Towards Augmentative Communication in Virtual Environments

João Paulo Dias Galveias
joao.galveias@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisboa, Portugal

Abstract

In Portugal, there are an estimated 150.000 deaf people and about one million suffer from some kind of hearing loss. These people communicate using *Língua Gestual Portuguesa* (LGP), yet only about 100,000 people can speak this language. Some solutions use Extended Reality (XR) to translate LGP to text or speech or to teach people how to speak LGP. The problem with these solutions is that it takes a long time for people to learn how to speak LGP, while the translator solutions only help the person who does not speak LGP, meaning that the language barrier between LGP speakers and non-LGP speakers will still be there. In this work, we propose a new approach that uses the tracking capabilities of XR devices to provide augmentative communication in Virtual Reality (VR). Our approach improves the communication between deaf people and non-deaf people since it enables bidirectional communication. Our approach uses template matching for the translation of gestures into speech and natural language techniques and a 3D avatar to translate speech into LGP gestures. With our approach, we can recognize a selected group of LGP gestures and translate them into speech, and translate speech into gestures during a real-time conversation scenario. To evaluate our approach we developed a VR prototype that enables remote communication between an LGP speaker and a non-LGP speaker in a virtual environment. We tested three different dialogues where a pair of users exchanged information. Results show that our approach provides effective mechanisms enabling augmentative communication between an LGP speaker and a non-LGP speaker.

Keywords: Augmentative Communication, Gesture Recognition, Virtual Reality, Sign Language, Natural Language Techniques

1. Introduction

There are an estimated 150.000 deaf people in Portugal and about one million who suffer from some kind of hearing loss. In Portugal, *Língua Gestual Portuguesa* (LGP) was invented to give these people a way of communication. In 2010 it was estimated that only 100.000 people know how to speak LGP and that only 30.000 of those are actually deaf. Looking at these numbers, it is clear that there is a big discrepancy between people who know LGP and those who do not, making it a significant problem when trying to interact with people in their daily-basis. LGP interpreters tend to be recruited to intermediate conversations between deaf people and non-deaf people, for example in work meetings, conventions and other social events. However, this is not adequate for more private settings where a deaf and a non-deaf person want to have a social interaction.

Despite existing interest in creating solutions that make technology more accessible to the deaf community, the lack of technological advancements and deep knowledge about LGP itself means that only a few solutions have been developed that seek to solve this issue. These solutions aim to either recognize and translate sign language gestures using vision-based or sensor-based techniques, or translate speech to sign language using natural language techniques and an animated avatar. Some solutions combine both, however these rely on very expensive equipment and have a short vocabulary.

We want to provide means of augmentative communication so that deaf people and non-deaf people can have basic communications with each other in real-time, using a predefined set of gestures.

We can highlight our research statement as follows:

Our approach provides effective mechanisms for enabling augmentative communication between non-LGP and LGP speakers in Extended Reality.

To achieve this, we need to be able to translate a set of sign language gestures into speech and we also need to be able to translate speech into sign language gestures.

As a baseline, since it is very complex to develop an LGP gesture translator that performs with the same reliability as for example *Google Translate*, we want it to be possible for a person to communicate with LGP basic data such as age, names as well as scenarios of a meet and greet and asking for help in a timely manner, considering the limitation that gestures impose on communication speed.

The main contributions of this work include: **1)** a new approach to augmentative communication that enables remote communication between an LGP speaker and a non-LGP speaker in a virtual environment; **2)** the development of recording mechanisms for both static and dynamic sign language gestures and a database with 175 static gestures and 72 dynamic gestures; **3)** the development of gesture recognition techniques for both static and dynamic sign language gestures; **4)** a virtual reality prototype used to study our approach, that enables the augmentative communication between an LGP speaker and a non-LGP speaker; and **5)** results of the user study we carried out with the purpose of evaluating and validating our research statement hypotheses.

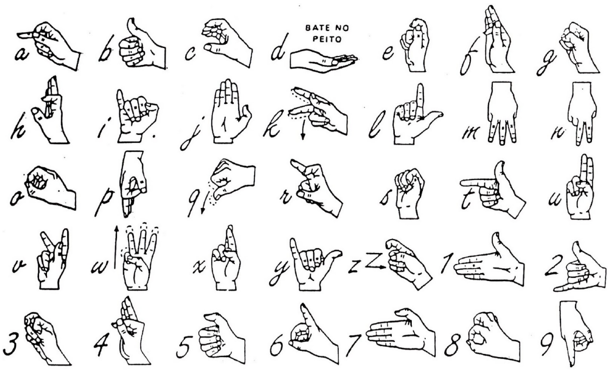

Figure 1: LGP Alphabet and Digits

2. Related Work

To develop augmentative communication in VR, we have to consider two situations. One is when a person does not speak and understand LGP and the other one is when a deaf person cannot hear and understand a spoken language, such as Portuguese. There is the need to develop two translators, one for the person who does not speak LGP and another one for the deaf person. For the LGP to speech translator, we need to be able to capture the gestures either with a vision-based or a sensor-based method. Then we need to recognize and classify these gestures. Lastly, we need to translate this text classification into speech. The main difficulties of this translator have been the tracking of the different areas of the body that give meaning to the sentences, such as the facial expressions, wrist rotations, arms motion and fingers positions. For the speech to LGP translator, we need to be able to translate the voice input into text. Then this text needs to be analysed semantically and morphologically to have the corresponding LGP gestures. Lastly, we need an avatar to perform these gestures. The main difficulty has been the lack of knowledge regarding sign language grammar, especially LGP [9].

Sign language translation has been explored for quite a while by several researchers. Contrary to popular belief, sign language is not a universal language. Since sign languages are natural languages, their gestures and grammars are different from country to country and even regions within a country. Our work translates LGP, mainly focusing on simple words, such as words that describe objects (e.g. *"Cadeira"* (Chair)), and sentences that can be made by a few simple gestures added together, for example *"Bom dia"* (Good Morning), as well as some LGP alphabet and digits gestures (see fig. 1) when doing the gesture recognition.

2.1. Sign Language to Text

One of the major challenges of sign language in VR is the ability to recognize sign language gestures. These gestures tend to have fast movements and the arms or hands are often overlapped. Li et al. [13] made a distinction between two types of gestures, static and dynamic gestures. A static gesture is a gesture that has no movement and can be detected solely by recognising its hand shape, whereas a dynamic gesture is a gesture that has movement and is recognized by detecting patterns in its movement. To recognize both types of gestures researchers developed two techniques, vision-based and sensor-based.

Vision-based solutions are the most commonly used, given that the devices required for them tend to be cheaper than devices used in sensor-based solutions. These devices can vary from smartphone cameras or webcams to some more expensive ones such as Leap Motion Controller (LMC), Oculus Quest 2 or Microsoft Kinect. These solutions usually combine image processing techniques alongside machine learning algorithms for gesture recognition and classification.

According to Kaur et al. [10] vision-based solutions can be divided into two categories: appearance-based and 3D model-based.

2.1.1. Vision-based: Appearance-based method

The Appearance-based methods rely on extracting features from the 2D visual appearance of the hand and then comparing them to annotated templates. They can be used for dynamic or static gestures and have a good realtime performance, however they tend to be less accurate due to noise from the background of the image since they do not have depth information [21]. These solutions use image processing techniques like edge detection and skin colour detection algorithms for feature extraction and pattern recognition.

Shrenika and Bala [18] developed a vision-based solution for American Sign Language (ASL) recognition. Their system recognizes the sign language gestures of the alphabet and the 0 to 9 digits. They capture the gestures using a normal camera and then pre-process the images converting them from RGB to greyscale and using the Canny Edge Detection algorithm to detect the edges of the hand. Lastly, they compare the pre-processed image with images of gestures from a database, using template matching and the Sum of Absolute Differences method to obtain the image from the dataset that better matches the captured gesture. Panella and Altilio [14] developed an application that used the front camera of a smartphone to capture the gestures and recognize them using template matching. Nuno Pereira [16] developed a system to translate LGP gestures from videos to text using a Convolutional Neural Network achieving an accuracy of 97.46%.

2.1.2. Vision-based: 3D Model-based method

3D Model-based solutions are based on the analysis of 3D features extracted from the hand shape such as the hand joints. These extracted features are used to calculate more significant features like the distance between the fingertips and the palm centre. These values are then compared with annotated templates for template matching approaches or are used in machine learning algorithms to predict the hand gesture. These are more computationally heavier than appearance-based solutions but tend to be more accurate since they can exploit depth information [21]. These solutions tend to use virtual reality devices equipped with depth cameras and algorithms that can extract the hand shape data necessary for gesture recognition.

Chong and Lee [5] used features like fingertip position, the hand palm position and the hand palm sphere radius from the LMC. Then they used this data to calculate the standard deviation of the absolute 3D palm positions, the 3D Euclidean distances of the palm center to each of the fingertips, the angles between two adjacent fingers and the distance between one fingertip and the consecutive fingertip. The data was normalized in the range [0, 1], to remove the factor of difference in the sizes of the subjects' hands. Then they conducted a study comparing the performance of Support Vector Machine (SVM) and Deep Neural Network (DNN) classifiers and the significance of the calculated features, using the ASL alphabet and digits gestures. The result was that the DNN outperformed the SVM with an accuracy of 85.65% vs. 67.54% and the most significant feature was the distance between one fingertip and the consecutive fingertip. They also pointed out that the letters "H" and "U" tended to be misclassified as each other due to having a similar hand shape. Vaitkevičius et al. [20] did a similar work but used a Hidden Markov Model (HMM) instead, while García-Bautista et al. [7] used the Microsoft Kinect and Dynamic time warping algorithm.

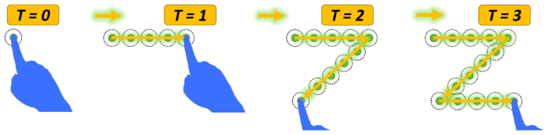

Figure 2: Proposed approach to record a dynamic gesture using a spatial path

Alexander Schäfer et al. [17] created an approach to record and recognize both static and dynamic gestures using the Oculus Quest 2. To recognize static gestures they used the raw data from the HMD and template matching using the Sum of Absolute Differences. To record dynamic gestures, they start by recording the initial static gesture and then record the spatial path by following the movement of a joint or a fingertip as you can see in Figure 2. During the recognition, they recognize the initial static gesture and then they reconstruct the corresponding dynamic gesture's path with invisible 3D points attached with colliders alongside it. If the user's joint or fingertip collides with all those points sequentially until the end it means that gesture was recognized. If the hand's distance to the next point starts increasing and surpasses a certain threshold, then the gesture is discarded.

2.1.3. Sensor-based method

Compared to vision-based solutions, which have higher accuracies in controlled environments, sensor-based solutions are more reliable in real life environments such as in the street or inside a shop. These solutions rely on gloves with in-built sensors on each joint, orientation sensors, accelerometers, magnetometers and gyroscopes.

Lei and Dashun [12] designed their own data gloves based on an ARM9 with 10 flex sensors. Their gloves measured the bending degree of the finger joints and the attitude angle of the palm. The goal was to make a portable device that could do simple sign recognition while being connected through Bluetooth to embedded systems. They used template matching to recognize Chinese sign language gestures. They used five letters from the Chinese phonetic alphabet "a", "b", "c", "zh", "ch" and five Chinese words like "hello", "thanks", "goodbye", "sorry", "ok" for their evaluation. The lowest accuracy rate they got was 83.3% for the letter "a" and 86.7% for the word "hello". The highest accuracy they got was 96.7% for the letter "zh" and 96.7% for the word "ok".

Instead of building their own data glove, Tubaiz et al. [19] used commercial data gloves "DG5-VHand" to develop a continuous Arabic sign language recognition system. They used 80 words to form 40 sentences which were performed 10 times each by a sign language speaker. For the classification algorithm, they chose to do a modified K-Nearest Neighbours (KNN) to make it suitable for sequential data. They obtained an accuracy of 98.9% in sentence recognition and 98.82% in word recognition. They also compared their work against a vision-based system that used the same database and Hidden Markov Model (HMM) for classification and had an accuracy of 75% in sentence recognition [3].

2.2. Text to Sign Language

Text to sign language translator is a useful tool to help hearing people learn how to communicate with deaf people. This language translation system can use one of three different approaches [2]:

- The direct translation approach, where they translate each word in the sentence into its corresponding word translation, ignoring all the syntax, semantics, and morphology of the language.

- The transfer-based approach, where the sentence is analysed in terms of syntax, semantics and morphology of the source language, then goes through a set of transfer rules to transform it from the source language into the target language text representation and then this representation changes according to the syntax, semantics and morphology of the target language.
- The Interlingua-based approach, where the source text is transformed into an interlingua representation, an abstract language-independent representation. Then this interlingua representation is transformed into the target language text.

Since sign language has its own grammar rules, the most used translation approaches are transfer-based and interlingua-based. The most common architecture of a translator has the following structure represented in Figure 3 [1]. Assuming the sentence is orthographically correct, the sentence is tokenized and then these tokens go through a POS-Tagger that assigns parts of speech to each one. In the Name Entity Extraction, the names of entities are identified to be treated differently since names like ”Apple”, which is a company, do not translate into ”apple”, which is the fruit. The next step is the Stemmer which analyzes the suffixes and prefixes of the words and separates them into individual words or just transforms the word into its root form. Then a list of glosses is created using the annotated words and a dictionary in the Lexical Transfer module. The Structure Transfer module is where the glosses are reordered according to the sign language rules. The Animation modules will use these glosses to look up the corresponding animations in a database to perform the gestures.

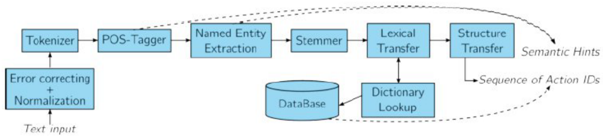

Figure 3: Text processing architecture (from [1])

This architecture is used in PE2LGP, which is a Portuguese text to sign language translator. This software started being developed by Inês Almeida [1] in 2014. She developed the base architecture for the text translation shown in Figure 3, as well as software for the avatar gestures. Then in 2016, Ruben Santos [6] added a module to create gestures manually and add them to the database. In 2020, Matilde Gonçalves [9] refined the translation process. More recently Inês Lacerda [11] improved the avatar's animations, especially the facial animations in order to improve facial expressions and emotions.

Patel et al [15] also used a similar architecture with a few changes in their English Speech into Indian Sign Language (ISL) translator. They reorder the sentences according to the ISL right after the POS-Tagger module. When the sentence is processed they use a transfer-based conversion to generate the ISL words and characters. These words go through the ESIGN editor tool, which has a database of signs with their translation to HamNoSys notation. HamNoSys is The Hamburg Sign Language Notation System, which is a direct correspondence for all sign languages. For the animations, they use these HamNoSys notations to look up the associated Signing Gesture Markup Language (SIGML) files that contain the gestures' animations code. These SIGML files will be used by their avatar to perform the gestures. Bhagwat et al [4] have a very similar system to translate simple Marathi into ISL. The difference of their system is that it is also able to dynamically insert gestures for YN-questions, WH-questions, imperative sentences and negative sentences. Anuja et al [2] built an English to ISL translator using a similar architecture and combined Autodesk with Maya Trax Editor for motion capture to create an avatar and its gestures' animations.

With these works in order to achieve effective augmentative communication we can highlight the following challenges: translate and synthesize text-to-speech and speech-to-text, capture the LGP gesture's hand pose data, classify the LGP gestures, analyse and translate text into LGP glosses and produce legible animated LGP gestures.

3. Approach

To achieve effective augmentative communication we developed an approach to deal with the numerous challenges explained in the previous section. We divided our approach into modules and sub-modules to handle these challenges.

Since we have two different roles, LGP Speaker and non-LGP Speaker, we have an initial Role Selector module where the user can choose their role for the software to know the adequate translation method of the language they are speaking. Then we have the Sign Language to Speech and the Speech to Sign Language modules, to deal with the types of translation necessary for communication. These modules are then divided into sub-modules (see fig. 4).

To exchange the messages from one user to another, we used Photon Unity Networking 2 (PUN2) to develop the chat network.

3.1. Speech to Sign Language

The Speech to Sign Language module is divided into two sub-modules, the Speech-to-Text translator and the Text to Sign Language translator.

The first one is the Speech-to-Text translator, which captures voice input and gives the corresponding input as text, then the Text to Sign Language translator

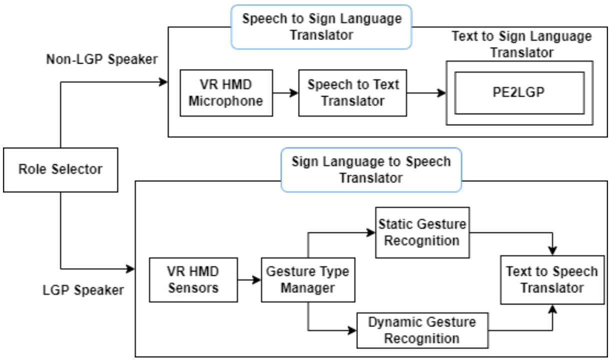
Speech to Sign Language Translator
Text to Sign Language Translator
Non-LGP Speaker
VR HMD Microphone
Speech to Text Translator
PE2LGP
Role Selector
Sign Language to Speech Translator
Static Gesture Recognition
VR HMD Sensors
Gesture Type Manager
Text to Speech Translator
LGP Speaker
Dynamic Gesture Recognition

Figure 4: Approach's architecture

receives that text input and produces the corresponding Sign Language gestures.

3.1.1. Speech to Text

To deal with the translation of speech into text, we used the Microsoft Cognitive Services Speech (MCSS) Software Development Kit (SDK), which is free and supports the Portuguese language. When the user speaks we create a *SpeechRecognizer* object using the MCSS. This object will process any voice input captured by the Oculus Quest 2 microphone for a maximum of 15 seconds and will return the recognized text. Then this text is transmitted to the Text to Sign Language sub-module.

3.1.2. Text to Sign Language: PE2LGP

The Text to Sign Language sub-module is the PE2LGP [11] software mentioned in the Background section. We adapted the software to be used in VR.

The PE2LGP contains three modules to handle the different phases of the Text to Sign Language translation. When the PE2LGP receives the text input, it sends it to an online translator module. This translator receives that text input and processes it using the text processing architecture explained previously in the Background section. Then it outputs a set of translation information containing the translated glosses, phonemes, pauses, facial expressions, intensities, conditions and compound gestures. The Lookup sub-module receives this set and searches the corresponding gestures of those glosses in the gesture database, if a gloss does not have its corresponding gesture in the database, then that gloss is translated by being fingerspelled. Lastly in the Animation sub-module, the avatar receives these gestures a produces their gestures as well as facial expressions.

3.2. Sign Language to Speech

The Sign Language to Speech module consists of three sub-modules, the Gesture Type Manager, the Gesture Recognition and the Text-to-Speech.

To recognize both static and dynamic gestures, we had to develop one Gesture Recognition sub-module for each type. Then to manage them we developed the Gesture Type Manager sub-module which will activate the corresponding Gesture Recognition sub-module according to the selected gesture type. When the selected Gesture Recognition sub-module receives the hand pose data from the Oculus Quest 2, it will output the corresponding text translation to the Text-to-Speech sub-module. This sub-module synthesizes the input text into speech that an avatar speaks, using lip sync, to the non-LGP user.

3.2.1. Gesture Recording

To record static gestures we use the 3D hand pose data from the *OVRSkeleton* class from the Oculus SDK, which provides us with the transform data of 23 joints and fingertips. We perform a feature extraction on this data by transforming the 3D position from world space into the local space of the wrist joint to make the gestures independent of the position they are performed in the virtual space. Then we calculate the distance between each joint and fingertip to the wrist, the distance between one fingertip and the consecutive fingertip and the distance of each fingertip to the thumb tip. The latter is used to help with gestures like ”g” and ”s” that have a very similar pose, where the main difference is the thumb position. For the orientation, we save the distal and intermediate phalanges' *localEulerAngles* and the hand's up, right and forward vectors. These hand pose values are registered in a JSON file and stored in a gesture database.

For the recording of the dynamic gestures, we save the initial static hand pose using the method we just described for static gestures. Then we record the several positions of the index fingertip along the gesture's motion. The first recorded position is treated as the origin point (0, 0, 0) of the gesture's path and we save the following positions using the *InverseTransformPoint* Unity's function, which calculates the local position of a point relative to the origin point. This way the gestures can be recognized regardless of the person's position and orientation in the world. We then register the pair of data *(initial static gesture, gesture path)* in a JSON file and store it in a gesture database.

3.2.2. Gesture Recognition

Regarding the static gestures, we try to match the current hand pose with the gestures from the database in every frame. We do the same process explained in the Gesture Recording section except saving the gesture in the database. Then we compare the values with the gestures from the database using the Sum of Absolute differences. This comparison is divided into two steps. In the first step, we sum the absolute differences of all distances and angles, calculating the finger angles based on the work of Cédric Girardin [8][1] and giving more weight to the thumb's bones and joints values when calculating the difference in values of the distance of each joint and fingertip to the wrist. If the total of this step is lower or equal to a predefined threshold then we move on to the next step, otherwise

[1] https://github.com/cpvrlab/vrTrainingFingerAlphabet

we discard this gesture and proceed to the next one. In the second step, we calculate the angle difference between the wrist's up, left and right vectors of the current hand shape. If any of these differences is higher than a predefined threshold then we discard the gesture since their rotations differ from each other. This is done to avoid recognising gestures such as a palm facing downwards and a palm facing outward as being the same gesture. We sum the total of both steps and if this total is higher than a predefined threshold then we discard the gesture, if it is lower we compare that value we the lowest registered so far. If it is lower than it then we update that value with this new one and save this gesture as the best candidate so far. After iterating over the whole database, we consider the hand pose to be recognized as the gesture that was saved as the best candidate.

For the dynamic gestures, we try to match the current hand shape with a static gesture from the dynamic gesture database in every frame by doing the same process described for the static gestures. The main difference is that instead of only accepting the gesture with the lowest Sum of Absolute Differences, we accept all gestures and create a list with them. We do this because different dynamic gestures may have the same initial static gesture, thus why we want to accept all possible gestures.

Having the list of all possible gestures we then create a Unity GameObject for each gesture. If there is already a GameObject created for a gesture of that list then we skip that gesture. The GameObject has a script with dynamic gesture information such as the gesture's name, the initial position and the several positions of the index fingertip along the gesture's motion from the JSON file. The initial position of the GameObject is the index fingertip's current position and the gesture motion positions are transformed from the local space into world space by using the *TransformPoint* Unity's function, which will calculate the world position of those points relative to the initial position. When the distance between the index fingertip and the GameObject is smaller than a predefined threshold, the GameObject's position is updated to the next position of the gesture motion. When the index fingertip reaches the last gesture motion's position, the gesture is recognized and we delete that GameObject.

3.2.3. Text to Speech

To convert the text into speech we used the Microsoft Azure Cognitive Services SDK for Unity, which has a speech service API to convert text into speech. When the LGP user sends the translated text message, the non-LGP user side receives that text and creates a *SpeechSynthesizer* object using the MCSS. This object will create audio clips from that text and will play them using the avatar present in the scene.

3.3. Augmentative Communication

To enable augmentative communication between the LGP user and non-LGP we used PUN2 to create an online room to exchange text translations between the users.

At the beginning of augmentative communication, the users have a menu where they choose which role they will have during the communication. After having chosen the role, they will change to the corresponding scene. The first user that selects its role will be the one that will create a room, whereas the second user will join that room, using the NetworkManager class from the new scenes for both these operations. Both users have a visual indication on the *"Enviar"* button (LGP side) and the *"Falar"* button (Non-Lgp side), which will be green when the user is connected to a room and red when it is not connected.

When the LGP-speaking user sends a message to the non-LGP-speaking user, that message is sent via Remote Procedure Call (RPC) call to the non-LGP user with the text message and the name of the method which will use it. The non-LGP user side receives this RPC call and executes the method that synthesizes the text into audio clips. Conversely, when the non-LGP user sends a message to the LGP user, that message is sent via RPC call to the LGP user, that receives it in the PE2LGP module, where the text will be translated into LGP gestures.

4. User Evaluation

Our user evaluation aimed to show that two people, one LGP speaker and the other non-LGP speaker, can communicate effectively in a virtual reality environment. We carried out a user study where pairs of participants, an LGP speaker and a non-LGP speaker, were asked to complete three different dialogues. We wanted to show that if one person communicates a message to the other, their partner can clearly understand the message as well as feel the presence of the other person in the environment. Additionally, we wanted to find out if the Oculus Quest 2 would affect negatively the precision and speed of the signs, and if the visual elements present in the environment such as the mirror and the hand models would improve them. We also wanted to know if the avatar "Catarina" was well positioned in the scene, so the LGP speaker could easily understand the gestures she makes. Alongside this user evaluation, we also tested both Text-to-Speech and Speech-to-Text API as well as the Gesture Recognition sub-modules.

4.1. Participants

In the data collecting sessions, we had six participants (1 male and 5 female). Their ages ranged from 22 to 29 years (M = 22,14; SD = 2,81). All participants had a dominant right hand. Only one participant had previous experience with VR. Only one participant suffered some level of hearing loss. Of the six participants, only four knew how to speak LGP. One of them attended a bilingual school, where she learned LGP, while the other three were LGP interpreters. The other two participants did not have any prior knowledge about LGP gestures and were taught during the data collecting sessions.

In the user evaluation, we had four pairs of participants (4 male and 4 female), four non-LGP speakers and four LGP speakers. Their ages ranged from 22 to 54 years (M = 32,75; SD = 12,35). Three participants had a basic education level while five participants had a university education level. All participants had a dominant right hand. Only one participant had previous experience with VR. Only one participant suffered some level of hearing loss. Of the four LGP speakers, only two knew how to speak LGP. One of them attended a bilingual school, where she learned LGP, while the other one was an LGP interpreter. The other two LGP speakers did not have any prior knowledge about LGP gestures and were taught during the experimental sessions.

4.2. Procedure

Our user studies included two phases, one for gathering gesture data for the database and the other for evaluating our approach.

4.2.1. Data Collection Session Procedure

For the data collecting sessions, we were able to recruit six participants, four LGP interpreters and two people that did not have prior knowledge of LGP.

All sessions followed the same structure and lasted for about two hours. At the beginning of every session, participants were introduced to the experiment by receiving some context about their role in the experiment, the reason for the experiment, the goals of the experiment and its planning. After that, we would do a quick Q&A so the participants could ask any questions about the experiment to clarify their doubts. Then participants would fill out a consent form, a COVID-19 form and a profile questionnaire where we asked for information about their gender, age, education level and dominant hand. We would also ask if they had any previous experience with VR HMD and if they possess any kind of hearing loss. Before recording gestures, we spent about 10 minutes teaching the participants how the Oculus Quest 2 and its hand tracking works, we also explained what they needed to do during the recording session and recorded some sample gestures to get them used to the HMD and the experiment workflow. The participants that were not an interpreter but were going to assume the role of an LGP speaker in the experiment were taught the gestures during this period.

During the experiment, participants performed the gestures which needed to be recorded while having five minutes pauses between every three gestures. These pauses were necessary due to the high ambient temperatures during the summer season along with the heat that participants felt using the Oculus Quest 2 for a long period. Furthermore, since most of the participants were new to the Oculus Quest 2, some of them would feel sick after a few minutes of usage, thus the necessity of these breaks.

4.2.2. User Test Session Procedure

For the user test sessions, we were able to recruit eight participants, four LGP speakers, two LGP interpreters and two people that were taught during the gesture recording, and four non-LGP speakers.

All sessions followed the same structure and lasted for about one hour. The beginning of this experiment is similar to the data collecting experiment section 4.2.1, participants were introduced to the experiment followed by a quick Q&A about it in the same way we did for the data collection session section 4.2.1. They were then asked to fill out a consent form and a COVID-19 form. Before beginning the experiment, we spent about 10 minutes teaching the participants how the Oculus Quest 2 and its hand tracking works, we also explained how the scenes for their corresponding roles worked, by teaching the function of each button, how to press them and how to execute the tasks necessary for the augmentative communication test. Then we gave them some time to get used to the HMD, the buttons and the experiment workflow.

During the experiment, the participants went to different rooms and started by choosing their roles in a role selection scene. Then they performed each dialogue with five to ten minutes breaks in between each dialogue in light of the explanation we gave in section 4.2.1. At the end of the experiment, the participants filled out a profile questionnaire equal to the one used in section 4.2.1 and a feedback questionnaire about the experiment.

4.3. Evaluation Metrics

During the user tests, we gathered data to evaluate all modules of our approach. To evaluate the Speech-to-Text and the Text-to-Speech translators we measured their Translation Error Rate (TER) values to measure the quality of their translations since they have a significant influence on augmentative communication.

For the sign to speech translator, in the feedback questionnaire, we asked the participants whether they agreed or disagreed with some statements regarding this module to evaluate its quality and took notes of gestures that were not recognized during the user tests. We asked if the HMD negatively affects the gesture's velocity or precision and if the visual elements, mirror and hand models, helped in the execution of the gestures. We also tested if the Gesture Recognition sub-modules were able to recognize the following letters and digits: A, C, D, E, I, L, N, O, P, Q, R, T, U, V, 2, 4, 9. While for the dynamic gestures we tested the following gestures: *"Olá"* (Hello), *"Bom* *dia"* (Good morning), *"Prazer em conhecer"* (Nice to meet), *"Precisar"* (Need), *"Ajudar"* (Help), *"Quanto custa ...?"* (How much ... cost?), *"Cadeira"* (Chair), *"Obrigado"* (Thank you), *"Não"* (No), *"Sim"* (Yes), *"Ir"* (Go), *"Lisboa"* (Lisbon).

Regarding the speech to sign translator, we asked in the feedback questionnaire if the gestures of the avatar "Catarina" were legible and if the avatar was well positioned in the scene, having the hand gestures and facial expressions clearly visible.

To evaluate augmentative communication, we counted the number of times a participant asked the other participant to repeat what they said, when the number of repetitions was above five we considered that dialogue failed and that augmentative communication was not possible in that dialogue. When one of the participants was not able to communicate the correct message after trying five times we also considered that dialogue failed. In case of a failed dialogue, we still carried out the dialogue until the end to measure the individual performances of both translators. When both the LGP speaker and the non-LGP speaker successfully performed two of these three dialogues, we considered that our approach provided means for effective augmentative communication. In the feedback questionnaire, regarding their feeling of co-presence, we asked if participants felt present in the virtual environment, if they felt the presence of their partner in the virtual environment, if they understood their partner's communications clearly, if they felt their communications were being transmitted to their partner without suffering any modifications and if they felt they participate in clear and coherent dialogues.

4.4. Results

Having completed the experimental sessions, we calculated the TER values of both speech services APIs. The Text-to-Speech translator had a TER value of 0%, while the Speech-to-Text translator had a TER value of 3.14%, having misrecognized the sentence *"Qual é tua idade?"* (What is your age?) as *"Tua idade."* (Your age) in two experimental sessions and the sentence *"Onde vais?"* (Where are you going?) as *"Vais."* (You go) in one experimental session.

During the evaluation sessions, we annotated which letters and digits were not recognized by the Sign Language to Text translator. The letter "A" was the only letter to not be recognized in three experimental sessions where it was tested, while the digit "4" was incorrectly recognized as the letter "F" when tested in one experimental session. The letter "R" was tested in three experimental sessions and was recognized in two of them. The letters and digits (C, D, E, I, L, N, O, P, Q, T, U, V, 2, 4, 9) and all dynamic gestures were recognized in all experimental sessions where they were tested.

For augmentative communication, the first dialogue (Basic greetings) failed in three experimental sessions because LGP-speaking participants were not able to communicate their names, given that the static gesture recognition module was unable to recognize the letter "A". In one of the experimental sessions, the LGP-speaking participant was unable to communicate their name and age, which had the letter "A" and the digit "4". However, the participants were able to carry out the other two dialogues until the end successfully.

4.4.1. User feedback

As described in section 4 we used feedback questionnaires containing a set of statements to be answered using a 6-point Likert-scale ranging from Strongly Disagree (1) to Strongly Agree(6).

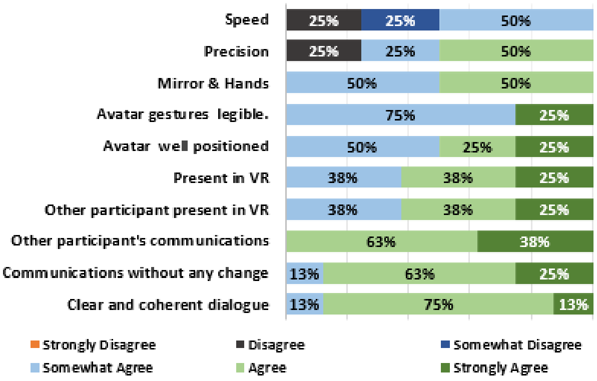

Fig. 5: User studies Likert-scale results

Regarding the feedback about the limitations the Sign Language to Speech imposes on the user when asked if the HMD negatively affected the gestures' speed, the answers were inconclusive since half LGP-speaking participants disagreed with that statement and the other half agreed with it (see fig. 5 first bar). When asked if the HMD negatively affected the gestures precision, three of four LGP-speaking participants agreed with this statement, while the other LGP-speaking participant disagreed (see fig. 5 second bar). All LGP-speaking participants agreed that the visual elements such as the mirror and hand models helped during the execution of the gestures (see fig. 5 third bar).

Regarding the feedback about the adaptations we did to the PE2LGP avatar, all LGP-speaking participants agreed that the avatar gestures were legible and that it was well positioned in the scene since they had no difficulty seeing its gestures and facial expressions (see fig. 5 fourth and fifth bars).

The feedback results about augmentative communication were positive (see fig. 5 from the sixth to the tenth bar). All participants agreed that they participated in clear and coherent dialogues, they felt that their communications were being transmitted to the other participant without suffering any modifications and understood the communications, sent by the other participant, clearly. Regarding the feeling of co-presence, they also felt they were present in the virtual environment as well as the presence of the other participant.

4.5. Discussion

Considering the results obtained with the experimental sessions, we can note that both speech services APIs have low TER values, meaning that their outputs will rarely affect the augmentative communication negatively.

Regarding the Sign to Speech translator, the only static gestures that were not recognized were the letter ”A” and the digit ”4”. The letter ”A” was not recognized, because some fingers are occluded from the HMD field of view, leading to imprecise values for those finger positions. The digit ”4” was incorrectly recognized as the letter ”F” because the hand pose is similar to the letter ”F” hand pose and some fingers also suffer some occlusion (see Figure 6). The letter ”R” was not recognized in one of the experimental sessions also because of the occluded fingers and possibly because of the lighting in the room, since the gesture was recognized in the other two experimental sessions. For the dynamic gestures, all tested gestures were recognized in the experimental sessions. Considering the results from the feedback questionnaire, the two LGP interpreters believed that the HMD did not affect negatively the gestures speed, while the other two LGP-speaking participants, who were taught the gestures before the experiment, believed that the HMD did affect negatively the gestures speed. We assume that since LGP interpreters are less dependent on their vision and more familiar with the gestures, the HMD did not hinder their gesturing speed, whereas for the participants that were thought the gestures before the experiment the HMD decreased their field of view and hindered their gestures speed. Regarding the gestures precision, while using the HMD, three of the four LGP-speaking participants believed that the HMD did affect negatively the gestures precision. This was due to the imprecisions that happened with the hand tracking software when some of the fingers were occluded or when the HMD would lose track of the hand due to some sudden movement. All participants agreed that the hand models and mirror were helpful during the execution of the gesture and that they provided a real-time representation of the hand shapes and movement, being especially helpful in the dynamic gestures.

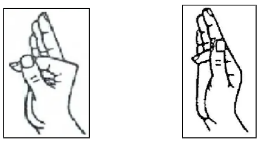

Fig. 6: Comparison between the gesture ”4”, on the left, and the gesture ”F”, on the right

Regarding the Speech to Sign translator, the LGP-speaking participants agreed that the Avatar ”Catarina” was well positioned in the scene and the hand gestures and facial expressions were visible and legible.

For augmentative communication, in three of the four experimental sessions, two of the three dialogues were successful, while in the other one the three dialogues were successful. In the feedback questionnaire all participants agreed they had a sense of co-presence, they felt present in the virtual environment, received the communications and understood their information clearly and felt their message was being transmitted with no modifications. Considering these results we can prove our initial research statement, given that our approach showed it was capable of providing effective mechanisms to enable augmentative communication between non-LGP and LGP speakers in Extended Reality.

4.5.1. Limitations

The proposed approach showed promising results taking into account the hardware and algorithms limitations. Most of the alphabet and digits LGP's gestures have parts of the fingers occluded leading to poor hand tracking data, this issue gets more noticeable with dynamic gestures since the majority of the dynamic gestures have both hands interact with each other and with the face.

At the beginning of the training sessions of the experimental sessions, the LGP interpreters were making each dynamic gesture without any breaks in between movements. However, our approach requires each dynamic gesture to be done individually and confirmed, slowing their communication speed.

In our approach, we were not able to use avatars that represented the user's gender correctly, given that the PE2LGP only had the female avatar ”Catarina” to execute the LGP gestures, and the Oculus Integration SDK's female avatar was the only one that we could find with lip-sync for the Portuguese language.

One of the main problems we faced during this work was the lack of LGP-speaking users that could help us build a bigger gesture database and to participate in the user studies. Our initial plan was to recruit deaf people and LGP interpreters, however, it was surprisingly challenging to recruit people willing to participate in our user studies, given that deaf people dislike seeing and working with avatars because of their rigid animations, lack of facial expressions and they make them feel uncomfortable. On the other hand, interpreters also do not like to work with avatars and sign language translation systems as they are afraid these will replace them in the near future, leaving them unemployed. We believe that with a diversified set of LGP-speaking users and a big gesture database, the use of machine learning algorithms can improve significantly the results of gesture recognition.

5. Conclusions and Future Work

In this dissertation, we presented a proof of concept towards augmentative communication, consisting in the communication between an LGP speaker and a Non-LGP speaker. The LGP speaker's gestures are recognized and translated into text and later on into speech while the Non-LGP speaker's speech is translated into text and later on into LGP gestures performed by an Avatar, making it possible to have basic interactions between two different languages.

We conducted a user study with eight participants to find if our approach could provide effective mechanisms for enabling augmentative communication between non-LGP and LGP speakers in Extended Reality, by performing three different dialogues between the two types of speakers. Results showed that XR can be used for augmentative communication, although we found some challenges in translating the letter ”A” and similar letters and numbers such as ”4” and ”F”. The communication of words and sentences using dynamic gestures showed to be possible. The results also revealed that it was possible to have clear and coherent dialogues and our approach provided effective mechanisms enabling augmentative communication between an LGP speaker and a non-LGP speaker.

Future work could address these challenges by creating an LGP to Portuguese translator similar to what the PE2LGP does but in the opposite direction of its pipeline. Replacing the avatar with videos of interpreters performing LGP gestures and making the approach more acceptable to deaf people. The new VR HMDs with RGB passthrough capabilities can be used to improve the gesture recognition performance by combining the appearance-based and 3D model-based approaches.

We believe that our approach can be further developed to be applied to all the different sign languages that exist. With the recent increase in research on VR conferences, meetings and remote work, our approach provides the initial steps for the inclusion of deaf people in those environments without the necessity of the presence of a human interpreter. It can also be applied to VR games that want to include sign language-speaking non-playable characters, sign language players' inputs and the possibility of multiplayer games where sign language-speaking players and non-sign language-speaking players can communicate. These future developments lead us to believe that Extended Reality technology can indeed provide for more inclusive and accessible environments.

References

[1] I. Almeida. Exploring challenges in avatar-based translation from european portuguese to portuguese sign language. Master's thesis, Instituto Superior Técnico, October 2014.

[2] K. Anuja, S. Suryapriya, and S. M. Idicula. Design and development of a frame based mt system for english-to-isl. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 1382–1387, 2009.

[3] K. Assaleh, T. Shanableh, M. Fanaswala, F. Amin, and H. Bajaj. Continuous arabic sign language recognition in user dependent mode. *JILSA*, 2, 01 2010.

[4] S. R. Bhagwat, R. P. Bhavsar, and B. V. Pawar. Translation from simple marathi sentences to indian sign language using phrase-based approach. In *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 367–373, 2021.

[5] T.-W. Chong and B.-G. Lee. American sign language recognition using leap motion controller with machine learning approach. *Sensors*, 18(10):3554, 2018.

[6] R. Emanuel, R. dos Santos, and J. A. M. Pereira. Pe2lgp: From text to sign language (and vice versa). Master's thesis, Instituto Superior Técnico, 2016.

[7] G. García-Bautista, F. Trujillo-Romero, and S. O. Caballero-Morales. Mexican sign language recognition using kinect and data time warping algorithm. In *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 1–5, 2017.

[8] C. Girardin. Vr-trainingsapplikation-app für fingeralphabet. Technical report, Berne University of Applied Science, 2020.

[9] M. Gonçalves, L. Coheur, H. Nicolau, and A. Mineiro. Pe2lgp: tradutor de português europeu para língua gestual portuguesa em glosas. *Linguamática*, 13:3–21, 2021.

[10] H. Kaur and J. Rani. A review: Study of various techniques of hand gesture recognition. In *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pages 1–5, 2016.

[11] I. Lacerda. Generating realistic sign language animations. Master's thesis, Instituto Superior Técnico, Oct. 2021.

[12] L. Lei and Q. Dashun. Design of data-glove and chinese sign language recognition system based on arm9. In *2015 12th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, volume 03, pages 1130–1134, 2015.

[13] C. Li, X. Zhang, and L. Jin. Lpsnet: a novel log path signature feature based hand gesture recognition framework. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 631–639, 2017.

[14] M. Panella and R. Altilio. A smartphone-based application using machine learning for gesture recognition: Using feature extraction and template matching via hu image moments to recognize gestures. *IEEE Consumer Electronics Magazine*, 8(1):25–29, 2019.

[15] B. D. Patel, H. B. Patel, M. A. Khanvilkar, N. R. Patel, and T. Akilan. Es2isl: An advancement in speech to sign language translation using 3d avatar animator. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–5, 2020.

[16] N. Pereira. Tradução vídeo de língua gestual portuguesa: Reconhecimento dinâmico de configurações de mão. Master's thesis, Instituto Superior de Engenharia do Porto, Oct. 2020.

[17] A. Schäfer, G. Reis, and D. Stricker. Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications. *Applied Sciences*, 12(4), 2022.

[18] S. Shrenika and M. Madhu Bala. Sign language recognition using template matching technique. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–5, 2020.

[19] N. Tubaiz, T. Shanableh, and K. Assaleh. Glove-based continuous arabic sign language recognition in user-dependent mode. *IEEE Transactions on Human-Machine Systems*, 45(4):526–533, 2015.

[20] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak. Recognition of american sign language gestures in a virtual reality using leap motion. *Applied Sciences*, 9(3), 2019.

[21] Y. Zhu, Z. Yang, and B. Yuan. Vision based hand gesture recognition. In *2013 International Conference on Service Sciences (ICSS)*, pages 260–265, 2013.