what a real signer would do. However, it implies a very large database containing an important number of inflected signs or a synthesis engine capable of understanding the syntax of the representation and of creating the corresponding sign (e.g., by creating the desired size of the object as a linear combination of the two extreme sizes of the same object present in a database).

Timing Information. There are two philosophies in timing management for signing avatars. This timing can be (i) indicated explicitly, in a relative or absolute manner, in the representation of the utterance at the input of the synthesis model (in this case, it is an additional constraint for the model) [45], or (ii) computed by the synthesis model according to the constraints of the system [28, 60]. For example, if the sentence is a concatenation of movement segments present in a database, the timing will be constrained by the length of these segments. Sequence of gloss representations, whether parametric or not, do not account for the timing but only for the linear order of the signs.

5.1.2. Examples of Representations

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

EMBRScript: Representing Absolute Timing. The *EMBRScript* represents non-contextualized signs in the form of *k-pose-sequences*. With this script, an utterance is considered as a sequence of glosses, except that the timing of the signs is explicitly indicated in an absolute way [45]. Apart from this timing information, the shortcomings of this representation are the same as those of the simple gloss sequence representation (no consideration of the context and synchronization of non-manual channels).

Example:

I 300 → 1190

DANCE 1220 → 2620

NOT 2650 → 3000

ATLAS and HLSML: Representing Inflections. For the *ATLAS* project of the Italian team of Turin, a focus is done on the inflected signs: a sign is defined as the combination of a base sign (the citation form) and context-dependent modifiers. These modifiers can be of different types: sign relocation, speed modification, trajectories modification, sign resizing, sign iteration or hand configuration modification [60, 59].

Their article [59] takes the example of the sentence “Cloudy at north-east. During the evening, cloudiness increases at north-west” which is interpreted in their animation language as:

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

In the same way, HLSML was developed by López-Colino et al. [101, 102] for their Spanish Sign Language avatar. It uses an XML-based notation to represent inflected signs (see Fig. 21) or to change the location of a sign. Moreover, time information can be included in the representation to constrain the realization of the signs.

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

Fig. 21. The representation of the inflected sign [TO GIVE] in the expression "to give a book" in HLSML. The hand configuration used for the sign [TO GIVE] is the one of the sign [BOOK] named "clBOOK" ("book classifier").

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

P/C Model: Representing Synchronization. A common way to analyse a sentence structure in natural language processing for oral languages is to display it in the form of a syntax tree [103]. However, the multichannel aspect of SL makes it difficult to describe an SL utterance as a syntax tree. In his work, Huenerfauth proposes to represent an SL utterance in the form of a 3*D* syntax tree [104]. He thus defines a formalism, called *Partition/Constitute (P/C) model* which is a 2*D* representation of a 3*D* syntax tree to which restrictive rules have been added. This representation allows to visualize from left to right the temporal axis and from top to bottom the body channels. The nodes of the 3*D* tree are represented by rectangles (leaves are the "atomic" rectangles that surround a text while the root is the rectangle that surrounds the whole, the other rectangles are the different branches). This representation allows to manage the coordination of the different channels between themselves as well as the use of certain proforms (see Fig. 22 and 23).

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

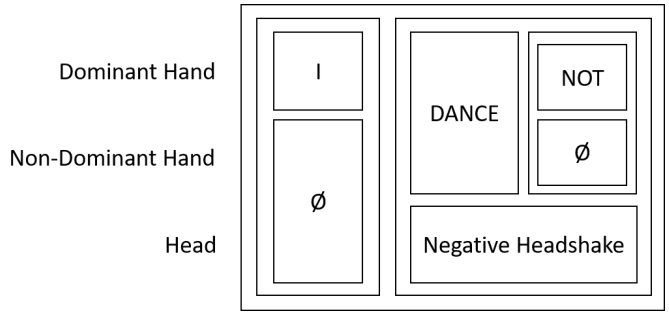

Fig. 22. Example of the P/C modeling of the sentence “I did not dance” in ASL. The coordination of the negative headshake with the signs [DANCE] and [NOT] is explicit (figure done using the representation described in [104]).

Azee: Representing Function-to-forms Associations. With the *Azee* representation, the common assumption that an SL sentence is defined as a sequence of glosses is questioned. The originality of *Azee* is that it is based on the minimal linguistic assumption that the language is a system where observable forms are associated, in a systematic way, to a meaning. To capture those systematic associations, *Azee* implements *production rules*: invariant function-to-forms correspondences where

Table 2. Comparison of the utterance representations.

Representation	NMFs synchronization	Co-occurrence of Signs	Sign Inflection	Timing Information
Sequence of glosses	✗	✗	✗	✗
Sequence of parameterized glosses	✗	✗	✓	✗
EMBRScript [45]	✗	✗	✗	✓
ATLAS [105, 60]	✗	✗	✓	✗
HLSML [101, 102]	✗	✗	✓	✓
P/C Formalism [104]	✓	(✓)	✗	✗
Azee [106, 107]	✓	✓	✓	✓

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

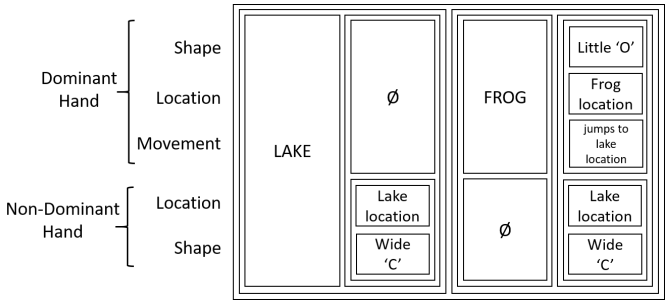

Fig. 23. Example of the P/C modeling of the sentence "The frog jumps into the lake" in ASL. The ∅ symbol for the non-dominant hand during the realization of the [FROG] sign makes it possible for the non-dominant hand to hold the classifier of [LAKE] (figure done using the representation described in [104]).

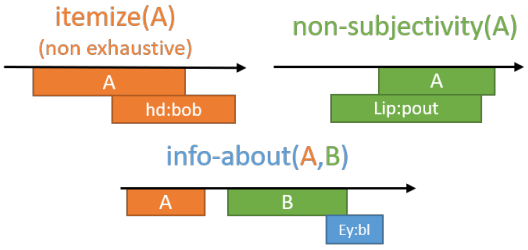

Fig. 24. Individual production rules of Azee (figure inspired from [109]). The *itemize(A)* production rule, for instance, designate the function-to-form association used to do a non-finite enumeration. It makes it possible to create expressions like "theatres, restaurants, etc." in which case, the *itemize(A)* rule is used twice with the parameter "A" taking successively the values "theatre" and "restaurant". When performing the enumeration, a head movement (hd:bob) is done at the end of each item.

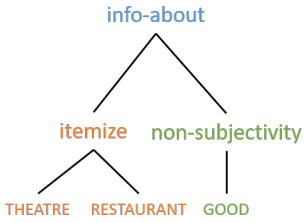

Fig. 25. The rule tree of the sentence "Theatres, restaurants, etc. are usually deemed good" (figure inspired from [109]).

a *function* is the desired semantic meaning produced by the *forms*, the "*visible states and movements of the body articulators*" [106].

Those production rules capture, using the same process, the formation of isolated signs (in this case, the states of the different channels for the production of this sign will be the *forms*, while the meaning of the sign, the gloss, will be the *function*) and higher-level syntactic mechanisms such as the relationship between entities (e.g., "A is an instance of B" or "A is an information on B") [107, 108]. Some rules can thus be parameterized according to the context (parameters A and B of the previous examples) and, since all mechanisms are put on the same level, the nesting of production rules is done in a natural and direct way. A and B can be isolated signs or complex syntactic functions. A rule tree allows to visualize this type of nesting (see example of rules in Fig. 24 and a rule tree in Fig. 25).

In practice, the production rules are derived from the analysis of the content of a corpus until an invariant *form* is found for a large number of instances of the same *function*. For example, if a left-to-right headshake is found with many instances of negative utterances, a production rule associating the form *left-to-right headshake* to the function *negation* will be created. These production rules can be written in the form of temporal scores (see Fig. 26) and are unambiguously interpretable by a synthesis system. The rules are defined without *a priori*: the lexical sequence as the base of an SL production is therefore questioned. The sequence of signs is seen as one possible *form*

in the same way as an eye blink or a headshake. In addition, if timing information has been found consistently for many instances of the same *function*, this timing is added to the *forms* of the *function*.

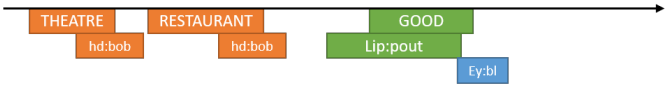

Fig. 26. The *Azee* representation of the sentence "Theatres, restaurants, etc. are usually deemed good" in the form of a sign language score (figure inspired from [109]).

Moreover, there is ongoing work in order to associate *Azee* formalism with a pictogram representation to make it more accessible to the deaf community and obtain a writing system that can be directly interpreted and synthesized by an animation system [110].

Table 2 provides an overview and a comparison of the utterance representations.

5.2. Utterance Synthesis Approaches

Two main utterance synthesis approaches are often distinguished: the **concatenative** and the **articulatory** synthesis [64] (see Fig. 27). In the case of concatenative synthesis, the utterance objective is considered as a set of smaller objectives. Concatenative synthesis involves a motion/sign database and consists in concatenating small chunks of existing data while articulatory synthesis computes the sign language utterance directly from the gesture specification. Hybrid avatars with animation systems taking advantage of both techniques are also emerging.

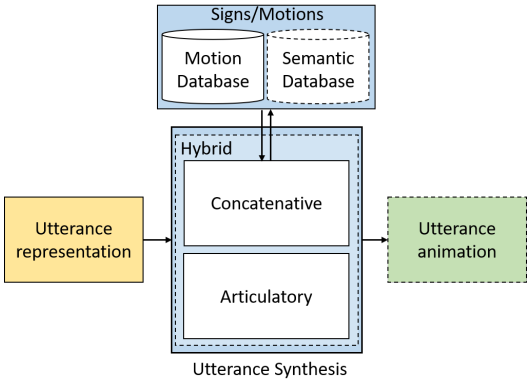

Fig. 27. Overview of the utterance synthesis techniques.

5.2.1. Concatenative Synthesis

It is the process in which chunks of pre-recorded or pre-synthesized motions are concatenated successively and/or on the different SL channels (see Fig. 28). Motion interpolation or blending techniques are used to build the transitions between two chunks of motions. Motion blending consists in doing, at each frame, an interpolation of motions present in the database to create new motion having the characteristics of the initial motions. The quality of the result strongly depends on the length and granularity of the chunks of motion. The simplest and most common approach is to consider motions at a gloss-level, but smaller motion chunks can also be used, leading to a more precise control of the avatar animation. Motion synthesis is therefore based on an annotated database of signs (or finer grained motions) that is queried to compose the utterance. The annotation of the data is essential: the granularity of the synthesis is restricted by the granularity of the annotation, and the presence or absence of artifacts in the final animation depends partially on the quality of the segmentation of the data. The database is built and annotated offline while the sign concatenation can be done online. Utterance representations based on sequences of glosses are particularly suited to this synthesis technique. The following paragraph details the different avatars technologies that make use of concatenative synthesis for utterance synthesis depending on the nature of the content of the original database.

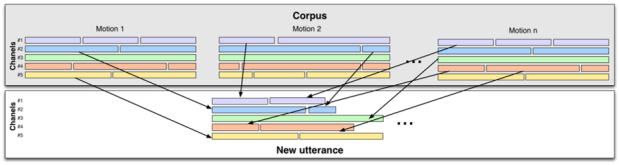

Fig. 28. Multichannel Concatenative synthesis (image extracted from [83]).

Using Hand-Crafted Animations. In this case, concatenative synthesis relies on a database of hand-crafted motions and on motion interpolation to smoothen the transitions between the concatenated gestures. *Paula* [57] and *Elsi* [29], the avatars presented in Section 4.3.1, rely on this principle. *Paula*'s motion is created in real-time by automatically combining and possibly modifying the hand-crafted key postures in accor-

dance to the desired sign language utterance. *Elsi* can sign full utterances created by concatenation of isolated signs designed by rotoscopy. Unfortunately, she can create very few novel utterances as rotoscopy does not provide for context-dependent signs. However, this method suits the application of *Elsi* who does not need to convey novel messages since she has to provide information in train stations to deaf passengers by combining pre-recorded fill-in-the-blank sign language utterances and isolated signs.

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

- *Paula* [57].
- *Elsi* [29].
- The Turkish Sign Language avatar of Yorganci et al. [63].
- The Spanish Sign Language avatar of López-Colino et al. [101, 102].

Using Automatically Generated Signs. It is also possible to build a sign language utterance by editing and concatenating signs extracted from a database of automatically keyframed or of procedurally synthesized signs. Annotation tags are queried to retrieve the relevant keyframes/signs from the database. The avatars relying on automatic keyframing for the generation of isolated signs (Section 4.3.1) often synthesize utterances by simple concatenation of whole signs.

Examples of avatars using concatenative synthesis with automatically generated signs:

- The EMBR avatar of Kipp et al. [45, 52].
- The avatar of Krnoul et al. [66].
- The parametric avatar of Irving et al. [62].
- The *eSIGN* avatar [111].

Using MoCap Data. Concatenative synthesis on MoCap data consists in combining and concatenating previously captured motion chunks (often corresponding to a sign or gloss segment). The transition between the motion chunks can be done using different kinds of interpolation and blending methods [24]. This concatenative synthesis produces realistic motions on play-back sequences but the quality of the utterance as a whole strongly depends on the quality of the synthesized transition segments. Moreover, the signs will be played by the avatar the exact same way as they were recorded. It means that the objects described in the new utterance will have the same aspect as the ones recorded. The creation of novel utterance is thus limited by the motions available in the database.

Examples of avatars using concatenative synthesis with MoCap data (see section 6 for more details):

- The Sign3D avatar [28]. This system, characterized by high fidelity captured motion (both corporal and manual data, facial expression and gaze direction), proposes data-driven synthesis at the sign/gloss level.
- The SignCom avatar [83]. This system allows for concatenative synthesis at a phonological level.

5.2.2. Articulatory Synthesis

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

Sentences build this way can be precisely controlled but the work of describing the utterance using a sign or a gesture specification (taken as input of such a system) can be extremely tedious. Furthermore, the resulting animation lacks realism and

may be rejected by the Deaf community (in [112], the authors show that "natural movement" are the second most important requirement for the acceptance of the avatar). To our knowledge, only the *GessyCA* system [72] uses the articulatory synthesis alone to build utterances. The sequence of signs described with *QualGest* is assembled by combining sequentially and in parallel the atomic *gestems* using synchronization operators. Procedural approaches to animate an avatar at a sign level from a sign representation are commonly used but they are rarely extended to the utterance level. However, hybrid approaches that mix articulatory and concatenative principles are more developed.

5.2.3. Hybrid Synthesis

Hybrid models take advantage of the strengths of concatenative and articulatory synthesis. The CUNY American research group worked on both procedural animation techniques and data-driven synthesis approaches using MoCap data and compared the two approaches in [82]. The sign language research team of the University of East Anglia also studied both technologies, first with *Tessa* (concatenative) [96], followed by the eSIGN project (articulatory) [113]. They proposed to use MoCap data to add realism to their procedurally animated avatar.

DePaul University research group implemented this idea on *Paula*, their avatar animated with hand-crafted keyframes [58]. Moreover, *Paula* was recently improved with IK solvers that procedurally modify the hand-crafted animations in order to synthesize context-dependant mechanisms such as proforms making it a fully hybrid avatar [114, 107]. The Italian Sign Language avatar of Lombardo et al. [60, 59] defined a contextualized sign as the combination of a base sign (defined mainly by manual and automatic keyframing but also by MoCap) and context-dependent modifiers (procedural methods or hand-crafted poses). It showed great promise but the project seems to have been frozen as no subsequent article has been published on the subject. In the same philosophy, the ASL avatar of Adamo et al. [115] relies on a multilayer system with a concatenative synthesis engine to query individual signs that are altered by procedural prosodic modifiers.

6. Existing Systems

In this section, we present three SL avatars using different sign and utterance synthesis approaches. To have a more exhaustive overview, Table .3 lists and describes the existing avatars.

6.1. JASigning and AnimGen

The Java Avatar Signing (*JASigning*) system is a multiplatform tool for the synthesis of any sign language [116]. It is freely available for research purposes[7]. It integrates *AnimGen*, an animation engine for SL synthesis developed for the ViSiCAST [74] and, later, the eSIGN [111] projects.

For isolated sign synthesis, *AnimGen* takes as input the *SiGML* notation [47] of a sign and translates it into low-level parameters for the animation based on inverse kinematic controllers.

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

The *AnimGen* module was used internationally in numerous works from 2001 until today [21, 113, 116, 117, 74, 48, 118, 75, 119]. Notably, Hanke et al. [117] whose work focuses on the study of timing differences on the SL channels, used *AnimGen* to test their hypothesis on the synchronicity between the hand configuration with respect to the hand location. In more recent work, *JASigning* constituted the basis of Ebling's animation module to test her machine translation system [21].

The *AnimGen* module is suited for simple translation tasks as it allows the creation of new signs and utterances. Indeed, it only depends on a sign representation (*SiGML*) and not on annotated data. However, the motions generated lack the naturalness of human motion. In addition, it depends on a concatenative synthesis module to create utterances: simultaneous

[7] http://vh.cmp.uea.ac.uk/index.php/JASigning

mechanisms cannot be captured by *AnimGen* and the inflections that can be added to the signs are limited by *SiGML*.

6.2. SignCom and Sign3D

The avatars *SignCom* [83] and *Sign3D* [28] aim to animate French Sign Language avatars with natural and realistic movements from captured human data. They rely on two databases: (i) a raw motion database in which the captured motion is stored as a hierarchical structure (the skeleton) with a set of transformations applied to each joint, and (ii) a semantic database that serves as a mapping between the gloss level annotations and the movements contained in the first database. The *Sign3D* system operates at a sign/gloss level. Following a stereotyped syntactic scheme, it allows to precisely synthesize novel utterances by replacing signs or groups of signs. The building of utterances is done following an interactive graphical language [28]. *SignCom* is a multichannel system that operates on parallel channels (lower body, torso, arms, hands, head, face, and eyes) corresponding to phonological tracks [83]. The channels are traversed in a tree-like manner through a scripting language that activates animation engines adapted to the different parts of the body. Interpolation, blending and fading in/out operations ensure co-occurrence between channels.

For the synthesis of isolated signs, the appropriate sign is retrieved from the movement database. To do this, the semantic database makes it possible to find the movement(s) labelled with the desired gloss. If several movements correspond to the gloss, a choice can be made using various criteria: length of the time segment, profile of the movement, etc. The sign can be edited if necessary with a temporal inversion of the signal to change [TAKE] to [GIVE] or [LIKE] to [DO NOT LIKE]. As in most data-based systems, the synthesis of novel signs remains limited by the content of the MoCap database. For utterance synthesis, the captured and possibly edited captured signs are concatenated and the transitions are synthesized by motion interpolation or motion blending. An additional criterion for the retrieval of signs can be added: distance from previous and/or following movements [24].

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

6.3. Paula-Azee

Paula (see Fig. 29) of DePaul University is an American Sign Language avatar based on a multi-track animation engine while *Azee* is a language modeling system, first used for French Sign Language, that aims to represent the syntax of SL utterances in a non-sequential way (see Section 5.1) [120]. *Paula* and *Azee* were developed separately but they both rely on a multi-track system which makes their connection possible.

Fig. 29. *Paula* from the DePaul ASL Avatar Project (image extracted from [121]).

Paula is an hybrid system mainly animated with hand-crafted keyframes but that can also take advantage of procedural approaches to add precision or realism. To build the set of hand-crafted animations of the isolated signs, the *PDTS* classification of Jonhson & Liddell (that defines signs as a sequence of postures and transformation segments) is used as a template to manually draw the keyframes of *Paula* [57, 58]. Procedural animation of the spine can be added to increase the naturalness of the animation.

For utterance generation, the hand-crafted animations of *Paula* can be modified procedurally thanks to IK solvers in order to synthesize proforms or spatial referencing mechanisms [114, 107]. Moreover, *Paula*'s animation system relies on a timeline of parallel tracks, including "mouth shape" track,

"blink" track, "head movement" track, or "gloss" track, that is very similar to the *Azee* sign score shown in Fig. 26. Animation blocks can be placed in an asynchronous way on the different tracks of Paula's animation timeline. The resulting animation is a combination of the effects of the multiple processes. A mapping between *Paula*'s utterance specification tool and *Azee*'s formalism was created allowing *Paula* to be driven by *Azee*'s linguistic modeling [107].

7. Discussion

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

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

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

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

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

Still, those synthesis methods lack the expressiveness of human motion whereas, in data-driven techniques, the resulting animation has the authenticity of natural human motion without needing to add special treatments. MoCap is thus a great tool for linguists and computer animators to analyze and synthesize motion. It can be a way to find motion laws using statistics on the data or to observe some linguistic phenomena of interest. However, the calibration, capture and post-processing of the data are complex, tedious and time-consuming. The MoCap equipment can be invasive with the presence of markers or sensors potentially impacting the realization of the signs.

The capture of SL signs and utterances can be performed in a limited space as the signer does not move during the linguistic production but it also brings important technical constraints [122]: (i) the need to accurately capture gestures with small but meaningful variations (the finger motions particularly), (ii) the temporal dynamics (velocity, acceleration, jerk) must be preserved which requires a high sampling rate and (iii) the whole body is involved in sign language production: facial expressions, torso motions and manual characteristics must be captured simultaneously. Concerning the content of the corpus, two main trends are often confronted. In Duarte's PhD thesis [123], they are named *breadth* and *depth*. Breadth consists in capturing a large number of different signs to try to cover

the whole language, while depth consists in capturing a limited set of signs numerous times with many variations (for example, by varying the location of the sign in space). As sign languages do not contain a finite number of signs due to mechanisms such as proforms or role-shifts, and because capturing data is costly, covering all the possible signs in all the possible contexts (the *breadth* solution) is not a viable design solution. Moreover, the motion data, after being captured, is not directly workable. In the case of passive MoCap, the signer whose motion are recorded is covered by numerous markers (sometimes more than 100). Those markers are often not or wrongly identified by the motion capture system leading to a task of manual verification and relabelling. Then, potential gaps due to the occlusion of markers have to be filled. The resulting data (either from active or passive MoCap) may be filtered to remove unwanted noise. Finally, the joint orientations of the skeleton have to be reconstructed from the sensor data or from marker positions in order to obtain workable skeletal motion data in the form of a motion file.

The use of data leads to the development of optimized motion retrieval techniques to prevent slowing down the animation process. Moreover, the data must be annotated at a more or less fine-grained level depending on the animation technique. This annotation work can be tedious and time-consuming, especially when done manually. In addition, new sign language utterances are hard to generate from a limited set of motions in the database and context-based variation in the captured motions is not easy to synthesize which is a problem considering the great iconicity and variability of signed languages. Machine learning methods associated to MoCap data are a promising way to reduce the limitations of data-driven techniques. In the near future, with the advent of deep learning, captured motion data could be replaced by video data, thus facilitating the generation of SL content.

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

In the case of utterance animation, concatenative synthesis consists in concatenating chunks of motion (corresponding to the isolated signs previously defined or to sub-lexical structures) in order to create the utterance. This kind of synthesis is based on a database built offline and often relies on a sequence of glosses to retrieve the relevant motions. The transitions between the motions are smoothened using signal processing functions like blending or interpolation, and modifiers of the original signs are often added to add prosodic cues or to take into account the context of the sign. Concatenative synthesis is an inherently sequential technique that can be seen as contradictory to sign languages philosophy; however, concatenating chunks of motions along the temporal axis are not the only option of concatenative synthesis. Little work has been dedicated to the concatenation of smaller motions on the different channels of sign language (hand configuration, motion, orientation, facial expression, etc.) and this could be the focus of future research. Currently, data-driven utterance synthesis systems rely on concatenative synthesis to achieve three linguistic purposes: (i) replacing signs or groups of signs within an utterance, (ii) replacing phonetic or phonological components and in this way modifying the grammatical or semantic aspects of the utterance, or (iii) altering prosody in the produced sign language utterances [84].

Articulatory synthesis aims to build sentences on-the-fly based on a sign or gesture specification. The animation can be computed from biocontrol or inverse kinematics models. This approach can sometimes lead to real-time animation as no data has to be retrieved from a pre-built database.

The benefits and drawbacks of articulatory and concatenative approaches are oddly complementary. On the one side, the articulatory techniques allow for a real-time generation of novel utterances but are poorly accepted by the Deaf community due to their inexpressive and robotic motions. On the other side, concatenative approaches based on MoCap technologies or hand-crafted signs allow for authentic, human-like motion. However, the variety of utterances that can be synthesized is limited by the initial corpus. Moreover, the sequential aspect of concatenative approaches, enforced by the sequential representation taken as input, does not do justice to the richness of the language. However, new, non sequential ways of specifying utterances like *Azee* [106] or the *P/C Formalism* [104], or specifications taking sign inflections into consideration such as *ATLAS* [105] or *HLSML* [101], are being developed to overcome the limitations of the current representations.
Hybrid models, taking advantage of the strengths of both the concatenative and articulaty approaches could result in a generic and well accepted avatar. Hybrid synthesis is a promising technique for sign language animation and will certainly be one of the main concerns of future work.

Moreover, even though Non-Manual Features (NMFs) are mentioned and that some of the described techniques are used to animate the face and torso of the avatar, the focus of this survey is the animation of the avatar's arms and hands. By no means do we want to lessen the importance of NMFs which are paramount in any sign language animation, a great part of the meaning being conveyed by them. A survey dedicated to the animation of facial expressions for sign language avatars was proposed by Kacorri [1].

Furthermore, it is interesting to note that the challenges of animating SL avatars may be similar to the issues raised by the generation of communicative gestures for Embodied Conversational Agents (ECA), or more broadly for expressive virtual characters. Indeed, as with SL avatars, ECA gestures convey meaning and must demonstrate non-verbal behaviours such as shrugs or head movements that can be associated with the production of a speech utterance [124]. Consequently, ECA's gestures, like those of SL, must be precise, realistic, and expressive. Different challenges have been identified in the ECA community for deictic [125] or metaphoric gestures (use of the form and motion of a gesture to convey abstract concepts [126]), or more broadly for expressive virtual agent gestures guided by semantics [127]. Such challenges are very similar to those of the SL animation community.

Finally, as sign language avatars are mainly intended to be used by the Deaf, their approval by the community is necessary. A deaf person will be badly receptive to a robotic motion the same way a hearing person will to a robotic, unnatural voice. However, the quality of an avatar's motions is not the only factor guaranteeing the acceptance of the avatar: the avatar's appearance is also important. One of the main issues of the human-looking avatars is the risk of falling into the uncanny valley, first introduced by Mori in [128]. To prevent this risk, some research teams on sign language avatars choose to give a cartoon-like appearance to their virtual signers (e.g., avatar of Sign360 [90] or Adamo et al. [115]). 'Cartoon' avatars are more easily accepted by the deaf population due to their likable appearance but they bring specific problems (e.g., the gap between the proportions of the signer skeleton and of the $3D$ cartoon avatar can impact the accuracy of the animations). Surveys assessing the acceptance of sign language avatars by deaf people have been made by Kipp et al. [112], by Adamo-Villani et al. [129], and by Lu et al. [88, 82]. They show that non-manual features like facial expressions and natural movements are of great importance to deaf users. Such surveys provide precious insights on the acceptance of the sign language avatars (e.g., the choice of the colors for the avatar clothes, the presence of

shadows and the absence of mesh interpenetration are very important to improve the comprehensibility of the sign languages animations). Perceptual evaluation of signing avatar animations by deaf consumers should therefore be performed systematically to ensure a good response to the technology.

References

[1] Kacorri, H. Tr-2015001: A survey and critique of facial expression synthesis in sign language animation; 2015. CUNY Academic Works. https://academicworks.cuny.edu/gc_cs_tr/403.

[2] Woodward, J. How you gonna get to heaven if you can't talk with Jesus: On depathologizing deafness. TJ Publishers; 1982.

[3] Millet, A, Morgenstern, A. Grammaire descriptive de la langue des signes française: dynamiques iconiques et linguistique générale. UGA Editions; 2019.

[4] McKee, D, Kennedy, G. Lexical comparison of signs from american, australian, british and new zealand sign languages. The signs of language revisited: An anthology to honor Ursula Bellugi and Edward Klima 2000;:49–76.

[5] Zeshan, U. Interrogative constructions in signed languages: Crosslinguistic perspectives. Language 2004;:7–39.

[6] Martinet, A. La double articulation linguistique. Travaux du Cercle linguistique de Copenhague 1949;5(30-37).

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

[8] Battison, R. Lexical borrowing in American sign language. ERIC; 1978.

[9] Cuxac, C. La langue des signes française (LSF) : les voies de l'iconocité (French) [French Sign Language: the iconicity ways]. Faits de langues; Ophrys; 2000. ISBN 9782708009523. URL: https://books.google.fr/books?id=UuS7AAAAIAAJ.

[10] Boutora, L. Vers un inventaire ordonné des configurations manuelles de la langue des signes française. In: Journées d'Études sur la Parole (JEP). 2006, p. 12–16.

[11] Naert, L, Larboulette, C, Gibet, S. Annotation automatique des configurations manuelles de la Langue des Signes Française à partir de données capturées. In: Journees Françaises d'Informatique Graphique. Rennes, France; 2017,URL: https://hal.archives-ouvertes.fr/hal-01649769.

[12] Moody, B. La langue des signes, Tome 1 : Histoire et grammaire (French) [French Sign Language - First Volume: History and grammar]. International Visual Theatre (IVT); 1983.

[13] Battison, R. Phonological deletion in american sign language. Sign language studies 1974;5(1):1–19.

[14] Millet, A. La langue des signes française (lsf): une langue iconique et spatiale méconnue. Recherche et pratiques pédagogiques en langues de spécialité Cahiers de l'Apliut 2004;23(2):31–44.

[15] Sandler, W, Lillo-Martin, D. Sign language and linguistic universals. Cambridge University Press; 2006.

[16] Huenerfauth, M, Lu, P, Kacorri, H. Synthesizing and evaluating animations of american sign language verbs modeled from motion-capture data. 6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT) 2015;:22–28.

[17] Paabo, R, Foedisch, M, Hollman, L. Rules for estonian sign language transcription. Trames-journal of The Humanities and Social Sciences - TRAMES-J HUMANIT SOC SCI 2009;13. doi:10.3176/tr.2009.4.05.

[18] Wolfe, R, Cook, P, McDonald, JC, Schnepp, J. Linguistics as structure in computer animation: Toward a more effective synthesis of brow motion in american sign language. Sign Language & Linguistics 2011;14(1):179–199.

[19] Sallandre, M. Simultaneity in french sign language discourse. Amsterdam Studies in the Theory and History of Linguistic Science Series 4 2007;281:103.

[20] De Saussure, F. Cours de linguistique générale; vol. 1. Otto Harrassowitz Verlag; 1916.

[21] Ebling, S. Automatic translation from german to synthesized swiss german sign language. Ph.D. thesis; University of Zurich; 2016.

[22] Morrissey, S. Data-driven machine translation for sign languages. Ph.D. thesis; Dublin City University; 2008.

[23] Grosvald, M. Interspeaker variation in the extent and perception of long-distance vowel-to-vowel coarticulation. Journal of Phonetics 2009;37(2):173 – 188. URL: http://www.sciencedirect.com/science/article/pii/S0095447009000035. doi:https://doi.org/10.1016/j.wocn.2009.01.002.

[24] Naert, L, Larboulette, C, Gibet, S. Coarticulation analysis for sign language synthesis. In: International Conference on Universal Access in Human-Computer Interaction. Springer; 2017, p. 55–75.

[25] Gibet, S, Lebourque, T, Marteau, PF. High-level specification and animation of communicative gestures. Journal of Visual Languages & Computing 2001;12(6):657–687.

[26] Prillwitz, S, für Deutsche Gebärdensprache und Kommunikation Gehörloser, HZ. HamNoSys: Version 2.0; Hamburg Notation System for Sign Languages; An Introductory Guide. Signum-Verlag; 1989.

[27] Othman, A, Jemni, M. Statistical sign language machine translation: from english written text to american sign language gloss. arXiv preprint arXiv:11120168 2011;.

[28] Gibet, S, Lefebvre-Albaret, F, Hamon, L, Brun, R, Turki, A. Interactive editing in french sign language dedicated to virtual signers: requirements and challenges. Universal Access in the Information Society 2016;15(4):525–539.

[29] Braffort, A, Bolot, L, Filhol, M, Verrecchia, C. Démonstrations d'Elsi, la signeuse virtuelle du LIMSI. In: Conférence sur le Traitement Automatique des Langues Naturelles (TALN), Traitement automatique des langues des signes (atelier TALS). Toulouse, France; 2007,URL: https://hal.archives-ouvertes.fr/hal-02285133.

[30] Friedman, L. Phonology of a soundless language: phonological structure of the american sign language. Ph.D. thesis; UC Berkeley; 1976.

[31] Bébian, A. Mimographie, ou Essai d'écriture mimique propre à régulariser le langage des sourds-muets. L. Colas; 1825.

[32] Stokoe, WC, Casterline, DC, Croneberg, CG. A dictionary of American Sign Language on linguistic principles. Linstok Press; 1976.

[33] Sutton, V. Sign writing for everyday use. Sutton Movement Writing Press; 1981.

[34] Kato, M. A study of notation and sign writing systems for the deaf. Intercultural Communication Studies 2008;17(4):97–114.

[35] Lessa-de Oliveira, ASC. Libras escrita: o desafio de representar uma língua tridimensional por um sistema de escrita linear. ReVEL 2012;10(19).

[36] Liddell, SK, Johnson, RE. American sign language: The phonological base. Sign language studies 1989;64(1):195–277.

[37] Johnson, RE, Liddell, SK. A segmental framework for representing signs phonetically. Sign Language Studies 2011;11(3):408–463.

[38] Johnson, RE, Liddell, SK. Toward a phonetic representation of hand configuration: The fingers. Sign Language Studies 2011;12(1):5–45.

[39] Johnson, RE, Liddell, SK. Toward a phonetic representation of hand configuration: The thumb. Sign Language Studies 2012;12(2):316–333.

[40] Tkachman, O, Hall, KC, Xavier, A, Gick, B. Sign language phonetic annotation meets phonological corpustools: Towards a sign language toolset for phonetic notation and phonological analysis. In: Proceedings of the Annual Meetings on Phonology; vol. 3. 2016,.

[41] Eccarius, P, Brentari, D. Handshape coding made easier: A theoretically based notation for phonological transcription. Sign Language & Linguistics 2008;11(1):69–101.

[42] Glauert, J, Elliott, R. Extending the sigml notation—a progress report. In: Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT); vol. 23. 2011,.

[43] Losson, O. Modélisation du geste communicatif et réalisation d'un signeur virtuel de phrases en langue des signes grançaise. Theses; Université des Sciences et Technologie de Lille - Lille I; 2000. URL: https://tel.archives-ouvertes.fr/tel-00003332.

[44] Filhol, M. Modèle descriptif des signes pour un traitement automatique des langues des signes. Ph.D. thesis; Université Paris; 2008.

[45] Kipp, M, Heloir, A, Nguyen, Q. Sign language avatars: Animation and comprehensibility. In: Proceedings of the 10th International Conference on Intelligent Virtual Agents. 2011,.

[46] da Rocha Costa, AC, Dimuro, GP. Signwriting-based sign language processing. In: International Gesture Workshop. Springer; 2001, p. 202–

205.
[47] Kennaway, R. Avatar-independent scripting for real-time gesture animation. arXiv preprint arXiv:150202961 2006;.
[48] Kennaway, R. Experience with and Requirements for a Gesture Description Language for Synthetic Animation. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003, p. 300–311.
[49] do Amaral, WM, De Martino, JM, Angare, LMG. Sign language 3d virtual agent. In: Education and Information Systems, Technologies and Applications Conference. 2011,.
[50] De Martino, JM, Silva, IR, Bolognini, CZ, Costa, PDP, Kumada, KMO, Coradine, LC, et al. Signing avatars: making education more inclusive. Universal Access in the Information Society 2016;:1–16.
[51] Lebourque, T, Gibet, S. High level specification and control of communication gestures: the gessyca system. In: Computer Animation, 1999. Proceedings. IEEE; 1999, p. 24–35.
[52] Heloir, A, Kipp, M. Real-time animation of interactive agents: Specification and realization. Applied Artificial Intelligence 2010;24(6):510–529.
[53] Autodesk, . Maya. https://www.autodesk.fr/products/maya/overview; ???? Accessed: 2019-11-12.
[54] Blender, . Blender. https://www.blender.org/; ???? Accessed: 2019-11-12.
[55] Shantz, M, Poizner, H. A computer program to synthesize american sign language. Behavior Research Methods & Instrumentation 1982;14(5):467–474. URL: http://dx.doi.org/10.3758/BF03203314. doi:10.3758/BF03203314.
[56] Losson, O, Vannobel, JM. Sign language formal description and synthesis. In: Proc. 2. Euro. Conf. Disability, Virtual Reality & Assoc. Tech., Skövde, Sweden. 1998,.
[57] McDonald, J, Wolfe, R, Schnepp, J, Hochgesang, J, Jamrozik, DG, Stumbo, M, et al. An automated technique for real-time production of lifelike animations of american sign language. Universal Access in the Information Society 2016;15(4):551–566. URL: http://dx.doi.org/10.1007/s10209-015-0407-2. doi:10.1007/s10209-015-0407-2.
[58] McDonald, J, Wolfe, R, Wilbur, RB, Moncrief, R, Malaia, E, Fujimoto, S, et al. A new tool to facilitate prosodic analysis of motion capture data and a data-driven technique for the improvement of avatar motion. In: Proceedings of Language Resources and Evaluation Conference (LREC). 2016, p. 153–59.
[59] Lombardo, V, Battaglino, C, Damiano, R, Nunnari, F. An avatar-based interface for the italian sign language. In: Complex, Intelligent and Software Intensive Systems (CISIS), 2011 International Conference on. IEEE; 2011, p. 589–594.
[60] Lombardo, V, Nunnari, F, Damiano, R. A virtual interpreter for the italian sign language. In: International Conference on Intelligent Virtual Agents. Springer; 2010, p. 201–207.
[61] Ségouat, J. Modélisation de la coarticulation en langue des signes française pour la diffusion automatique d'informations en gare ferroviaire à l'aide d'un signeur virtuel. (French) [Modelling coarticulation in lsf for automatic broadcast of information in train stations using an avatar]. Ph.D. thesis; Université Paris Sud - Paris XI; 2010.
[62] Irving, A, Foulds, R. A parametric approach to sign language synthesis. In: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility. ACM; 2005, p. 212–213.
[63] Yorganci, R, Kindiroglu, AA, Kose, H. Avatar-based sign language training interface for primary school education 2016;.
[64] Grieve-Smith, AB. Signsynth: A sign language synthesis application using web3d and perl. In: Wachsmuth, I, Sowa, T, editors. Gesture and Sign Language in Human-Computer Interaction. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-47873-7; 2002, p. 134–145.
[65] Papadogiorgaki, M, Grammalidis, N, Tzovaras, D, Strintzis, MG. Text-to-sign language synthesis tool. In: Signal Processing Conference, 2005 13th European. IEEE; 2005, p. 1–4.
[66] Krňoul, Z, Kanis, J, Železnỳ, M, Müller, L. Czech text-to-sign speech synthesizer. In: International Workshop on Machine Learning for Multimodal Interaction. Springer; 2007, p. 180–191.
[67] Fotinea, SE, Efthimiou, E, Caridakis, G, Karpouzis, K. A knowledge-based sign synthesis architecture. Universal Access in the Information Society 2008;6(4):405–418.
[68] VCom3D, . Sign 4 me. http://www.vcom3d.com/language/sign-4-me/; 2018. Accessed: 2019-09-08.
[69] Delorme, M, Filhol, M, Braffort, A. Animation generation process for sign language synthesis. In: Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on. IEEE; 2009, p. 386–390.
[70] Losson, O, Vannobel, JM. Sign specification and synthesis. In: International Gesture Workshop. Springer; 1999, p. 239–251.
[71] Gibet, S, Marteau, P. A self-organized model for the control, planning and learning of nonlinear multi-dimensional systems using a sensory feedback. Applied Intelligence 1994;4(4):337–349.
[72] Lebourque, T. Spécification et génération de gestes naturels. Ph.D. thesis; Paris 11; 1998.
[73] Aubry, M, Julliard, F, Gibet, S. Modeling joint synergies to synthesize realistic movements. In: Gesture Workshop, Springer. 2009, p. 231–242.
[74] Kennaway, R. Synthetic animation of deaf signing gestures. In: International Gesture Workshop. Springer; 2001, p. 146–157.
[75] Smith, R, Morrissey, S, Somers, H. Hci for the deaf community: Developing human-like avatars for sign language synthesis 2010;.
[76] McDonald, J, Wolfe, R, Johnson, S, Baowidan, S, Moncrief, R, Guo, N. An improved framework for layering linguistic processes in sign language generation: Why there should never be a brows tier. In: International Conference on Universal Access in Human-Computer Interaction. Springer; 2017, p. 41–54.
[77] Wolfe, R, McDonald, J, Schnepp, J, Toro, J. Synthetic and acquired corpora: Meeting at the annotation. In: Workshop on Building Sign Language Corpora in North America, Washington, DC. 2011,.
[78] Yanovich, P, Neidle, C, Metaxas, D. Detection of major asl sign types in continuous signing for asl recognition. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). 2016, p. 3067–3073.
[79] Lin, JFS, Karg, M, Kulić, D. Movement primitive segmentation for human motion modeling: A framework for analysis. IEEE Transactions on Human-Machine Systems 2016;.
[80] Cheok, MJ, Omar, Z, Jaward, MH. A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics 2019;10(1):131–153.
[81] Braffort, A. Le regard et les mains: Annotation et analyse multipistes d'un corpus de lsf. Actes des 9èmes Journées Internationales de la Linguistique de corpus 2017;:14.
[82] Lu, P, Huenerfauth, M. Collecting and evaluating the cuny asl corpus for research on american sign language animation. Computer Speech & Language 2014;28(3):812–831.
[83] Gibet, S, Courty, N, Duarte, K, Le Naour, T. The signcom system for data-driven animation of interactive virtual signers: Methodology and evaluation. Transactions on Interactive Intelligent Systems 2011;.
[84] Gibet, S. Building french sign language motion capture corpora for signing avatars. In: Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, LREC 2018. Miyazaki, Japan; 2018,.
[85] Kapadia, M, Chiang, Ik, Thomas, T, Badler, NI, Kider Jr, JT, et al. Efficient motion retrieval in large motion databases. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM; 2013, p. 19–28.
[86] Awad, C, Courty, N, Duarte, K, Le Naour, T, Gibet, S. A combined semantic and motion capture database for real-time sign language synthesis. In: Proceedings of the 9th International Conference on Intelligent Virtual Agents; vol. 5773 of *Lecture Notes in Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag; 2009, p. 432–38.
[87] Hamon, L, Gibet, S, Boustila, S. Interactive editing of utterances in french sign language dedicated to signing avatars (édition interactive d'énoncés en langue des signes française dédiée aux avatars signeurs) [in french]. In: Morin, E, Estève, Y, editors. Traitement Automatique des Langues Naturelles, TALN 2013, Les Sables d'Olonne, France, 17-21 Juin 2013, articles courts. The Association for Computer Linguistics; 2013, p. 547–554. URL: https://www.aclweb.org/anthology/F13-2006/.
[88] Lu, P, Huenerfauth, M. Collecting a motion-capture corpus of american sign language for data-driven generation research. In: Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies. Association for Computational Linguistics; 2010, p. 89–97.
[89] Brun, R, Turki, A, Laville, A. A 3d application to familiarize chil-

dren with sign language and assess the potential of avatars and motion capture for learning movement. In: Proceedings of the 3rd International Symposium on Movement and Computing. ACM; 2016, p. 48.

[90] MoCapLab, , de France, RI, Digital, C. Sign 360. http://www.mocaplab.com/fr/projects/sign-360/; 2016. Accessed: 2018-09-14.

[91] Witkin, A, Popovic, Z. Motion warping. In: Siggraph; vol. 95. 1995, p. 105–108.

[92] Héloir, A, Gibet, S. A qualitative and quantitative characterisation of style in sign language gestures. In: Gesture Workshop. 2007,.

[93] Héloir, A, Courty, N, Gibet, S, Multon, F. Temporal alignment of communicative gesture sequences. Computer Animation and Virtual Worlds 2006;17:347–357.

[94] Gibet, S, Héloir, A, Courty, N, Kamp, JF, Gorce, P, Rezzoug, N, et al. Virtual agent for deaf signing gestures. AMSE, Journal of the Association for the Advancement of Modelling and Simulation Techniques in Enterprises (Special edition HANDICAP) 2006;67:127–136.

[95] Carreno, P, Gibet, S, Marteau, PF. Synthèse de mouvements humains par des méthodes basées apprentissage: un état de l'art. Revue Electronique Francophone d'Informatique Graphique 2014;8(1).

[96] Cox, S, Lincoln, M, Tryggvason, J, Nakisa, M, Wells, M, Tutt, M, et al. The development and evaluation of a speech-to-sign translation system to assist transactions. International Journal of Human–Computer Interaction 2003;16(2):141–161. doi:10.1207/S15327590IJHC1602_02.

[97] Pezeshkpour, F, Marshall, I, Elliott, R, Bangham, JA. Development of a legible deaf-signing virtual human. In: Proceedings IEEE International Conference on Multimedia Computing and Systems; vol. 1. 1999, p. 333–338.

[98] Alexanderson, S, Beskow, J. Towards fully automated motion capture of signs–development and evaluation of a key word signing avatar. ACM Transactions on Accessible Computing (TACCESS) 2015;7(2):7.

[99] Aouiti, N, Jemni, M, Semreen, S. Arab gloss annotation system for arabic sign language. In: 2015 5th International Conference on Information Communication Technology and Accessibility (ICTA). 2015, p. 1–6. doi:10.1109/ICTA.2015.7426932.

[100] Ormel, E, Crasborn, O, van der Kooij, E, van Dijken, L, Nauta, E, Forster, J, et al. Glossing a multi-purpose sign language corpus. In: Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and sign language technologies. 2010, p. 186–191.

[101] López-Colino, F, Colás, J. The synthesis of lse classifiers: From representation to evaluation. Journal of Universal Computer Science 2011;.

[102] López-Colino, F, Colás, J. Spanish sign language synthesis system. Journal of Visual Languages & Computing 2012;23(3):121–136.

[103] Woods, WA. Transition network grammars for natural language analysis. Communications of the ACM 1970;13(10):591–606.

[104] Huenerfauth, M. Generating american sign language classifier predicates for english-to-asl machine translation. Ph.D. thesis; University of Pennsylvania; 2006.

[105] Bertoldi, N, Tiotto, G, Prinetto, P, Piccolo, E, Nunnari, F, Lombardo, V, et al. On the creation and the annotation of a large-scale italian-lis parallel corpus. In: LREC 2010. 2010,.

[106] Filhol, M, Falquet, G. Synthesising sign language from semantics, approaching" from the target and back". arXiv preprint arXiv:170708041 2017;.

[107] Nunnari, F, Filhol, M, Heloir, A. Animating azee descriptions using off-the-shelf ik solvers. In: 8th Workshop on the Representation and Processing of Sign Languages: Involving the Language Community (SignLang 2018), 11th edition of the Language Resources and Evaluation Conference (LREC 2018). 2018, p. 7–12.

[108] Filhol, M, Hadjadj, MN. Juxtaposition as a form feature; syntax captured and explained rather than assumed and modelled. In: Language resources and evaluation conference (LREC), Representation and processing of Sign Languages. 2016,.

[109] Filhol, M, McDonald, J, Wolfe, R. Synthesizing Sign Language by Connecting Linguistically Structured Descriptions to a Multi-track Animation System. Cham: Springer International Publishing. ISBN 978-3-319-58703-5; 2017, p. 27–40. URL: https://doi.org/10.1007/978-3-319-58703-5_3. doi:10.1007/978-3-319-58703-5_3.

[110] Filhol, M. A human-editable sign language representation for software editing—and a writing system? arXiv preprint arXiv:181101786 2018;.

[111] Kennaway, R, Glauert, JR, Zwitserlood, I. Providing signed content on the internet by synthesized animation. ACM Transactions on Computer-Human Interaction (TOCHI) 2007;14(3):15.

[112] Kipp, M, Nguyen, Q, Heloir, A, Matthes, S. Assessing the deaf user perspective on sign language avatars. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility. ASSETS '11; New York, NY, USA: ACM. ISBN 978-1-4503-0920-2; 2011, p. 107–114. URL: http://doi.acm.org/10.1145/2049536.2049557. doi:10.1145/2049536.2049557.

[113] Elliott, R, Glauert, JR, Kennaway, J, Marshall, I, Safar, E. Linguistic modelling and language-processing technologies for avatar-based sign language presentation. Universal Access in the Information Society 2008;6(4):375–391.

[114] Filhol, M, Mcdonald, J. Extending the AZee-Paula shortcuts to enable natural proform synthesis. In: Workshop on the Representation and Processing of Sign Languages. Miyazaki, Japan; 2018,URL: https://hal.archives-ouvertes.fr/hal-01848979.

[115] Adamo-Villani, N, Hayward, K, Lestina, J, Wilbur, RB. Effective animation of sign language with prosodic elements for annotation of digital educational content. In: SIGGRAPH Talks. 2010,.

[116] Elliott, R, Bueno, J, Kennaway, R, Glauert, J. Towards the integrationn of synthetic sl annimation with avatars into corpus annotation tools. In: 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, Valletta, Malta. 2010, p. 29.

[117] Hanke, T, Matthes, S, Regen, A, Storz, J, Worseck, S, Eliott, R, et al. Using timing information to improve the performance of avatars. In: Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT). 2011,.

[118] Marshall, I, Safar, E. Grammar development for sign language avatar-based synthesis. In: Proceedings HCII. 2005, p. 1–10.

[119] Sugandhi, PK, Kaur, S. Online multilingual dictionary using hamburg notation for avatar-based indian sign language generation system. Int J Cogn Lang Sci 2018;12(8):1116–1122.

[120] Filhol, M, McDonald, J, Wolfe, R. Synthesizing sign language by connecting linguistically structured descriptions to a multi-track animation system. In: International Conference on Universal Access in Human-Computer Interaction. Springer; 2017, p. 27–40.

[121] University, D. Depaul asl avatar project. http://asl.cs.depaul.edu; 2020. Accessed: 2020-01-28.

[122] Courty, N, Gibet, S. Why is the Creation of a Virtual Signer Challenging Computer Animation ? In: Motion in Games 2010. LNCS; Netherlands; 2010, p. 1–11. URL: https://hal.archives-ouvertes.fr/hal-00516624.

[123] Duarte, K. Motion capture and avatars as portals for analyzing the linguistic structure of sign languages. Ph.D. thesis; Université Bretagne Sud; 2012.

[124] Kipp, M, Neff, M, Kipp, KH, Albrecht, I. Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In: International Workshop on Intelligent Virtual Agents. Springer; 2007, p. 15–28.

[125] Hartmann, B, Mancini, M, Pelachaud, C. Implementing expressive gesture synthesis for embodied conversational agents. In: International Gesture Workshop. Springer; 2005, p. 188–199.

[126] Ravenet, B, Pelachaud, C, Clavel, C, Marsella, S. Automating the production of communicative gestures in embodied characters. Frontiers in Psychology 2018;9:1144. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2018.01144. doi:10.3389/fpsyg.2018.01144.

[127] Gibet, S, Carreno-Medrano, P, Marteau, P. Challenges for the animation of expressive virtual characters: The standpoint of sign language and theatrical gestures. In: Laumond, J, Abe, N, editors. Dance Notations and Robot Motion, 1st Workshop of the Anthropomorphic Motion Factory, at LAAS-CNRS, Toulouse, France, 13-14 November, 2014; vol. 111 of *Springer Tracts in Advanced Robotics*. Springer; 2014, p. 169–186.

[128] Mori, M. The uncanny valley (in japanese). In: Energy; vol. 7 of *LNCS*. Japan; 1970, p. 33–35.

[129] Adamo-Villani, N, Anasingaraju, S. Toward the ideal signing avatar. EAI Endorsed Transactions on e-Learning 2016;3(11). doi:10.4108/eai.15-6-2016.151446.

Table .3. Non-exhaustive list of the existing sign language avatars.

Year	Name of the avatar and/or the project	Input of the animation system (specification language...)	Sign synthesis technique	Utterance synthesis technique	Targeted applications
1982	Shantz Avatar [55]	List of fingers angles	**Keyframe**: hand-crafted	/	Study of SL perception
1999 & 2003	*SignAnim* [97] & Tessa [96]	Gloss sequence	**Data-driven**: play-back	Concatenative	Subtitle-to-SSE translation & Face-to-face **speech-to-sign translation**
1999-2001	*GessyCA* [51, 25]	QualGest	**Procedural**: with IK-based controllers	Articulatory	Animation of gestures for human communication
2001-2004	*ViSiCAST* [74]	SigML	**Procedural**: with dynamic controllers	Concatenative	Broadcasting, Face-to-face transactions and Interactive Internet info
2005	*Vsigns* [65]	SWML	**Keyframe**: automatic using kinematics	Concatenative	SL **dictionnary** and/or SL editor for newscast animation
2006	*Gérard* [94]	Gloss + biomechanic features	**Data-driven** : augmented database	Concatenative	Expressive animation of SL
2007	*Elsi* [29]	Video footage + gloss sequence	**Keyframe**: hand-crafted	Concatenative	Train station pre-recorded information
2007-2008	Czech signed speech avatar [66]	HamNoSys + NMFs	**Keyframe**: automatic using kinematics	Concatenative	**Speech-to-sign translation**
2008	Greek SL avatar [67]	HamNoSys + NMFs	**Keyframe**: automatic using kinematics	Concatenative	**Text-to-sign translation**
2009	LSF Avatar [69]	Zebedee	**Keyframe**: automatic using kinematics	Concatenative	Sign Synthesis
2010	EMBR Avatar [52]	EMBRScript	**Keyframe**: automatic using kinematics	Concatenative	Animation of Embodied Conversational Agent
2010-2016	*JASigning* [116, 21]	SigML	**Procedural**: with dynamic controllers	Concatenative	Multiplatform SL avatar, **Text-to-Sign translation**
2011	*SignCom* [83]	MoCap & Gloss Sequence	**Data-driven**: augmented database	Concatenative	Interactive assisting Internet technologies, **translation**
2011	Italian SL Avatar[59]	Gloss + modifiers	**Keyframe**: hand-crafted	Concatenative + modifiers	**Text-to-sign translation**, Virtual interpreter
2012	LSE Synthesizer [101, 102]	HLSML	**Keyframe**: hand-crafted	Concatenative + modifiers	Multimedia applications, inflected sign synthesis
2015	Huenerfauth [16]	Huenerfauth	**Data-driven**: machine learning	Concatenative	Directional verbs study
2016	Yorganci [63]	MoCap & phonological notation	**Keyframe**: hand-crafted	Concatenative	**Educational** tool for deaf children
2016	*Sign3D* [28]	MoCap & Gloss sequence	**Data-driven**: augmented database	Concatenative	Interactive assisting Internet technologies, **Translation**
2016	Libras Avatar [50]	XML version of Movement/Hold	**Data-driven/Keyframe**: MoCap keyframes	/	**Translation** of written textbooks for children
2016-now	*Paula* [57, 114]	SLPA/Azee	**Keyframe**: hand-crafted and automatic using kinematics	Hybrid	**Translation**

Three sign language avatars: Raymond from the IRISA lab (http://lsf.irisa.fr/), Paula of DePaul University (http://asl.cs.depaul.edu) and the cat of MoCapLab (https://www.mocaplab.com/fr/)