Information Sciences 281 (2014) 762–780

Contents lists available at ScienceDirect

Information Sciences

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

An approach to generate and embed sign language video tracks into multimedia contents

Tiago Maritan U. de Araújo *, Felipe L.S. Ferreira, Danilo A.N.S. Silva, Leonardo D. Oliveira, Eduardo L. Falcão, Leonardo A. Domingues, Vandhuy F. Martins, Igor A.C. Portela, Yurika S. Nóbrega, Hozana R.G. Lima, Guido L. Souza Filho, Tatiana A. Tavares, Alexandre N. Duarte

Digital Video Applications Lab (LAViD), Federal University of Paraíba, Paraíba, Brazil

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form 6 March 2014
Accepted 8 April 2014
Available online 19 April 2014

Keywords:
Accessible multimedia content
Brazilian sign language
Machine translation
Accessible technologies for the deaf
Sign synthesis

ABSTRACT

Deaf people have serious problems to access information due to their inherent difficulties to deal with spoken and written languages. This work tries to address this problem by proposing a solution for automatic generation and insertion of sign language video tracks into captioned digital multimedia content. Our solution can process a subtitle stream and generate the sign language track in real-time. Furthermore, it has a set of mechanisms that exploit human computation to generate and maintain their linguistic constructions. The solution was instantiated for the Digital TV, Web and Digital Cinema platforms and evaluated through a set of experiments with deaf users.

1. Introduction

Deaf people naturally communicate using sign languages. As a result, many of them have difficulties in understanding and communicating through texts in written languages. Since these languages are based on sounds, most of them spends several years in school and fail to learn to read and write the written language of their own country [36]. Reading comprehension tests performed by Wauters [40] with deaf children aged 7–20 in the Netherlands showed that only 25% of them read at or above the level of a nine-year-old hearing child. In Brazil, about 97% of the deaf people do not finish the high school [21].

In addition, the Information and Communication Technologies (ICT) rarely address the specific requirements and needs of deaf people [16]. The support for sign language, for example, is rarely addressed in the design of these technologies. On TV, for example, sign languages support is generally limited to a window with an interpreter presented along with the original video program (wipe). This solution has high operational costs for generation and production (cameras, studio, staff, etc.) and requires full time human interpreters, which reduces their presence to a small portion of the TV programming. Furthermore, this traditional approach is not feasible for platforms with dynamic contents such as the Web. These difficulties result in major barriers to communicate, to access information and to acquire knowledge.

* Corresponding author. Tel.: +55 8332167093.

E-mail addresses: maritan@lavid.ufpb.br (Tiago Maritan U. de Araújo), lacet@lavid.ufpb.br (F.L.S. Ferreira), danilo@lavid.ufpb.br (D.A.N.S. Silva), leodantas@lavid.ufpb.br (L.D. Oliveira), eduardolf@lavid.ufpb.br (E.L. Falcão), leonardo.araujo@lavid.ufpb.br (L.A. Domingues), vandhuy@lavid.ufpb.br (V.F. Martins), igor.portela@lavid.ufpb.br (I.A.C. Portela), yurika@lavid.ufpb.br (Y.S. Nóbrega), hozana@lavid.ufpb.br (H.R.G. Lima), guido@lavid.ufpb.br (G.L. Souza Filho), tatiana@lavid.ufpb.br (T.A. Tavares), alexandre@lavid.ufpb.br (A.N. Duarte), alexandre@lavid.ufpb.br (A.N. Duarte).

http://dx.doi.org/10.1016/j.ins.2014.04.008
0020-0255/

The scientific literature includes some works addressing the communication needs of the deaf [17,18,26,27,35]. These works offer technological solutions for daily activities enabling deaf people to watch and understand television, to interact with other people, to write a letter, among others. The use of dynamic [17,18] and emotive captioning [26] in movies and television programs and the development of games for training deaf children [27] are examples of this type of solution.

Other works deal with machine translation for sign languages [4,13,19,20,30–33,38,42]. Veale et al. [38], for example, proposed a multilingual translation system for translating English texts into Japanese Sign Language (JSL), American Sign Language (ASL) and Irish Sign Language (ISL). The work explores and extends some Artificial Intelligence (AI) concepts to sign languages (SL), such as, knowledge representation, metaphorical reasoning, among others [30], but there is no testing or experimentation to evaluate the solution. Then, it is not possible to draw conclusions about its feasibility and translation speed and quality.

Zhao et al. [42] developed an interlanguage-based approach for translating English text into American Sign Language (ASL). It analyses the input data to generate an intermediate representation (IR) from their syntactic and morphological information. Then, a sign synthesizer uses the IR information to generate the signs. However, as well as in Veale et al.'s work [38], the solution lacks experimental evaluation. Morrissey [30] proposed an example-based machine translation (EBMT) system for translating text into ISL. However, the data set was developed from a set of "children's stories", which restricts the translation for that particular domain.

Fotinea et al. [13] developed a system for translating Greek texts into Greek Sign Language (GSL). This work uses a transfer-based approach for generate the sentences in GSL, but it main focus is the strategy of animation that explores the parallel structures of sign languages (e.g., the ability to present a hand movement with a facial expression simultaneously). To perform this task, a 3D avatar was developed to explore the parallel structure of sign languages. However, no testing or experimentation was conducted to evaluate it translation speed and quality.

Huenerfauth et al. [19,20] proposed modeling classifiers predicate[1] in a English to American Sign Language (ASL) translation system. Some tests performed with deaf users showed that contents exploring the use of classifier predicates (generated by the Huenerfauth solution) were significantly more natural, grammatically correct and understandable than the contents based on direct translation. The translation speed, however, was not evaluated by author.

Anuja et al. [4] proposed a system for translating English speech into Indian Sign Language (ISL) focused on helping deaf people to interact in public places, such as banks and railroads. The system also uses a transfer-based approach for translating speech entries into ISL animations. This solution is restricted to a specific domain and according to authors it takes a long (and unacceptable) time to generate the translation (the time values, however, were not described in the work).

San-Segundo et al. [31–33] proposed an architecture for translating speech into Spanish Sign Language (LSE) focused on helping deaf people when they want to renew their identity card or driver's license. This translation system consists of three modules: a speech recognizer, a natural language translator and an animation module. However, as well as in Anuja, Suryapriya and Idicula work, this solution is also restricted to a particular (or specific) domain and the time needed for translating speech into LSE (speech recognition, translation and signing) is around 8 s per sentence, which makes the solution unfeasible for real time domains (e.g., television).

These works can be separated in two classes: one class of works that translate speech in the source spoken language to the target sign language (i.e., they use speech recognition) [4,31–33], and other class that translate written texts to the target sign language (i.e., they do not use speech recognition) [13,19,20,30,38,42]. However, all these works have some limitations. The class of works that use speech recognition [4,31–33], for example, are just applied to specific domains and are not efficient considering signing and translation speed. Other works do not have an assessment of the feasibility and quality of the solution [13,38,42] or are also applied to specific domains [19,20,30]. These limitations reduce their applicability to real-time and open-domain scenarios, such as TV.

Another difficulty is that the development of their linguistic constructions (translation rules, signs dictionary, etc.) is in general a non-trivial task and requires much manual work. Moreover, as sign languages are natural and living languages, new signs and new grammatical constructions can arise spontaneously over time. This implies that these new signs and constructions must also be included in the solution, otherwise the quality of content generated by it tend to deteriorate over time, making it outdated.

To reduce these problems, in this paper, we propose a solution to generate and embed sign language video tracks in multimedia contents. Our current implementation targets the Brazilian Sign Language (LIBRAS), but we believe that the general solution can be extended for other target sign languages. The LIBRAS video tracks are generated from the translation of subtitle tracks in Brazilian Portuguese and are embedded in the multimedia content as an extra layer of accessible content.

A 3D avatar reproduces the signs and the solution also explores human computation strategies to allow human collaborators to generate and maintain their linguistic constructions (translation rules and signs). The implementation utilizes also a set of optimization strategies, such as a textual machine translation strategy for Brazilian Portuguese to gloss (a LIBRAS textual representation), which consumes little computational time, and LIBRAS dictionaries to avoid rendering the signs in real time, reducing the computational resources required to generate the LIBRAS video.

[1] Classifiers are linguistic phenomena used by sign language interpreters to make the signs more natural and easier to understand. They make use of the space around the signer in a topologically meaningful way. The interpreter's hands represent an imaginary entity in space in front of them, and they position, move, trace our re-orient this imaginary object to indicate location, movement, shape, among others [19].

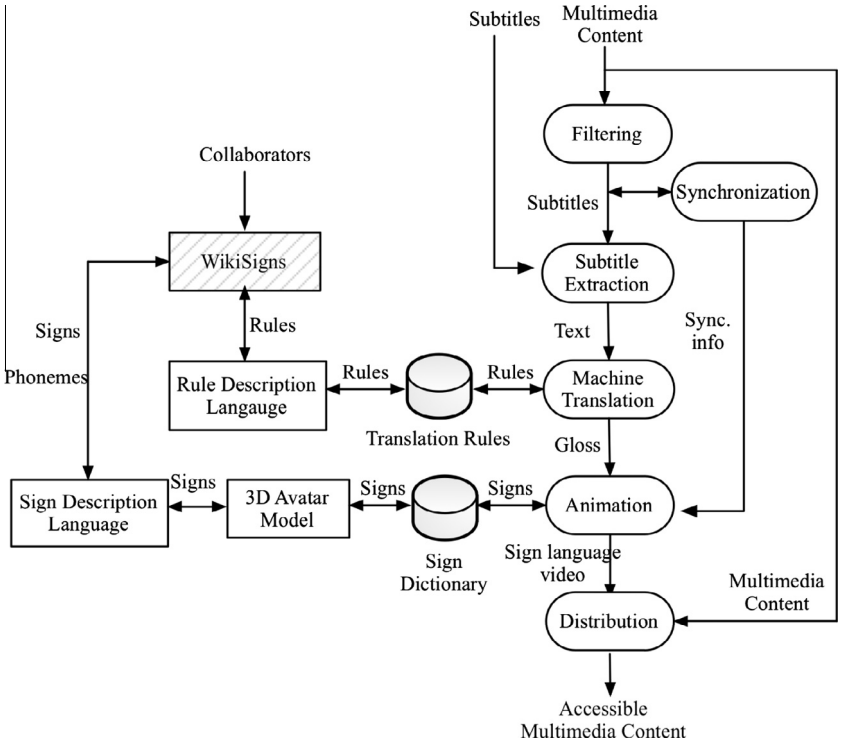

Fig. 1. Schematic view of the proposed solution.

Furthermore, we also developed prototypes for three different platforms (Digital TV, Web and Digital Cinema) and conducted a series of experiments with Brazilian deaf users in order to evaluate the solution.

As also mentioned by Kenaway et al. [24], it is important to point out that we do not intend to replace human interpreters, since the quality of machine translation and virtual signing are still not close to the quality of human translation and signing [25]. The idea is to develop a complementary, practical, high speed and low cost solution that can be used, for example, to provide information for the deaf in different platforms, especially when human interpreters are not feasible or are not available.

The rest of this paper is organized as follows. In Section 2, we describe the proposed solution. In Section 3, we describe the prototypes of the proposed solution developed for Digital TV, Web and Digital Cinema platforms. Some tests to evaluate the proposed solution are described in Section 4. Final remarks are given in Section 5.

2. The proposed solution

This section describes the architecture of the proposed solution and its software components. As mentioned in Section 1, the solution consists of a set of software components responsible for generating and embedding sign language video tracks on captioned multimedia contents through an automatic translation of subtitle tracks.

A schematic view of the proposed solution is illustrated in Fig. 1. Initially, a Filtering component extracts the subtitle tracks from the captioned multimedia contents.[2] Then a Subtitle Extraction component converts this subtitle stream (or file) into a sequence of words in the source textual language. The Machine Translation component maps this sequence of words into a sequence of language glosses (i.e., a text in the grammatical structure of the target sign language). Afterwards, an Animation component associates each gloss with the video of the corresponding sign in the Dictionary. Thus, the sequence of glosses is mapped to a sequence of sign videos that are synchronized with the subtitle track to generate the sign language track. Finally, a Distribution component embeds this sign language track into the original multimedia content, making it accessible for the deaf.

Two important features of this solution are the utilization of glosses as an intermediary representation between the source textual language and the target sign language and the usage of a Sign Dictionary to minimize the computational resources required to generate the sign language tracks in real time. The Sign Dictionaries are used to avoid the rendering of signs in real time, since this task is very time consuming. These dictionaries store pre-rendered video signs and each sign has a corresponding code (e.g., the gloss textual representation). Thus, it is possible to generate a sign language video from a smooth combination of signs in the Sign Dictionary.

[2] Optionally, a subtitle file (or stream) can be loaded directly into the solution.

Another important aspect of the solution is the usage of human computing strategies to conceive its linguistic constructions (i.e., translation rules and signs) in a semi-automatic and efficient way. The idea is that sign language specialists collaborate in the generation of these constructions and also improve the quality of content generated by improving the translation rules, including new signs, etc. To do this task, a human computation tool called WikiSigns has been developed and included in the solution, along with formal languages for describing translation rules (Rule Description Language) and signs (Signs Description Languages), and the model of a 3D avatar.

The synchronization between the original multimedia content and the sign language video is performed using the time axis synchronization model [6]. This model defines synchronization points that are inserted into the content using timestamps based on a global clock. In the proposed solution, the global clock is the clock of the subtitle track and it is used to generate the presentation timestamps of the sign language video track. The Machine Translation, Animation and Distribution components are presented in more details in the next subsections.

2.1. Machine Translation component

The Machine Translation component converts the source textual representation into a sign language textual representation (sequence of glosses). As mentioned earlier, this strategy was designed to translate contents efficiently (i.e., consuming little time) and for general domains. To perform this task, it combines statistical compression methods used to classify the tokens (words), simplification strategies to reduce the complexity of the input text and a set of morphological and syntactic rules defined by sign language specialists (i.e., it is a rule-based approach).

Initially, the source text is split into a sequence of tokens. Afterwards, these tokens are classified into morphological and syntactic classes using PPM-C [28], a variant of the Prediction by Partial Matching (PPM) algorithm[3] [8].

After classifying the tokens, a simplification strategy is applied to reduce the complexity of the input text. Initially, the text is simplified by removing some classes of tokens that are not defined in the target sign language (e.g., the Brazilian sign language does not contains articles and prepositions). In addition, some tokens are replaced (lexical replacement) to adapt the meaning of the sentence rewritten for the sign language. For example, the words home, house, habitation in Brazilian Portuguese are represented by the same sign in LIBRAS, the HOME sign. Furthermore, while the Brazilian Portuguese verbs have a high degree of inflection, the LIBRAS verbs do not inflect [39]. Then, the Brazilian Portuguese verbs would be replaced by non-inflected gloss verbs (i.e., the LIBRAS verbs). To do this replacement, we use a set of source language to sign language synonyms (e.g., a Brazilian Portuguese to LIBRAS Dictionary).

Proper names and technical terms are spelled in the sign language by handshapes that represent each letter in the word. Thus, the simplification strategy also applies a dactylology replacement to spell proper names and technical terms.

Finally, a set of translation rules is applied to translate the remaining tokens for a textual representation in the sign language. These translation rules are loaded from a translation rules database. The translation rules in these databases are described using a proposed formal language called Rule Description Language, which will be presented in Section 2.4.2.

2.2. Animation component

The Animation component is responsible for converting the sequence of glosses generated by the Machine Translation component into a sign language video. To perform this task, it uses a Sign Dictionary containing a video file for each sign in the language. Thus, it can be formally defined as a set of t tuples in the following format:

$$t = \langle g, v \rangle, \tag{1}$$

where g and v are, respectively, the gloss and the video for one sign. The video can be recorded with an interpreter or generated from a virtual animated agent (an avatar). However, the use of video recorded with interpreters has some problems. A major problem is that to compose sentences from the combination of independent signs (videos) it would be necessary to record all videos with the same interpreter under the same conditions (i.e., the same clothes, lighting, distance to camera, among others). Otherwise, the transition between consecutive signs would be not smooth enough [10].

Another problem is related to the dictionary's update. Since sign languages are living languages and new signs can arise spontaneously, it would be necessary to record new videos for these new signs with the same interpreter and under the same conditions of the previous signs. Furthermore, the manual generation of this dictionary is a very time consuming task.

To avoid these problems, in the proposed solution signs are represented by a 3D virtual animated agent (a 3D avatar). Thus, it is possible to generate all video (signs) under the same conditions and to update the dictionary more easily. Furthermore, in the proposed solution, the signs can be developed in a more productive way by using a human computation tool (WikiSigns). To perform this task, a Sign Description Language was developed, allowing deaf and sign language specialists to describe signs in WikiSigns. From this description, the signs can be rendered using the proposed 3D avatar model. The WikiSigns tool will be presented in Section 2.4.

[3] PPM builds a statistical model from a set of input data (training set) and uses this model to store the frequency of different sequences of elements found. After the model is built, the next element of the sequence can be predicted according to its previous N elements. The PPM-C variant is more efficient than the original implementation in terms of running time and data space exchange for marginally inferior compression [28].

Initially, the Animation component receives a sequence of glosses. Then, it retrieves the corresponding video for each gloss in the sequence from the Sign Dictionary. If no entry is found in the Dictionary for a given sign in the sentence, a video is then generated from the combination of it gloss letters (i.e., the sign is spelled). This strategy is used in order to avoid gaps in the representation of the sign language sentences and is the same strategy used by the sign language users to represent words or terms that do not have their own signs, such as proper names and technical terms.

After retrieving the videos signs from the Dictionary, the Animation component combines these videos to generate a single sign language video stream (sign synthesis). This strategy concatenates the videos based on the timestamps generated by the Synchronization component. A neutral configuration (i.e., 3D avatar position, background color, brightness, etc.) was defined at the start and end of each sign (video) to allow a smooth transition between consecutive signs (video signs). In addition, a video with the 3D avatar in the neutral position was also generated to be included during silent intervals.

As mentioned earlier, the proposed solution uses the time axis synchronization model [6] to synchronize the sign language track with the original multimedia content, where the global clock of multimedia content is used as a reference for generating the presentation timestamps (PTS) which works as synchronization points.

Finally, the Animation component send the sign language track to the Distribution component, which will embed it into the original multimedia content.

2.3. Distribution component

As mentioned earlier, the Distribution component is responsible for embedding the sign language video track generated by the Animation component into the original multimedia content. This distribution may be performed in three different ways according to the features provided by the target platform:

- **Mixing:** The sign language video frames are displayed in a window over the frames of the multimedia content, making the presentation of the sign language video track independent of the video player. However, one problem of this approach is that after the mixing process is performed, it is no longer possible to disable or remove the sign language video window.
- **Multiplexing:** The sign language video track is coded as a separate and independent video stream which will be encapsulated together with the input multimedia content into a single encapsulated transport stream (e.g., using the MPEG-2 Transport Stream protocol [22]). Thus, we have one single Transport Stream containing two video tracks. This approach makes the presentation of the sign language video track dependent of a player able to play both videos at the same time. On the other hand, it is possible to enable, disable, resize or reposition the sign language video.
- **Secondary screen:** In this case, the sign language video is displayed on a secondary screen. This approach is interesting in environments that are shared between deaf and non-deaf users, such as a Cinema, where the sign language window could disrupt non-deaf users. In this case, it would be possible to transmit the sign language video to be displayed on an user-specific device (e.g., their smartphones or tablets). According to ABNT NBR 15290 [1], the Brazilian specification for accessibility on TV, delays of up to four seconds are tolerated in live closed captioning systems. As a result, it is possible to assume a similar tolerable delay in sign language transmission systems and thus the expected transmission delay for the second video screen should not be a problem.

2.4. Human computation strategy

This section presents the strategy used to conceive the linguistic constructions (or linguistic contents) of the proposed solution in an efficient way. This strategy is composed by a human computational tool, called WikiSigns, which controls the generation of these constructions (contents); formal languages to describe translation rules and signs; and the model of a 3D virtual animated agent (a 3D humanoid avatar) used to represent the signs in the solution. In Section 2.4.1 we describe the architecture of the WikiSigns tool. The translation rule description language, the sign description language and the model of 3D are detailed in the Sections 2.4.2–2.4.4, respectively.

2.4.1. WikiSigns

As mentioned earlier, the objective of WikiSigns is that human collaborators can generate the translation rules and Sign Dictionary of the proposed solution in a semi-automatic way. To perform this task, the WikiSigns is composed by a set of modules (or components) responsible for generating translation rules and signs. Fig. 2 illustrates a schematic view of WikiSigns.

Initially, human collaborators access WikiSigns through a Web interface. From this interface, they can configure new signs or translation rules or edit the existing ones. When the user configures a new translation rule, the Rule Description Generator module records the user interaction in a XML representation, according to the Rule Description Language (described in the Section 2.4.2). This XML representation is then stored on a temporary database to be approved by sign language specialists, i.e., a supervision stage is applied before insertion into the database, which prevents the publication of incorrect rules. In addition, users can search and edit existing rules. Edited rules are considered as new rules and submitted to the same supervision stage.

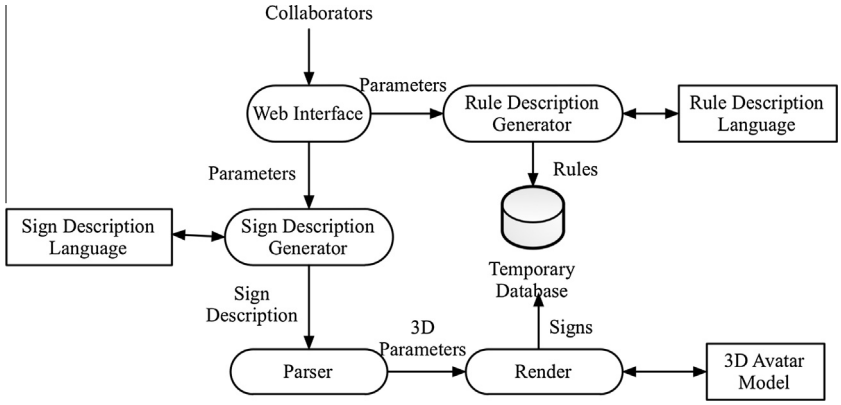

Fig. 2. Schematic view of WikiSigns.

When the user configures a new sign, a Sign Description Generator module converts the users' interactions into XML, according to the Sign Description Language (described in the Section 2.4.3). Afterwards, this XML representation is converted by the Parser module for a set of parameters based on the model of the 3D avatar (described in the Section 2.4.4) and the Renderer module renders a video for the new sign from these parameters. The video of the new sign is then returned to the user that can assess if it was generated correctly. Upon approval of users, a supervision stage is also applied over it before entering the Sign Dictionary.

In the next section we will describe implementations of this solution for Digital TV, Web and Digital Cinema platforms, developed as usage scenarios for the proposed solution. All three implementation used Brazilian Portuguese as input language and the Brazilian sign language (LIBRAS) as output.

2.4.2. Rule description language

The Rule Description Language is used to describe the translation rules that will be applied by the Machine Translation component. In this language, each translation rule is defined as a r tuple in the following format:

$$r = \langle e_1, e_2, \ldots, e_c \rangle, \tag{2}$$

where $e_1, e_2, \ldots, e_n$ is a set linguistic elements ordered according to the input sequence and c is the number of linguistic elements. The linguistic elements e_i are defined as

$$e_i = \langle ms_{class}, n_{pos}, n_{prop} \rangle, \quad i = 1, 2, \ldots, c \tag{3}$$

where ms_{class} identify it morphologically or syntactically class. n_{pos} indicates the new positioning of the element after the rule is applied with a value of -1 meaning that the element must be removed. n_{prop} is a optional field which indicates possible changes in the element (e.g., every verb in LIBRAS must be in the infinitive form).

```
<rule>
    <count>3</count>
    <class>
        <title>SUBJ</title>
        <newpos>1</newpos>
    </class>
    <class>
        <title>VERB</title>
        <newpos>2</newpos>
        <newproperty>inf</newproperty>
    </class>
    <class>
        <title>OBJ</title>
        <newpos>0</newpos>
    </class>
</rule>
```

Fig. 3. Example of a translation rule described with the rule description language.

```
<sign>
    <gloss>LIPS </gloss>
    <movement trajectory="circular" direction="clockwise" radius-size="small"
    repetition-flag="no-repetition" hands-used="right">
        <config>
            <handshape> 14 </handshape>
            <palm-configuration> orientation="parallel-to-body"
            palm-direction="backwards" fingers-direction="upwards"
            </palm–configuration>
        </config>
        <location local="head" subdivision="mouth"/>
        <facial-expression="neutral">
    </movement>
</sign>
```

Fig. 4. Example of the XML representation of the LIPS sign in LIBRAS.

Based on these definitions, we specify a XML representation to represent the attributes of the rules define above. Each rule has a count field that represents the number of linguistic elements. For each element, there is a title field that represents the morphological-syntactic class, a newpos field that indicates the position of the new element after the application of the rule, and a newproperty optional field that represents the attribute and indicates possible change in the linguistic elements. For the rule to be applied, the elements in the original text should appear in the same order defined in the defined rule. Fig. 3 illustrates an example of the XML representation of a rule. It indicates that a BP sentence in the "SUBJECT + VERB + OBJECT" order must be translated to a "OBJECT + SUBJECT + VERB" sentence in LIBRAS.[4]

2.4.3. Sign description language

The Sign Description Language is used to describe the signs that will compose the Sign Dictionary. From this description, the WikiSigns tool can render a video for the sign based on the 3D avatar model.

In this language, a sign can be defined as a set of movements, where each movement has an initial and final configuration of hands, arms and face, a type of trajectory (e.g., rectilinear, circular, semicircular, etc.), a direction (e.g., from left to right, from inside to outside, etc.) and flags to indicate which hands are used in the movement (e.g., left, right or both). Formally, we define a s sign as follow:

$$s = \langle gl, mov_1, mov_2, \ldots, mov_n \rangle, \tag{4}$$

$$mov_i = \langle cf_{ini}, cf_{fin}, traj, dir, lh_f, rh_f \rangle, \quad i = 1, 2, \ldots, n, \tag{5}$$

$$cf_t = \langle hs_l, hs_r, or_l, or_r, loc_l, loc_r, fe \rangle, \quad t = ini, fin, \tag{6}$$

$$or_h = \langle or\_palm, dir\_palm, dir\_fing \rangle, \quad h = l, r, \tag{7}$$

$$pa_h = \langle subd, loc \rangle, \quad h = l, r, \tag{8}$$

where gl is a gloss of the sign and $mov_1, mov_2, \ldots, mov_n$ are the set of movements of the signs. The cf_{ini}, cf_{fin} parameters represents the initial and final configuration of each movement (mov_i), respectively; *traj* and *dir* represent the type of trajectory (e.g., rectlinear, circular, semi-circular, pontual, etc.) and the direction of each movement, respectively and lh_f e rh_f are flags that indicate, respectively, if the left and right hands are used in the movement. hs, or, pa e fe represent the handshape, palm orientation (e.g., upwards, downwards, forwards, backwards, etc.), the location and facial expression of each configuration. The l and r indexes of these phonemes represent the left and right hand, respectively. Finally, the or_palm, dir_palm e dir_fing parameters represent, respectively, the reference plane of the palm orientation (parallel to the body or parallel to the ground), the palm direction and the fingers direction, whereas the *loc* e *subd* parameters represent, respectively, the location in the body (e.g., head, trunk or neutral space) and their subdivisions (e.g., forehead, nose, mouth, cheek to the head location).

From this formalization, an XML representation was defined to represent these parameters and, therefore, to describe signs. Fig. 4 illustrate example of XML representations for the LIPS sign in LIBRAS, respectively.

According to Fig. 4, the trajectory attribute represents the type of trajectory of the movement. The hands-used and repetition-flag attributes represent, respectively, the flags that indicate which hands are used in the movement (left, right or both) and the number of repetition of the movement. The direction and radius-size are unique attributes of circular movements and represent their direction (clockwise or anti-clockwise) and radius-size (small, medium or large). The handshape is represented by an integer value from 1 to 64, according to the handshape options available on Felipe and Monteiro [11]. The palm

[4] According to LIBRAS specialists, it is the most common rule for translating BP sentences to LIBRAS.

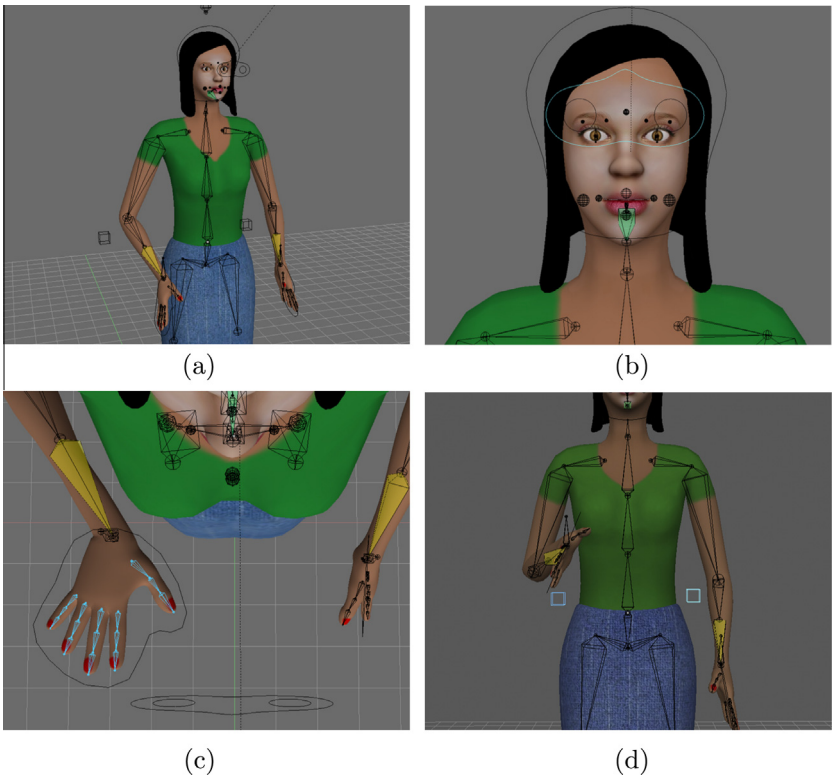

Fig. 5. (a) The 3D avatar model. (a) The 3D-virtual agent model. (b) Emphasis on bones of face, (c) hand and (d) body.

orientation phoneme (palm) has the orientation, palm-direction and fingers-direction attributes that represents, respectively, the *or_palm*, *dir_palm* e *dir_fing* parameters.

Thus, in Fig. 4, the LIPS sign was defined with the right hand performing a circular movement around the mouth. The handshape, palm orientation and location phonemes do not change during movement and therefore, the initial and final configuration of the movement are equal.

To describe these signs, we could also use another notation systems such as SignWriting, HamNoSys (Hamburg Sign Language Notation System) and Stokoe notation. These notation systems use visual symbols and numeric codes to represent signs and sentences in sign languages, which are more natural for deaf people. However, according to Morrissey [30], most of these systems are not used by the deaf, since they are not easily learned, written nor is there a standardized accepted form. In addition, the computational processing of the visual symbols of these notations, which involves digital image processing, is more difficult and time consuming than the proposed XML representation. Moreover, the proposed XML representation is an intermediary representation of signs, transparent for users, which describe signs using accessible and visual interfaces elements in the Wikisigns tool.

2.4.4. 3D avatar model

To represent the signs described by the Sign Description Language in the proposed solution, a 3D avatar was modeled and implemented. It was developed using Blender software[5] with an armor composed of 82 bones distributed as follows:

- 15 bones in each hand to setup handshape;
- 23 bones on the face to setup facial expressions and movements;
- 22 bones in arms and body to setup arm and body movements;
- 7 auxiliary bones (i.e., bones that do not deform the mesh directly).

Thus, to configure the movements of the fingers, for example, is necessary to define parameters of location and rotation for each of these 15 bones. The same should be done to the bones of the face of the avatar. The arm movement is performed by moving only two bones. The first one is located on the pulse of the avatar and the second one is an auxiliary bone which controls the deformation of the elbow and forearm. We use inverse kinematics to relate the deformation between bones related. Thus, if there is a movement in the wrist bone, for example, it will spread to the bones of the arm and forearm.

[5] www.blender.org.

The 3D-avatar model (with all bones) is illustrated in Fig. 5a. Fig. 5b–d illustrate the emphasis on the bones of the face, hand and body of this 3D model, respectively.

3. Usage scenarios: DTV, Web and Digital Cinema

In this section, we will present implementations of the proposed solution for Digital TV (DTV), Web and Digital Cinema. In Section 3.1, we will present the implementation of WikiSigns, which is common for all prototypes. In Sections 3.2, we will present LibrasTV, prototype of the proposed solution developed for the Brazilian Digital TV System (SBTVD). Finally, in Sections 3.3 and 3.4, we will present LibrasWeb and CineLibras, prototypes developed for Web and Digital Cinema platforms, respectively.

As mentioned earlier, all three implementation used Brazilian Portuguese as input language and the Brazilian sign language (LIBRAS) as output. LIBRAS is the sign language used by most of Brazilian deaf people and recognized by Brazilian law.

3.1. WikiSigns

The WikiSigns tool was implemented as described in Section 2.4.1. Its Web interface was developed using the PHP programming language with the aid of the Adobe Flash technology.[6] The Rule and Sign Description Generator modules, responsible for the generation of the XML representation of translation rules and signs, respectively, were also developed using the PHP programming language.

The manipulation of the 3D-avatar model (described in Section 2.4.4) was done automatically using scripts developed using Python programming language. These scripts are responsible for interpreting the intermediate language, configure the phonemes and render the signs using libraries of pre-recorded poses. Each pose in this library have the coordinates of location and rotation of the bones of the 3D avatar model. For example, for each facial expression, the user must configure the rotation and locations of the 23 bones on the face of the 3D-avatar model.

3.2. LibrasTV

The integration of the proposed solution into Digital TV (DTV) systems, called LibrasTV, can be performed in several ways. For example, (1) all components can be integrated into TV station and the LIBRAS video track would be generated and broadcasted as a secondary video stream to DTV receivers. Another option would be (2) to run all components in DTV receivers, generating the LIBRAS video in DTV receivers or loading this information from the interaction channel. LibrasTV, however, is based on the following strategy:

- The Filtering, Subtitle Extraction and Machine Translation components are grouped in a module called "LIBRAS Translator" which is integrated into the TV station (or content provider). This module receives a subtitle stream, extracts the Brazilian Portuguese sentences from that stream and translated them to a sequence of glosses in LIBRAS. This sequence of glosses is then encoded along with the synchronization information (timestamps) to be encapsulated in the DTV Transport Stream (TS). The process of codification of the LIBRAS elementary stream is based on a proposed encoding described in Appendix A.
- The Animation and Distribution components are grouped and implemented as a DTV interactive application that will run on DTV receivers. This application extracts the sequence of glosses and synchronization information encapsulated in the TS, decodes, synchronizes and displays the LIBRAS video track with the aid of the Sign Dictionary.
- The Sign Dictionary is loaded from the interaction channel or stored in an external memory device (e.g., a USB).

One of the main advantages of this approach is that it uses low bandwidth of the TV channel, since only a encoded sequence of glosses (text) is transmitted in TS. Another important feature of LibrasTV is that it respects the regional specificities of LIBRAS language, since each user can use his own Sign Dictionary. Thus, the LIBRAS video track can be customized according to the dictionary used. Furthermore, LibrasTV requires low processing in DTV receivers, since the Filtering, Subtitle Extraction and Machine Translation components are executed in the TV station.

To implement this solution, however, a encoding protocol must be defined to insert the sequence of glosses and synchronization information in the DTV TS. This protocol is presented in Appendix A.

In the next section, we will present some implementation details of the LibrasTV components for the Brazilian Digital TV System (SBTVD).

3.2.1. Implementation of LibrasTV components

As mentioned earlier, on LibrasTV, some components of the proposed solution are integrated into the TV station ("LIBRAS Translator" module), and others are executed as an interactive application in the DTV receiver.

[6] www.adobe.com/en/products/flashplayer.html.

The Filtering, Subtitle Extraction and Machine Translation components (which are part of the LIBRAS Translator module) are executed in the TV station and were implemented using the C++ programming language. The Filtering and Subtitle Extraction modules were developed based on MPEG-2 Systems [22] and ABNT NBR 15606-1 [2] specifications, respectively. These components receive a MPEG-2 TS, identify MPEG-2 TS subtitle packets (Filtering) and extract the Brazilian Portuguese sentences and synchronization information (timestamps) from these packets (Subtitle Extraction).

The Machine Translation component receives the Brazilian Portuguese sentences and translates them to a sequence of glosses in LIBRAS. The Morphological-syntactic classification is done based on a Portuguese language corpus, called "Bosque"[7] [14]. This corpus was developed by the Syntactic Forest project [14] and has 9368 sentences and 186,000 words obtained from "Folha de São Paulo",[8] a Brazilian newspaper, and from "Público",[9] a Portuguese newspaper as well. The entire corpus was morphologically and syntactically classified and fully reviewed by linguists.

From the Bosque sentences, the PPM-C algorithm is applied to classify the tokens morphologically and syntactically. The Markov order defined empirically for the PPM model was 5. This value was choose in order to maintain a good threshold between accuracy and run time. Afterwards, the simplification strategy is applied using a Brazilian Portuguese to LIBRAS Dictionary (BP-LIBRAS dictionary) to do the lexical replacement step (see Section 2.1). The BP-LIBRAS dictionary was developed in two parts. The first part was extracted from the "LIBRAS Illustrated Dictionary of Sao Paulo", a LIBRAS dictionary which has 43,606 entries. The other one was generated by a human specialist from the verbal inflection variation, where each inflected verb has its translation to its infinitive form. The full dictionary consists of 295,451 entries.

Finally, the translation rules are applied to translate the tokens to a sequence of glosses in LIBRAS. In this implementation, we used 11 translation rules defined by LIBRAS specialists using the Rule Description Language (see Section 2.4.2). These rules were independent of domain (i.e., were designed for open-domain scenarios) and were described using only morphological and syntactic elements. One example of translation rule defined in this implementation is the rule presented in Fig. 3, which translates a BP sentence in "SUBJECT + VERB + OBJECT" order to the "OBJECT + SUBJECT + VERB" order, the correct order in LIBRAS.

To encode and embed the sequence of glosses into a MPEG-2 TS stream, a Coding component was also implemented in the LIBRAS Translator module. This component was also developed using the C++ language and works as follows. Initially, it receives the sequence of glosses for the Machine Translation component and generates the LCM and LDM messages, according to the protocol defined in Appendix A. These messages are then encapsulated into DSM-CC stream events along with the synchronization information (timestamps) and are packaged into MPEG-2 TS packets for multiplexing. The Multiplexer receives these packets, multiplexes them along with the audio, video and data MPEG-2 TS packets, and sends a single MPEG-2 TS stream to be transmitted in the broadcast channel.

On the receiver side, a Ginga-J interactive application,[10] combines the functionalities of the Animation and Distribution components to generate and display the LIBRAS video in a synchronized way. This application has also a Decoding component, responsible for decoding the DSM-CC stream events and extracting the sequence of glosses and synchronization information encapsulated in these events.

The Decoding component was developed using the "Broadcast streams and file handling" classes, available on com.sun.broadcast package of Ginga-J. By using these classes, the application can decode DSM-CC stream events and extract the sequence of glosses and the synchronization information from them. The Animation and Distribution components were developed using the "Java Media Framework (JMF) 1.0", available on javax.media packages of Ginga-J.[11]

In this version of the prototype, the Sign Dictionary was generated based on the 3D avatar model described in the Section 2.4.4 and stored in an external USB memory device. The neutral configuration used at the start and end of the video of each sign was defined according to the suggestion of LIBRAS interpreters, placing the hands and arms extended in a straight line down and with a neutral facial expression (i.e., without applying movement in the facial bones).

Fig. 6 illustrates some screenshots of the LIBRAS window generated by LibrasTV. This application has been tested and validated on Openginga,[12] an open source implementation of the Ginga middleware.

In the next section, we will present the LibrasWeb, the prototype of the proposed solution for Web.

3.3. LibrasWeb

The prototype of the proposed solution developed for Web, called LibrasWeb, was implemented with all its components running on one (or more) server(s) in the cloud. It receives a captioned multimedia content, generates a LIBRAS video from its subtitle and mixes the LIBRAS video with the multimedia content, making it accessible.

An important feature of this prototype is that it can be seen as a cloud service that makes multimedia contents accessible for the deaf ("Accessibility as a Service" – AAAS) [5].

[7] www.linguateca.pt/floresta/corpus.html#bosque.

[8] www.folha.uol.com.br.

[9] www.publico.pt.

[10] Ginga-J is the procedural part of the Ginga middleware, the middleware of SBTVD. The Ginga-J APIs are based on the Java programming language [34].

[11] Similar APIs and packages are also available in others DTV middlewares, such as the Americans ACAP (Advanced Common Application Platform) and OCAP (OpenCable Application Platform) and European MHP (Multimedia Home Platform) [29].

[12] The Openginga is an open source implementation of the middleware Ginga available at www.gingacdn.lavid.ufpb.br/projects/openginga.

(a) (b)

Fig. 6. Screenshot of the execution of LibrasTV over Openginga.

Users access LibrasWeb through a Web interface. From this interface, it can submit a captioned video or a video with a separate subtitle file to be processed by LibrasWeb. In the next subsection, we will describe the implementation of LibrasWeb components.

3.3.1. Implementation of LibrasWeb components

On LibrasWeb, all components were developed using the C++ programming language. For Filtering, Subtitle Extraction and Machine Translation components, it reuses the same implementation of LibrasTV (see Section 3.2.1).

The Animation component receives the sequence of glosses from the Machine Translation component and generates a LIBRAS video track with the aid of the Sign Dictionary. To synchronize the LIBRAS video with the input multimedia content, the Animation component extracts the first clock reference of the input multimedia content, called PCR (Program Clock Reference). This clock is then used as the reference clock of the LIBRAS video. Timestamps for all signs are then generated based on this PCR and on presentation timestamps (PTS) of related sentences in subtitle.

The Distribution component receives the LIBRAS video along with the input multimedia content and mixes the LIBRAS video frames with the input video frames. To mix them in sync, the first step is adjusting the frame rate of the two videos to the same value. After this task, the LIBRAS video frames are mixed as a LIBRAS window over the video input, based on parameters for size and position provided by the user.

The process of adjusting the frame rate and mixing the videos together were implemented using FFmpeg,[13] an open source tool developed to record, manipulate, convert and transmit audio and video streams. As a result of this process, the Distribution component generates a new video file where the LIBRAS video overlaps the original multimedia content, making it accessible.

The Web interface of this prototype was implemented using the Ruby programming language and the Adobe Flash Player technology. It explores the use of interactive video, where the users' interactions are driven by interactive videos with LIBRAS interpreters. From this interaction, the user submits the multimedia content (video) along with parameters such as size and position of LIBRAS window, generating a request that will be processed by LibrasWeb.

Fig. 7 illustrates two screenshots of LibrasWeb interface. After setting the Web interface parameters, an instance of LibrasWeb is created and executed, generating an accessible copy of the input multimedia content (that is returned to the user). In Fig. 7a, it is shown the configuration screen for the LIBRAS window position. Fig. 7a shows four options for the LIBRAS window position (top left, top right, bottom left and bottom right) presented to the user on the interactive video. The user selects the desired option by clicking on one of the four positions. Fig. 7b illustrates the accessible copy of the multimedia content generated by the prototype and presented to the user on the screen. Optionally, the user can also download the (accessible) new copy of the content.

In the next section, we will present the prototype developed for Digital Cinema, called CineLibras.

3.4. CineLibras

The CineLibras, prototype of the proposed solution developed for Digital Cinema, was implemented considering the automatic generation of LIBRAS videos in Theaters. To perform this task, CineLibras runs on a server integrated with the Digital Cinema video player. The strategy used was to generate the LIBRAS video track from the movie subtitles and to transmit them to users' mobile devices (e.g., tablets or smartphones), allowing deaf people to view the LIBRAS translation on their own devices. Since cinemas are shared between deaf and non-deaf users, it is possible to display the LIBRAS translation without disrupt people without disabilities. This adjustment could be performed including special seats for deaf users in theater, where mobile devices would be embedded in the seats and programmed to receive the LIBRAS translations.

The CineLibras receives the movie subtitle stream in Brazilian Portuguese, generates a LIBRAS video from this stream and distributes the LIBRAS video to the users' mobile devices. It reuses components from the implementation of LibrasWeb (see

[13] www.ffmpeg.org.

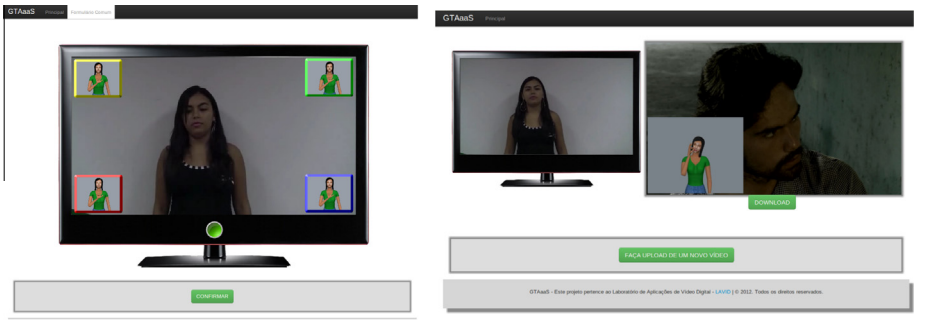

Fig. 7. Screenshots of LibrasWeb interface: (a) screen of configuration of LIBRAS window position and (b) screen of presentation of the accessible copy of content generated by LibrasWeb.

Fig. 8. Demonstration of CineLibras in the SBRC 2012. The movie appears in the background, whereas the LIBRAS video track is generated in real time by the CineLibras prototype and transmitted to the users' mobile devices.

Section 3.3.1), with changes in the Filtering, Subtitle Extraction and Distribution components. The Filtering and Subtitle Extraction components extract subtitles in DCP (Digital Cinema Package) format[14] [9] used to encode subtitles in Digital Cinema, and the Distribution component transmits the LIBRAS video (generated in real time by the Animation component) via HTTP streaming for connected mobile devices. In this solution, the delay and jitter does not tend to be a problem, since, as mentioned previously, it is acceptable to have a delay of up to four seconds in a live closed captioning system [1].

On the mobile devices, users connect to CineLIBRAS using players with support for receiving MPEG-2 TS via HTTP streaming. Some preliminary tests were performed on mobile devices with Android OS 2.2, 2.3 and 3.0 using VLC Media Player,[15] a video player with support for receiving videos via HTTP streaming.

Fig. 8 illustrates a demonstration of CineLibras performed in the 30th Brazilian Symposium of Network and Distributed Systems (SBRC 2012[16]) which took place in the city of Ouro Preto.

4. Results and discussion

After implementing these prototypes, some tests with them were performed to evaluate the proposed solution. These tests include quantitative measures and qualitative evaluation with Brazilian deaf users and were performed in three parts. In the first part, some accessible contents generated by the proposed solution (prototypes) are evaluated by Brazilian deaf users with respect to their level of understanding (intelligibility), the quality of translation and the naturalness of such content. In the second part, the translation delay is evaluated in order to investigate whether the proposed solution is able to generate accessible contents into environments that require real-time translation such as TV. Finally, in the third part, the WikiSigns tool is evaluated by Brazilian deaf users and LIBRAS interpreters in order to investigate the effectiveness and efficiency of users in the generation of the Sign Dictionary of the proposed solution.

[14] DCP is a collection of digital files used to store and transmit audio, video, data and subtitle streams in Digital Cinema.
[15] www.videolan.org/vlc.
[16] www.sbrc2012.dcc.ufmg.br.

Table 1
Videos used in tests.

Video	Gender	Duration	Description
Video1	Movies, series and soap operas	65 s	This video is a part of a movie produced by UFPB TV (the TV ofFederal University of Paraiba – UFPB), developed with academic purposes
Video2	News	26 s	This video is a part of a news program presented on 14 October 2008 on TV Globo, a Brazilian TV station
Video3	Variety shows	70 s	This video is a part of a variety show presented on 11 November 2011 on TV Record, a Brazilian TV station
Video4	Children's programs	888 seg	This video is a short film 3D animation produced by Blender Foundation[a]

[a] www.blender.org/blenderorg/blender-foundation.

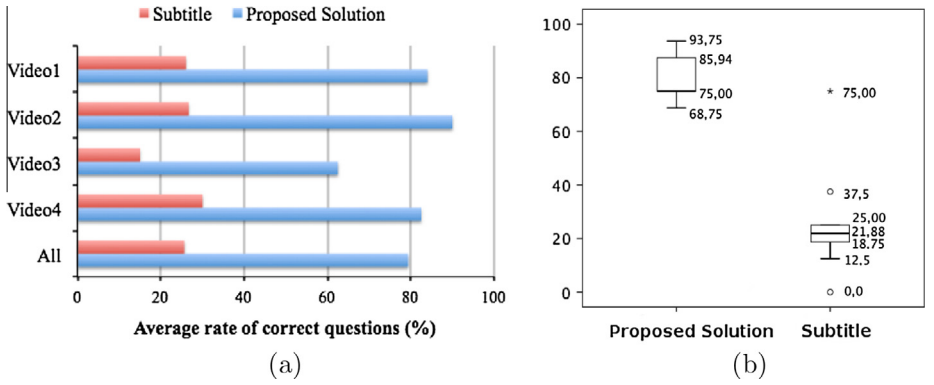

Fig. 9. Results of the comprehension tests (sixteen questions about the contents presented). (a) Average results of users performance; (b) box plot diagram of users' performance.

4.1. Content intelligibility

The subjective tests to assess the intelligibility of the accessible contents generated by the proposed solution were performed with twenty Brazilian deaf volunteers of Foundation to Support People with Disabilities (FUNAD), located in the city of João Pessoa, Brazil. The group of users consisted of nine men and eleven women ranging in age from 13 to 56 (with an average value of 28.6 years).

Initially, they were randomly divided into two groups of ten users: one group to evaluate video with subtitles and another to evaluate videos with LIBRAS video tracks generated by the LibrasWeb. Then, they were invited to watch four videos (see in Table 1[17]) with its treatment (subtitle or LIBRAS video track generated by the LibrasWeb) and to complete a questionnaire about some aspects of the solution.

The applied questionnaire had two parts. In the first part, users answered sixteen questions about the contents (videos) presented to assess their level of comprehension.[18] In these questions, users have to select which of four alternatives (A, B, C or D) is related to the content presented, where only one of the alternatives is correct. For all questions, the fourth alternative (D) represented a "I do not know" option, which was included to prevent users randomly choose one of the alternatives when they did not know the correct answer. In the second part, users answered five questions rating these contents on a 1-to-6 scale[19] for LIBRAS grammatically, understandability, naturalness, quality of presentation, among others. During tests, LIBRAS interpreters mediated communication with the deaf users.

Fig. 9 shows the results of the comprehension tests (first part of questionnaire). According to Fig. 9a, for all videos, users who watched contents with LIBRAS videos tracks generated by the proposed solution (LibrasWeb) had a greater average rate of correct questions (79.38%, for all videos, with a standard deviation of 9.34%) than users who watched videos with subtitles (25.63%, for all videos, with a standard deviation of 19.86%).

Observing the dispersion of these results (see Fig. 9b), users who watched contents with LIBRAS video tracks (LibrasWeb) had also a lower dispersion in their performance results (the median, first and third quartile values of the distribution were

[17] These videos belong to different genres (news, movies, series and soap operas, children's programs and variety shows) and were chosen prioritizing the most representative genres of Brazilian TV. According to a survey conducted by Fundação Getúlio Vargas (FGV) and Brazilian Association of Radio and Television (ABERT) [12], these genres of content represent approximately 82% of Brazilian TV programming.

[18] To assist users, each video was presented two times before they answer the comprehension questions.

[19] A 1-to-6 scale was chosen because according to Morrissey [30], even scales encourages users to make positive or negative evaluations, avoiding neutral evaluations. In addition, this scale was also used in other related work (e.g., San-segundo et al. [31]).

Table 2
Average values for the evaluated aspects (scales from 1 to 6).

Aspects	Proposed Solution		Subtitle	
	Average value	Standard deviation	Average value	Standard deviation
Understandability	4.60	1.68	3.70	2.33
Grammatically	4.60	1.56	4.13	2.05
Naturalness	4.40	1.74	–	—
Quality of hand movements	4.80	1.40	–	–
Quality of facial expressions	4.56	1.89	–	–

75.00%, 75.00% and 85.94%, respectively). Furthermore, no outlier was identified in this distribution, which indicates that all users had an average rate of correct responses between 68.75% and 93.75%.

For accessible contents with subtitles, the median, first and third quartile values of the distribution were 21.88%, 18.75% e 25.00%, respectively. This means that less than a quarter of users had an accuracy rate greater than 25.00%. In addition, three outliers were identified in this group: a negative outlier, representing a user who did not hit any test question; and two positive outliers, representing two users who have obtained a average rate score of 37.50% and 75.00%.

To check if the difference in performance between the two groups is significant, we applied a *t*-test to these results. Considering a confidence level of 95% and 18 degrees of freedom, the *t*-value obtained for this test, **7.74**, was greater than the critical value for the *t*-test, **2.12**. Thus, we can observe (with a confidence level of 95%) that there was a significant difference in comprehension of contents by Brazilian deaf users when multimedia contents had LIBRAS video tracks generated by the proposed solution in comparison with the level of comprehension of contents when multimedia contents had subtitles.

Others aspects of the solution such as quality of translation, naturalness of presentation, among others, were also evaluated by deaf users (second part of questionary). Table 2 shows the average results.

For contents with subtitles, some inconsistencies in results were found. For example, with respect to the grammatically, some users indicated that the contents with subtitles were compatible with the LIBRAS grammar (an average score of 4.13), whereas, in fact, the subtitles contents were in Brazilian Portuguese grammar.

Furthermore, the understandability score was not compatible with the performance of users in the comprehension tests. Some users had evaluated that these contents were reasonably understood (an average score of 3.70), but the results of the comprehension tests showed that most users had not understood well the contents (average accuracy of 25.00% of the questions). The Pearson correlation coefficient and the Spearman's rank correlation coefficient [41] obtained for the two variables (understandability score and comprehension tests) were **0.033** and **−0.182**, respectively, indicating a low correlation between the two variables. One possible explanation for this inconsistency is that, according to Wohlin et al. [41], humans try to look better when they are evaluated, which may have disrupted the output of test.

For the accessible contents generated by the proposed solution, all aspects had a moderate score (greater than 4.30). All these measures, however, had a high standard deviation (greater than 1.30), which indicates that the opinions of users were divergent. As in San-Segundo et al. [31], we observed some probable causes for this outcome during the tests. For example, during the tests, there were discrepancies between users about the structure of same sentences in LIBRAS. Like other sign languages (e.g., Spanish Sign Language San-Segundo et al. [31]), LIBRAS has an important level of flexibility in the structure of sentences. This flexibility is sometimes not well understood and some of the possibilities were considered as wrong sentences. In addition, there were also discrepancies between users about the correct signing of some signs. For example, users disagreed about the correct signing of the COFFEE and MARKET signs.

One alternative to reduce these discrepancies would be to use custom LIBRAS dictionaries. However, the development of custom LIBRAS dictionaries is a very time consuming task. Another alternative would be to invest more efforts to standardize LIBRAS. In this case, a wider dissemination of LIBRAS in ICT (e.g., in TV, Web and Cinema), would help to standardize it.

We also performed an automatic evaluation of the machine translation output. To perform this test, initially, we asked two LIBRAS interpreters[20] to translate all sentences of the Bosque corpus[21] into a sequence of glosses in LIBRAS, generating a reference translation for the entire corpus. Then, we translated all the sentences of Bosque using the LibrasWeb and calculated the values of WER (Word error rate) and BLEU (Bilingual Evaluation Understudy) based on the reference translation. We chose these objective measures because they were also used in other related works (although in different domains) [33,37]. We also calculated the values of BLEU and WER for a Signed Brazilian Portuguese (SBP) solution, i.e., a solution based on direct translation from BP to LIBRAS (without considering grammar differences), (e.g., the solution proposed by Amorim et al. [3]). The idea was to analyze the output of LibrasWeb machine translation and SPB results and compare them. Table 3 illustrates the percentual values of BLEU (with different n-gram precisions) and WER for both solutions.

According to Table 3, in these tests, LibrasWeb measurements were better than SBP measures for all n-grams precisions. The values of BLEU 4-gram = 12% and WER = 75.3%, respectively, helps to evaluate how difficult is this task in an open scenario such as Digital TV. However, this result is not sufficient to conclude that the proposed translation is good or not.

[20] One LIBRAS interpreter was responsible for translating and the other for reviewing.
[21] The Brazilian Portuguese corpus used in the implementation of the Morphological-syntactic classification module – see Section 3.2.1.

Table 3
BLEU and WER for LibrasWeb machine translation output and a SBP solution [3].

	LibrasWeb (%)	SBP solution [3] (%)
BLEU		
1-gram	48.5	40.7
2-gram	30.1	22.2
3-gram	18.9	11.4
4-gram	12.0	5.5
WER	75.3	87.7

Table 4
Measure of the average delay of each component of LibrasTV.

Components	Avg. value (ms)	Std. dev. (ms)	Max. value (ms)	Min. value (ms)
Filtering and Subtitle Extraction	0.024	0.022	0.554	0.017
Machine Translation	0.975	2.957	80.126	0.220
Coding	0.215	0.089	1.061	0.072
Decoding	0.170	0.143	0.519	0.020
Animation and Distribution	42.445	8.747	59.998	20.000
Total	43.805	9.434	142.21	20.509

Table 5
Signs of LIBRAS used in tests.

Sign	Movement type
PRESIDENT	Rectilinear
LIPS	Circular
TEACHER	Semi-Circular
SILENT	Pontual
UNCLE	Pontual

According to Su and Wu [37], objective evaluation based on objective measures is insufficient to evaluate the quality of translation for sign languages, since they are visual and gestural languages. In addition, the results of the comprehension tests performed with Brazilian deaf users showed that they had a good comprehension of the LIBRAS tracks generated by the proposed solution, even with a translation still not close to human translation.

4.2. Delay of translation

The test to calculate the average delay of the solution components (LibrasTV, in this case) was performed using a real DTV signal as input during a whole day (24 h). During this time, the MPEG-2 TS of "TV Record" Brazilian DTV channel was tuned in real time and streamed to the LIBRAS Translator and Multiplexer.[22] The whole time MPEG-2 TS packets with closed caption data were received by the LIBRAS Translator. The LIBRAS window was generated by the LibrasTV prototype from these closed caption data and the delay of each LibrasTV module was measured and stored. The average, standard deviation, maximum and minimum values of these measures are shown in Table 4.

According to Table 4, the average delay to run all LibrasTV components was less than 44 ms. The maximum delay obtained (considering the maximum delay of each component) was 142.26 ms, whereas the minimum delay was 20.509 ms.

Considering that according to Brazilian law [1] in a live closed captioning system is acceptable to have a delay of up to four seconds, we also applied a *t*-test to check if the delay of this test is within this range of tolerance (4 s). For a confidence level of 95% and 2192 degrees of freedom, the *t*-value obtained for this test (**19632.87**) is greater than the critical value for the *t*-test (**1.96**). Thus, as our test was conducted with an open and representative vocabulary (24 h of Brazilian TV programming, which according to a survey conducted by Fundação Getúlio Vargas (FGV) and Brazilian Association of Radio and Television (ABERT) is quite diversified [12]) and in a real scenario, it can be stated with a confidence level of 95% that the proposed solution can generate LIBRAS video tracks in a live and real time scenario, such as TV.

[22] To perform this tests, we used two mini-computers with an Intel Dual Core T3200 2 GHz processor and 4 GB GB of RAM memory. Oneof these computers was used to run the LIBRAS Translator prototype and the other to run the Openginga with the LibrasTV application prototype. The operating system used in both was the Linux Ubuntu 10.0.4 kernel 2.6.32.

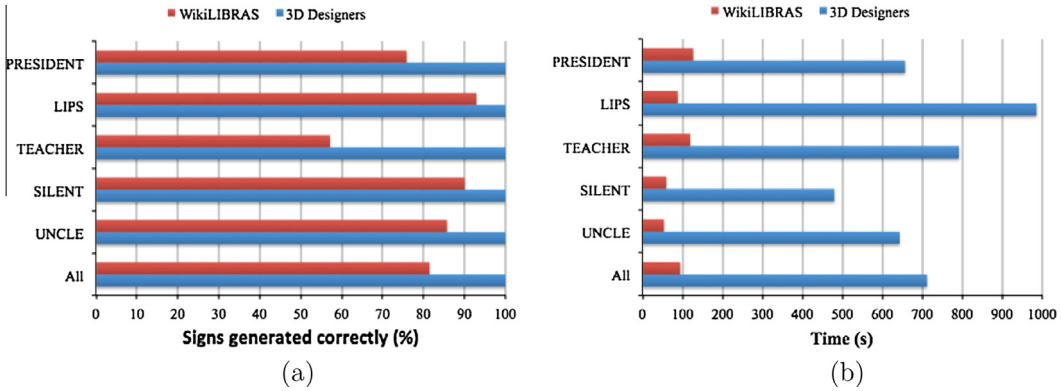

Fig. 10. Results of WikiSigns tests (WikiSigns vs Manual). (a) Average percentage of signs generated correctly by users; (b) average time to generate signs correctly.

4.3. WikiSigns evaluation

Finally, the subjective tests to assess the WikiSigns tool were performed with eleven Brazilian deaf volunteers and three LIBRAS interpreters from FUNAD. The group of users consisted of seven women and seven men ranging in age from 12 to 42 (with an average value of 25.4 years).

Users were invited to generate five signs using WikiSigns (see in Table 5[23]) and to complete a questionnaire indicating the signs generated correctly and the main difficulties found for the signs generated incorrectly. During tests, the average time spent by users to generate the signs (efficiency) was also stored in WikiSigns.

To compare with the performance of users in WikiSigns, a similar experiment was also performed with three 3D designers[24] animating the same signs in the Blender tool[25] based on the 3D avatar model.[26]

The average results of these tests are illustrated in Fig. 10. According to Fig. 10a, we can observe that 3D designers generate all signs correctly using Blender tool, whereas deaf users and Brazilian interpreters generates correctly, on average, about 81.43% of signs in WikiSigns. However, according to Fig. 10b, the average time spent by users using WikiSigns (93.96 s) was much smaller than the average time spent by 3D designers using the Blender tool (711.33 s).

Among the difficulties pointed out by LIBRAS interpreters and deaf users to generate the signs using WikiSigns, the main difficulty mentioned was to understand some parameters of the WikiSigns Web interface. Then, a proposal for future work is to include video with LIBRAS interpreters in the Web interface to assist users during navigation.

We also applied a *t*-test to check if this performance difference, with respect to the average time to generate a signal, is significant. Considering a confidence level of 95% and 15 degrees of freedom, the *t*-value obtained for this test, **12.53**, is greater than the critical value for the *t*-test, **2.13**. Thus, it can be stated (with a confidence level of 95%) that there is a significant difference in the average time of generation of signs when LIBRAS specialists use WikiSigns in comparison with the average time of generation of those signs by using animation tools.

Besides producing contents in a shorter time interval, the number of deaf and LIBRAS interpreters is much greater than the number of available 3D-designers. In addition, 3D-designers need to learn LIBRAS or need signs references to animate the LIBRAS signs. Thus, it is possible to create a LIBRAS Dictionary more productively using WikiSigns, especially considering that one LIBRAS dictionary has about 10,286 signs [7].

5. Final remarks

In this paper, we described an approach to generate and embed sign language video tracks into multimedia contents. In the proposal, the source language subtitle streams are translated to a target sign language video which are synchronized and embedded into the original multimedia content as an extra layer of accessible content. The propose solution was also developed to generate accessible contents in scenarios that require live and real-time translation (e.g., TV) and it has a human computation tool (WikiSigns) that allow semi-automatic and collaborative generation of their linguistic constructions (translation rules and signs).

[23] The signs were chosen according to the type of movement, because the interaction in WikiSigns is directed by the type of movement of the sign. Thus, it is possible to cover the various types of interaction in WikiSigns. In addition, these signs cover the most common types of movements in a sign language (rectilinear, circular, semi-circular and pontual) [15].

[24] The three designers were experienced and participate in research projects involving 3D modeling and animation in the Digital Video Applications Lab (LAViD) of Federal University of Paraiba (UFPB), Brazil, where two were undergraduates and the other was a Master's student at UFPB.

[25] www.blender.org/.

[26] A reference video of each sign represented by a LIBRAS interpreter was also provided to assist them in the generation of signs.

Table A.6
LibrasTV encoding protocol: (a) LCM message, (b) LDM messages and (c) values for the resolution field.

(a)	
LCM{	
libras_control_id	8 bits
libras_control_length	16 bits
resolution	8 bits
window_line	16 bits
window_column	16 bits
window_width	16 bits
window_height	16 bits
}	
(b)	
LDM{	
libras_data_id	8 bits
libras_data_length	16 bits
number_of_signs	16 bits
for (i = 0; i < N; i++){	
gloss_bytes_length	8 bits
for (j = 0; j < M; j++){	8 bits
gloss_data_bytes	8 bits
}	
}	
}	
(c)	
Values	Resolution
0	1920 times 1080
1	1280 times 720
2	640 times 480
3	960 times 540
4	720 times 480
5	320 times 240
6–255	Reserved for future use

Furthermore, we developed three prototypes of the proposed solution for DTV, Web and Digital Cinema platforms using Brazilian Portuguese as input language and the Brazilian sign language (LIBRAS) as output, and performed a set of tests with Brazilian deaf users to evaluate the solution. This evaluation showed that the proposed solution is efficient and able to generate and embed sign language video tracks into different contents and scenarios, including scenarios real-time and open scenarios such as TV. Moreover, the proposed solution could improve the level of comprehension of multimedia contents when compared with contents with subtitles, the most common accessible strategy available on ICT platforms. The human computation tool was also evaluated by LIBRAS interpreters and Brazilian deaf users, and it was possible to observe that it is able to reduce the average time to produce the signs of the Sign Dictionary when compared with 3D designers animating signs in a animation tool.

Among the perspectives for future works, a natural evolution would be to adapt the proposed solution for audio inputs. Thus, it would be possible to investigate the generation of sign language videos from speech. Another proposal for future work involves the incorporation of motion capture equipment, for example, Microsoft Kinect[27] in WikiSigns, to improve the process of generating new signs. We also plan to explore semantic components in the translation, such as, better treatment of non-manual elements, use of classifiers, as well as the use of summarization strategies to reduce the length of the sentences translated.

Finally, another proposal for future work involves the inclusion of mechanisms to allow the revision of the generated translations by human collaborators. This would extend the role of human collaborators in the solution and allow the production of better quality translations for content that does not require real-time translation.

Appendix A. LibrasTV encoding protocol

The LibrasTV encoding protocol allows that the sequences of glosses and the synchronization information can be encapsulated in MPEG TS stream.[28] It is basically composed of two types of messages: LIBRASControlMessage (LCM), a control message, and LIBRASDataMessage (LDM), a data message.

[27] www.xbox.com.

[28] This protocol was submitted as a candidate and is being evaluated by the bodies responsible for defining the standards used in the Brazilian DTV System (SBTVD).

The LCM messages are used for periodic transmission of the initial settings of the LIBRAS video presentation (e.g., resolution, size and position of LIBRAS video), whereas the LDM messages are used for transmission of the sequence of glosses in LIBRAS. The syntax of LCM and LDM messages are presented in Table A.6.

According to Table A.6, the LCM and SDM messages begin with its identification and length fields (libras_control_id and libras_control_length for LCM and libras_data_id and libras_data_length for LDM). These fields are used to identify the type of message (LCM or SDM) and the message length in bytes, respectively.

The LCM is also composed by the following fields: resolution, window_line, window_column, window_width and window_height. The resolution field defines the resolution of the graphic layer used to display the window (e.g., 1920×1080, 720×480, etc.). The possible values for this field are shown in Table A.6. The window_line and window_column fields define the initial window position coordinates (of top left corner) on graphic layer, whereas window_width and window_height define the initial window size.

On LDM, the gloss_data_bytes fields transport the sequence of glosses (used to reference signs on Sign Dictionary) that are being encoded. Since these fields are inside a loop, several signs (glosses) can be transmitted in the same message. The number_of_signs field specifies the number of signs encoded in each LDM message.

To encapsulate the LCM and LDM in MPEG-2 TS,[29] an alternative is to use DSM-CC (Digital Storage Media-Command and Control) stream events [23]. The DSM-CC stream events are transmitted (encapsulated) in structures called Stream Event Descriptors, which allow that synchronization points are defined at the application level, allowing the synchronization of applications with other related media (e.g., audio and video). Its structure is basically composed of a event identification field (eventID), a temporal reference (eventNPT) and a private data field (privateDataBytes). Thereby, it is possible to encapsulate the LCM and LDM messages in the private data field (privateDataBytes) and the synchronization information in the temporal reference field (eventNPT), and thus, synchronizes the LIBRAS encoding protocol messages with other media and embed them into a MPEG-2 TS.

References

[1] ABNT, ABNT NBR 15290 – Accessibility in TV Captions Specification, 2005.
[2] ABNT, ABNT NBR 15606-1 – Digital Terrestrial Television – Data Coding and Transmission Specification for Digital Broadcasting – Part 1: Data Coding Specification, 2007.
[3] M.L.C. Amorim, R. Assad, B. Loscio, F.S. Ferraz, S. Meira, Rybenátv: soluç ao para acessibilidade de surdos para tv digital (rybenátv: a solution for deaf accessibility on digital tv), in: Proc 16th Symp Multimed Web – Webmedia'10, Belo Horizonte, Brasil, 2010, pp. 243–248.
[4] K. Anuja, S. Suryapriya, S.M. Idicula, Design and development of a frame based MT system for English-to-ISL, in: Proc World Congr Nat Biol Inspir Comput, NaBIC'09, Coimbatore, India, 2009, pp. 1382–1387.
[5] T.M.U. Araújo, F.L.S. Ferreira, D.A.N.S. Silva, E.L.F. ao, L.D. Oliveira, L.A. Domingues, L.H. Lopes, Y. Sato, H.R. Lima, A.N. Duarte, G.L.S. Filho, Accessibility as a service: augmenting multimedia content with sign language video tracks, J Res Pract Inf Technol (2012) in press.
[6] G. Blakowiski, R. Steinmetz, A media synchronization survey: reference model, specification and case studies, IEEE J. Sel. Areas Commun. 14 (1996) 5–35.
[7] F.C. Capovilla, W.D. Raphael, A.C.L. Mauricio, Novo Diet-Libras: Língua de Sinais Brasileira (New Diet-Libras: Brazilian Sign Language), Edusp, fourth ed., 2010.
[8] J. Cleary, I.H. Witten, Data compression using adaptive coding and partial string matching, IEEE Trans. Commun. 32 (1984) 396–402.
[9] L. Digital Cinema Initiatives, Digital Cinema System Specification, 2008.
[10] R. Elliott, J.R. Glauert, J.R. Kennaway, A framework for non-manual gestures in a synthetic signing system, in: Proc Camb Workshop Ser Univers Access Assist Technol, Cambridge, UK, 2004, pp. 127–136.
[11] T.A. Felipe, M.S. Monteiro, Libras em Contexto: Curso Básico (Libras in context: Basic Course), WalPrint Gráfica e Editora, Rio de Janeiro, Brasil, sixth ed., 2007.
[12] FGV/ABERT, Pesquisa sobre TV Digital no Brasil (a survey about digital TV in Brazil), 2012. <Http://www.abert.org.br/site/images/stories/pdf/TV_Programacao.pdf>.
[13] S.E. Fotinea, E. Efthimiou, G. Caridakis, K. Karpouzi, A knowledge-based sign synthesis architecture, Univ. Access Inf. Soc. 6 (2008) 415–418.
[14] C. Freitas, P. Rocha, E. Bick, Floresta sintá(c)tica: bigger, thicker and easier, in: Proc 8th Int Conf Comput Process Port Lang, PROPOR'08, Aveiro, Portugal, 2008, pp. 216–219.
[15] S. Gibet, T. Leborque, P.F. Marteau, High-level specification and animation of communicative gestures, J. Vis. Lang. Comput. 12 (2001) 657–687.
[16] L. Haddon, G. Paul, Technology and the market: demand, users and innovation, technology and the market: demand, users and innovation, in: ASEAT Conference Proceedings Series, Edward Elgar Publishing, Cheltenham, UK, 2001, pp. 201–215.
[17] R. Hong, M. Wang, M. Xuy, S. Yany, T.S. Chua, Dynamic captioning: video accessibility enhancement for hearing impairment, in: Proc 13th Int Conf Multimed, Firenze, Italy, pp. 421–430.
[18] R. Hong, M. Wang, X.T. Yuan, M. Xuy, J. Jiang, S. Yan, T.S. Chua, Video accessibility enhancement for hearing-impaired users, ACM Trans. Multimed. Comput. Commun. Appl. (TOMCCAP) 7 (2011) 24:1–24:19.
[19] M. Huenerfauth, Generating american sign language animation: overcoming misconceptions and technical challenges, Univ. Access. Inf. Soc. 6 (2008) 419–434.
[20] M. Huenerfauth, L. Zhao, E. Gu, J. Allbeck, Evaluating American sign language generation through the participation of native ASL signers, in: Proc IX Int ACM SIGACCESS Conf Comput Access, ASSETS 2007, Tempe, USA, 2007, pp. 211–218.
[21] IBGE, Censo Demográfico 2000: Características gerais da populaç ao (Census 2000: general population), Technical Report, Brazilian Institute of Geography and Statistics, 2000. <Http://www.ibge.gov.br/home/estatistica/populacao/censo2000/populacao/censo2000_populacao.pdf>.
[22] ISO/IEC, Iso/iec13818-1 TR Information Technology – Generic Coding of Moving Pictures and Associated Information: Part 1: Systems Specification, 1996.
[23] ISO/IEC, Iso/iec13818-6 TR Information Technology – Generic Coding of Moving Pictures and Associated Information: Part 6: Extension for Digital Storage Media Command and Control Specification, 1998.
[24] J.R. Kenaway, J.R.W. Glauert, I. Zwitserlood, Providing signed content on the internet by synthesized animation, ACM Trans. Comput.–Hum. Interact. 14 (2007) 1–29.

[29] MPEG-2 TS is the transport protocol adopted by all current TV systems [22].

[25] M. Kipp, Q. Nguyen, A. Heloir, S. Matthes, Assessing the deaf user perspective on sign language avatars, in: Proc 13th Int ACM SIGACCESS Conf Comput Access, Dundee, Scotland, 2012, pp. 1–8.
[26] D.G. Lee, D.I. Fels, J.P. Udo, Emotive captioning, Comput. Entertain. 5 (2007) 3–15.
[27] S. Lee, V. Henderson, H. Hamilton, T. Starner, H. Brashear, S. Hamilton, A gesture based American sign language game for deaf children, in: Proc Conf Hum Factors Comput Syst, CHI'2005, Portland, USA, pp. 1589–1592.
[28] A. Moffat, Implementing the PPM data compression scheme, IEEE Trans. Commun. 38 (1990) 1917–1921.
[29] S. Morris, A. Smith-Chaigneau, Interactive TV standards: a guide to MHP, OCAP and Java TV, Elsevier, 2005.
[30] S. Morrissey, Data-Driven Machine Translation for Sign Languages, Ph.D. Thesis, Dublin City University, Dublin, Ireland, 2008.
[31] R. San-segundo, J.M. Montero, R. Córdoba, V. Sama, F. Fernández, L.F. D'Haro, V.L.L. na, D. Sánchez, A. Garcia, Design, development and field evaluation of a Spanish into sign language translation system, Pattern Anal. Appl. 15 (2011) 203–224.
[32] R. San-segundo et al, Proposing a speech to gesture translation architecture for Spanish deaf people, J. Vis. Lang. Comput. 19 (2008) 523–538.
[33] R. San-segundo et al, Speech to sign language translation system for Spanish, Speech Commun. 50 (2008) 1009–1020.
[34] G.L. Souza Filho, L.E.C. Leite, C.E.C.F. Batista, Ginga-j: the procedural middleware for the Brazilian digital TV system, J. Braz. Comput. Soc. 12 (2007) 47–56.
[35] T. Starner, A. Pentland, J. Weaver, Real-time American sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Anal. Mach. Intel. 20 (1998) 1371–1375.
[36] M.R. Stumpf, Língua de sinais: escrita dos surdos na internet (sign languages: writing of the deaf in internet), in: Proc V Conf Iberoam Inform Educ, Vi nadelmar, Chile, 2000, pp. 1–8.
[37] H.Y. Su, C.H. Wu, Improving structural statistical machine translation for sign language with small corpus using thematic role templates as translation memory, IEEE Trans. Mach. Transl., Audio, Speech, Lang. Process. 17 (2009) 1305–1315.
[38] T. Veale, A. Conway, B. Collins, The challenges of cross-modal translation: English to sign language translation in the Zardoz system, Mach. Transl. 13 (1998) 81–106.
[39] E. Veloso, V. Maia, Aprenda LIBRAS com eficiência e rapidez (Learn LIBRAS quickly and efficiently), Ed. Mãos Sinais, Rio de Janeiro, Brazil, 2011.
[40] L.N. Wauters, Reading Comprehension in Deaf Children: The Impact of the Mode of Acquisition of Word Meanings, Ph.D. Thesis, Radboud University, Nijmegen, Netherlands, 2005.
[41] C. Wohlin, P. Runeson, M. Hst, M.C. Ohlsson, B. Regnell, A. WesslTn, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publisher, Norwell, USA, 2000.
[42] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, M. Palmer, Machine translation system from english to american sign language, in: Proc 4th Conf Assoc Mach Transl Am, Cuernavaca, Mexico, 2000, pp. 54–67.