FAITH: A Desktop Virtual Reality System for Fingerspelling

Selan R. dos Santos
selan@dimap.ufrn.br

Leonardo Bezerra
leonardo@lcc.ufrn.br

Antonino A. Feitosa Neto
antonino@lcc.ufrn.br

Departamento de Informática e Matemática Aplicada,
UFRN, 59072-970, Natal/RN, Brazil

Silvano Maneck Malfatti
malfatti@lncc.br

Laboratório ACiMA, LNCC, Av. Getúlio Vargas, 333,
25651-075 Petrópolis/RJ, Brazil

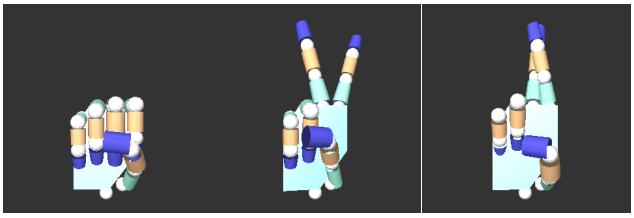
Abstract

Fingerspelling is a vital component of the Linguagem Brasileira de Surdos (LIBRAS), which is the preferred form of communication among the hearing-impaired people. Achieving fingerspelling fluency relies heavily on the visual comprehension of the manual representation of letters. Of particular importance are the transitions between letters. We present the development of a fingerspelling learning tool based on desktop virtual reality technology that places emphasis on realism through the principles of 3D animation with a simple user interaction interface. We aim at producing animations corresponding to the LIBRAS fingerspelling that appear natural and can be quickly and reliably recognized by members of the hearing impaired community. We tested our proposed system, called FAITH*, by means of two experiments. The first focused on measuring the tool's efficiency for the task of unsupervised teaching, while the second evaluated the tool's accuracy in supporting communication among people who do not know fingerspelling and those who do.*

1. Introduction

Virtual Reality (VR) technology is an valuable asset for helping people suffering from various types of disabilities. VR has been successfully applied to areas such as rehabilitation [1, 15], therapy [8, 7], assistance to visually impaired people [9, 4], and education in deaf community [5].

Within the context of education for the hearing impaired, *fingerspelling* or dactylology is one of the basic forms of communication. In this type of conversation, a signer uses certain handshapes to represent the characters of a given alphabet, or the numbers. This mechanism of communication is useful, for example, to spell proper nouns, acronyms, technical terms, foreign language words, etc., as shown in Figure 1. Learning how to fingerspell fluently is a valuable skill, and it is usually considered as a independent component of a sign language. In Brazil, the LIBRAS[1] sign language is the preferred mean of communication among the deaf community.



(a) 'S' **(b)** 'V' **(c)** 'R'

Figure 1: A sequence of screenshots spelling this symposium's acronym "SVR".

Signers fluent in LIBRAS use a combination of hand and arm positioning and orientation, hand motion, as well as facial expressions (e.g. mouthing and eye gazing). Normally, each country has its own sign language, often derived from other sign languages. LIBRAS, for instance, was based on the French sign language LSF[2] and is regarded by the current legislation as a language on its own right, with its own grammatical structure.

Traditionally, the standard method to learn fingerspelling and LIBRAS is to observe the signs made by a certified signer/teacher, repeat those signs, and receive a feedback

[1] *Linguagem Brasileira de Sinais.*
[2] *Langue des Signes Française*, http://cis.gouv.fr/

indicating whether the signs have been executed correctly. However, the volume of students wanting to learn LIBRAS far outnumbers the quantity of certified teachers available. An alternative teaching method is to use conventional dictionaries and/or reference books. Unfortunately, this approach is not as efficient as a real teacher, because dictionaries and books might convey inaccuracy when representing motion in a static (printed) form.

One way of tackling the above limitation is to produce visual tools that allow students to exam an animated sign and be able to repeat the animation as many times as necessary to fully learn its movements. The use of pre-recorded digital video, either streamed online[3] or as a DVD, is a typical example of such approach. However, this strategy faces the following problems: i) the sign is always watched from the same point of view and presented only as individual signs, i.e. there is no smooth transition from one sign to another; ii) part of the hand might be obstructed by the signer's own hand or arm; and iii) some signs are fairly complex and need to be examined closely and from different perspectives if it is to enable learners to understand handshapes properly.

To address the issues discussed above, we have investigated how virtual reality technology could serve as the basis for a tool aimed at supporting the generation of signs made by an artificial human hand. We have set out three design requirements for our tool. Firstly, it should generate real-time animation of signs, with a continuous passage from one sign to the next. Secondly, it should make use of simple graphics resources and be able to run on commodity PCs without dedicated graphics hardware. Finally, the system should allow users to interactively exam the hand model via rotations, and control both the speed and direction (i.e. forwards/backwards) of the animation rendered.

This paper presents a desktop virtual reality system called *FAITH* (*F*ingerspelling Computer-*AI*ded *T*eac*H*ing Tool) for automatic fingerspelling. Our objective is twofold. First, to offer a *teaching tool* to aid students learning the movements involved in fingerspelling. Second, to provide an interactive *communication tool* that lets a person who does not know a sign language to type in letters to be translated into hand movements, thereby aiding communication with people who understand fingerspelling.

To evaluate *FAITH* system, we have invited experienced signers to analyze the accuracy of the fingerspelling movements. We also organized a between-subjects experiment design to investigate the learning experience, comparing the use of digital video data with *FAITH* system in terms of learning speed and precision in sign recognition.

The next section briefly describes the previous work related to sign language animation. Then, we shall describe FAITH system, detailing the system's design, the animation generation process and interactive features. The final sections describe the evaluation design, present the results and concluding remarks.

[3]See, for example, `http://www.libras.org.br/`

2. Related Work

Most of the work related to the use of animation for sign language faces a common challenge: to produce a smooth and realistic animation of handshapes that are as close as possible to the movements of a real human hand. The major problem here is collision avoidance during finger motion, and gradual transformation from one handshape to another. The latter aspect should try to avoid the introduction of glitches in finger movements, such as fingers passing through one another or bending in an unnatural fashion.

Commonly, there exist two approaches to implement animation of signs in an automatic signing system [3]: to store signs as a sequence of digital frames, and; to represent a sign as a set of parametric data allowing the system to generate a sign movement in real-time using interpolation between key frames.

The work done by Wolfe *et al.* [13], for instance, addresses the problem of realistically displaying fingerspelling by implementing a core display engine that uses video clips created in a commercial animation package and store them as AVI files. They tackled the problem of smooth animation between pairs of letters by generating a total of 272 clips, one for each possible pair of character transition. The main problem with this type of approach is that the user is limited to observing the animation from the same point of view. Consequently, it is not possible, for instance, to zoom in and exam, say, an intricate configuration of fingers from different points of view or even rotate the model to see the animation from a different point of view.

Sedgwick *et al.* [10], on the other hand, use hand control to define key frame positions for a fully articulated hand and interpolate the rotations of the joints. They did not make use of inverse kinematics to control joints movement because it might yield unpredictable and unnatural hand movements. They also argued that the use of collision detection algorithms would produce unnatural solutions and the constraints of a human hand would not be accounted for in a general collision detection algorithm.

Yeates *et al.* in [14, 5] present the Auslan Jam, a system to generate graphical sign language using a half body avatar. The main focus of their research was to compare the use of two different types of orientation representation for the joints movements: Euler angles and quaternion. They have found that quaternion representation is more reliable than Euler representation because of the better results in animation obtained with the slerp (spherical linear interpolation) interpolation algorithm if compared to the ones generated with linear interpolation of Euler angles.

3. The FAITH System

Learning the LIBRAS sign language or fingerspelling is a complex task. There are many subtleties on each movement that makes them unique. The correct arm positioning, the precise mouthing, the accurate rotation angle, all these variables make it difficult for one to learn it by oneself. Therefore, the idea behind FAITH system is to allow students to take their time while analyzing each movement, seeing it through different angles, and repeating the animation as many times as they might need.

We have decided to initially focus our interest on fingerspelling alone, because this is a problem relatively small if compared to the animation of a full-body avatar, capable of facial expressions, complex and coordinated movements of hand, arms and head. Nonetheless, both problems share similarities. For example, the two types of animation (hand and body) can be modeled by a hierarchy of articulated joints. This close proximity between these two instances of the same problem allowed us to design a general solution that may be later extended to include the representation of the LIBRAS language.

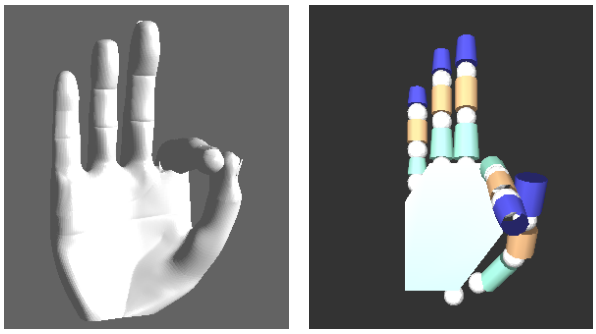
3.1. Hand Modeling

One of the challenges in designing the hand model for *FAITH* system was that we had to find the correct balance between realism and a good model representation. By realism we mean a model that is very similar to the shape of a regular human hand, presents the same type of hand movements, and exhibits the right fingers proportion. All these features should be combined in such a way that encourages users to accept the model and, at the same time, feel comfortable when imitating the movements generated by the system. By good model representation we are referring to a model that would have a controlling hierarchical skeleton associated with, and would no require a considerable amount of memory to run.

Throughout *FAITH*'s development, two types of hand models were created; a realistic and an abstract, as show in Figure 2. The hand is controlled by a simplified hierarchical structure of nodes, each of which associated to the main joints of the human hand anatomy.

The hierarchical representation of joints behind both models has 16 joints, as show in Figure 3. Each individual finger has three types of joints, the metacarpophalangeal, the proximal interphalangeal, and the distal interphalangeal joint. The thumb has also three joints, the carpometacarpal, the metacarpophalangeal, and the interphalangeal joints. We believe our model satisfactorily approximates a real human hand.

Each individual joint is organized in a hierarchical structure, having the wrist as the 'root' and the joint close to the fingertips (i.e. the interphalangeal joint) as the tree's 'leaves'. Therefore, when a geometric transformation is applied to a joint up in the tree structure it affects all its descendants down to the 'leaves'. This effect resembles the application of a transformation in a scene graph except that the transformation is propagated from the 'root' to the 'leaves', in a depth-first traversal of the tree. Table 1 summarizes the list of joints and their respective degrees of freedom (DOF).



(a) (b)

Figure 2: Types of hand model used in *FAITH* system. In both cases the handshape represents the 'F' letter.

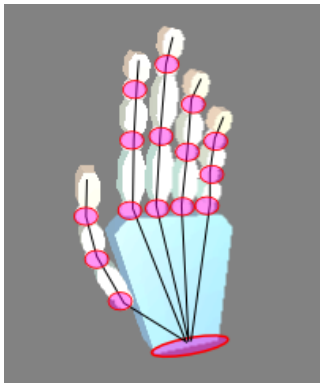
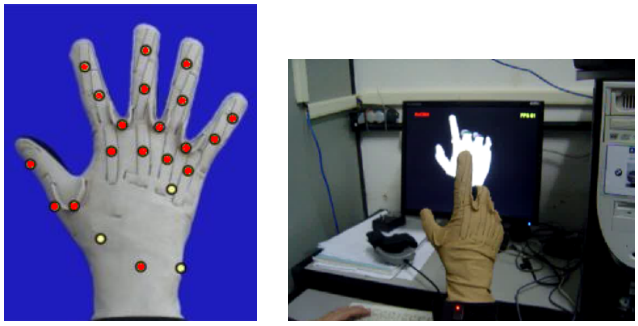


Figure 3: The hierarchical joint representation behind the hand models.

Note that each joint at the base of the four fingers and the thumb has two degrees of freedom, which allows them two types of movements: to move from a fully open position to a fully bent position (respectively, extension and flexion), and to move sideways ("wiggling"). The rest of the joints has only one degree of freedom. The joint at the wrist has three degrees of freedom, which affords the same movements as a base-finger joint (i.e. open-close and sideways) in addition to the rotation of the hand along its main axis.

Table 1: List of the joints of a hand (and their common abbreviations) and the corresponding degrees of freedom.

Part	Joint name	DOF
Thumb	Thumb Interphalangeal (TIP)	1
	Thumb Metacarpophalangeal (TMP)	1
	Trapeziometacarpal (TMC)	2
Fingers	Distal Interphalangeal (DIP)	1
($\times 4$)	Proximal Interphalangeal (PIP)	1
	Metacarpophalangeal (MCP)	2
Palm	Metacarpocarpal	3
Total		23



(a) (b)

Figure 4: Using data glove to capture key handshapes. (a) shows the location of the 22 sensors on the Immersion Wireless CyberGlove II. (b) presents a picture captured while the keyframe acquisition system was running and had captured the final key handshape corresponding to an 'L'.

3.2. Keyframe Capture

The alphabet coded in the *FAITH* system is displayed to users via handshapes. A handshape is determined by a set of joint angles, and some of them include an animation of the entire hand, such is the case of letters 'X', 'K', and 'Z'.

One of the system requirements (c.f. Section 1) was to render animation with smooth change from one handshape to the next. Another consideration was that we should avoid movements that lead to finger interpenetration, since this would represent a situation impossible for learners to reproduce. To comply to these requirements, we have decided to use keyframe animation [12].

Keyframe animation has been chosen for two reasons. First, it lets us specify intermediate handshapes in such a way that the complete animation avoids entirely finger interpenetration. Second, animation between intermediate handshapes is done through simple linear interpolation of individual joint angles.

We have implemented a separate system to capture the keyframes (or key handshapes) that comprise each alphabet sign. The keyframe acquisition system was developed with the EnCIMA virtual reality engine [6]. This engine supports the Immersion Wireless CyberGlove II[4], and renders our hand model in real time. This data glove has a total of 22 sensors, located at key position of the hand, as show in Figure 4a. The two glove sensors located at the base of each finger had to be combined so that they could be mapped to the 16 joints of our hierarchical hand skeleton model. Other glove sensors were simply ignored, since they do not fit in our hand model.

A specialist signed each handshape, stopping at key positions that were recorded by the system and stored as a set of XML files. Figure 4b illustrates this procedure, while the Code 1 contains (part of) a typical XML output. In this code fragment all nodes have three coordinates, (x, y, z), that determine the joint angle associated to the local coordinate system, located at the knuckles, as shown in Figure 5. Note also, that the parent-child relation is indicated by the "parent" field.

Code 1 Part of a XML file storing a keyframe of the 'V' handshape. A handshape is represented by a tree of joints, each of which having 1 our 2 degrees of freedom. Angles are specified according to a local system of coordinates.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<scene id="0" name="'V' Handshape">
  <comment>Test scene comment</comment>
  <node id="0" group="hand" name="base">
    <cartesian x="+15.0" y="+10.0" z="5.0"/>
  </node>
  ...
  <node id="10" group="index" name="TMC" parent="0">
    <cartesian x="3.0" y="0" z="-13.0"/>
  </node>
  <node id="11" group="index" name="TMP" parent="10">
    <cartesian x="1.0" y="0" z="0"/>
  </node>
  <node id="12" group="index" name="TIP" parent="11">
    <cartesian x="1.0" y="0" z="0"/>
  </node>
  <node id="13" group="thumb" name="MCP" parent="0">
    <cartesian x="20" y="110.0" z="-1"/>
  </node>
  <node id="14" group="thumb" name="PIP" parent="13">
    <cartesian x="60.0" y="0" z="0"/>
  </node>
  <node id="15" group="thumb" name="DIP" parent="14">
    <cartesian x="30" y="0" z="0"/>
  </node>
</scene>
```

3.3. System Architecture

The *FAITH* system consists of four modules: core, render, animation, and interface. Next we shall briefly discuss

[4] http://www.inition.co.uk/, last access on 15/02/2009.

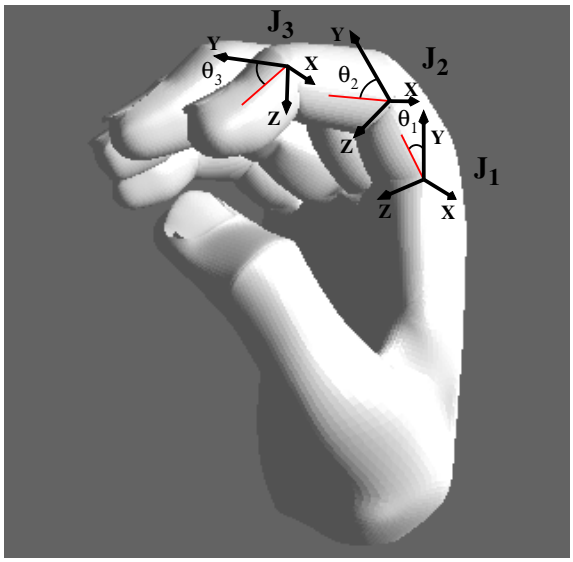


Figure 5: Illustration of the local coordinate systems associated with the joints of the index finger. There are 3 joints, J_1, J_2, and J_3, each showing that the phalanges are bent at positive angles, respectively Θ_1, Θ_2, and Θ_3. These angles are measured between Y and Z axes, thus representing 1-DOF.

each of this modules, highlighting their function and implementation.

The system has been implemented in platform-independent C++, and has been developed following object-oriented design principles. Many system components are designed to accept custom extensions, by means of subclassing existing components to offer further functionality. The rendering is done via OpenGL [11].

3.3.1 Core Module

The *core module* controls the whole system and exchanges information among the other modules. It basically creates instances of classes associated with each module and manages memory resources.

Additionally, the core module manages the time involved in the animation. We have decided to separate the rendering from the simulation loop. This enables the system to run the animations at the same speed, regardless of the processing power of the machine on which the system is running. This principle is often used in dynamics simulation found in computer games; it steps the simulation (in our case the animation) forward by a fixed amount of time, drawing the current stage after each of these steps. A time accumulation scheme is utilized to make sure that simulation and rendering are synchronized and run at same speed [2].

3.3.2 Render Module

The *render module* is responsible for the three-dimensional (3D) rendering of the animations, either using the abstract or the realistic hand models. The basis of the rendering is a tree data structure that represents the joint hierarchy associated with the hand. The advantage of using a tree hierarchy is obvious: all geometric transformations (mainly rotations) applied to a parent note automatically propagate its effect to all of its descendants.

To calculate the positions of joints in 3D space for rendering, joints are individually modeled by storing a fixed positional offset from the immediate parent joint — this is calculated based on average length of an adult hand. A local coordinate system is defined at the base of each joint, according to the right-hand rule. When all the joint angles are set to zero the hand assumes a position similar to the stop sign, henceforth called "neutral" hand position.

Each node within the tree structure is associated with either a polygonal model (for the abstract version of hand) or a sub-mesh (for the realistic version of hand). Also, a node contains a geometrical transformation matrix, and the node's current state (e.g. the current animation angles) that is updated each simulation step.

Rendering is done by traversing the tree following a depth-first search from the root node; each node model is rendered by transforming it to the node's local coordinate space.

3.3.3 Animation Module

The *animation module* handles the actual interpolation from one intermediate handshape to the next, within a sequence of handshapes corresponding to a letter animation. We have used linear interpolation of joint angles. As a result, we are able to maintain all animations smooth and simple.

The length of time for an animation can be set to either regular or slow modes, respectively 2 and 4 seconds. Animations can be stopped any time, and played in reverse.

The important feature of this module is the realization of a smooth transition from one letter to any other, without needing to reset to the "neutral" hand position. This is accomplished by considering the current handshape, H_c, and the a target handshape, H_t. Then, the module determines which handshape, among those that form the set of key-handshapes of H_t, is "closer" to H_c. By "closer" we mean an average measure of Euclidean distance among corresponding joint nodes. Next, the interpolation is done to the selected "closest" handshape of H_t, and from there to the next handshape in the set of H_t.

This later feature is relevant since most of the traditional individual learning systems only animates individual letters. This way, students have always to learn a letter starting from the "neutral" hand position, which hinders fluent signing.

3.3.4 Interface Module

The *interface module* takes care of user input, loads and parses all the XML files corresponding to the alphabet. In fact, one of the most important resources available in the FAITH system is the interaction capabilities, such as real time rotation of the viewpoint, or the animation playback in forward or backward motion.

Our intent was to design an user interface as simples and uncluttered as possible, since its potential users may not be accustom to computers. Initially, the hand model is displayed and remains in the "neutral" hand position. Signs are, then, selected by simply pressing the corresponding letter on the keyboard. If the handshape is positioned in such a way that the user is unable to see the hand model, or one just wants to examine from a different point of view, it is possible to rotate the model about Y axis, Z axis, or combination thereof. This is done by just dragging the mouse with the left button pressed. The hand may be reset to the "neutral" position by pressing the *enter* key.

For those signs that include movements of the whole hand (e.g. 'H', 'K', 'X', and 'Z'), the animation comprises two stages. In the first stage, when a letter is hit on the keyboard, the hand model moves into the initial handshape. Next, all the other keys are disabled, leaving the user with only two possibilities: to press either the reset key, or the key that started the animation. In this case a number appears on the top-right corner, indicating that the chosen letter has a secondary animation (see Figure 6).

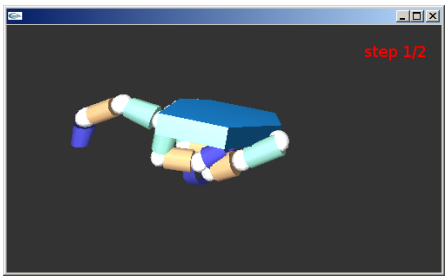
3.4. System Calibration

After the end of the keyframe acquisition phase, we proceeded to invite experienced signers to calibrate the handshapes, fine-tuning the angles between joints, finger orientation, and secondary animations.

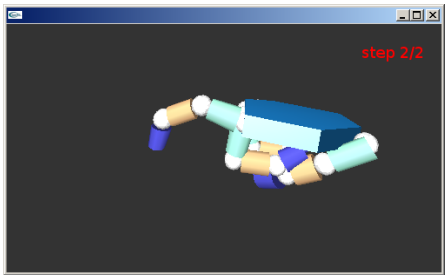
The system calibration is an important stage, since deaf and hearing impaired people tend to be very precise and accurate about signing, and a simple detail that might seem insignificant to hearing people would probably be pointed out by these particular group. Therefore, our objective was twofold, to improve the accuracy of sign representation, and ascertain whether the communication aspect of the tool was effective for the hearing impaired students.

This calibration stage took place at the Centro SUVAG-RN[5], a rehabilitation center for hearing impaired people of all ages. We were assisted by rehabilitation professionals, who introduced the system to groups of deaf teenager students and adult teachers. The latter are experienced not only in fingerspelling, but also in teaching the LIBRAS sign language. The observations collected during the calibration stage are presented and discussed in Section 5.

[5] `http://suvagrn.com.br/`, last access on 12/02/2009.



(a)



(b)

Figure 6: The handshape corresponding to the 'X' letter, which has secondary animation. In (a) we have the primary handshape, whereas in (b) we have the final position of the hand, after completion of the secondary animation.

4. Methods

The evaluation of the *FAITH* system consisted of two phases: i) the experimental study A: testing the tool's teaching aspect; ii) the experimental study B: testing the tool's communication aspect; and iii) subjective analysis, based on direct observation of the volunteers during experiment, and user feedback.

4.1. Experimental Study A: Testing the Teaching Aspect

This experimental study was designed to test the teaching aspect of the *FAITH* system. We considered the *FAITH* system as the experimental condition, while the use of a video database was regarded as the control condition. The experiment was divided in two sessions, one for each type of experimental condition used to teach fingerspelling.

In both sessions participants had to learn a set of five signs of different complexities. We assigned letters with similar level of learning difficulty to two subsets. The signs assigned to each subset were chosen following two steps. First, we divided signs in three categories: *easy*, consisting only of handshapes with no secondary animation nor fingercrossing; *average*, signs having no secondary animation but containing fingercrossing; *hard*, with signs that include secondary animation. Each subset contained two easy, two

average and one hard sign.

Data Collected

Before the actual sessions, participants were asked to complete a general background information questionnaire. After the learning phase, students were asked to sign the studied letters to experienced signers, who assigned them correctness scores. We also measured the time that each participant spent when using either of the learning method.

Participants and Materials

A total of 10 participants (5 male and 5 female) took part in the experiment. Their mean age was 21.5 years (SD = 1.4). Participants were assigned to each condition, following between-subjects randomized design. We selected volunteers who did not have previous contact with any form of sign languages education.

Participants used two types of hardware configuration: i) a desktop system powered by an AMD Athlon XP1900+ (1.6GHz), with 1Gb of RAM and an onboard video card (64Mb); ii) a desktop system powered by an Intel Core 2 Duo 6700 (2.66GHz), with 2Gb of RAM and a GeForce 8600 GT. The reason for testing different hardware configurations was to detect any possible variation in performance.

The Procedure and Hypothesis

Prior to participating in the experimental session, volunteers received brief instructions on the *FAITH* user interface and had a 5-minute training/familiarization session.

Next, participants were divided in two groups. The group A first learned the subset #1 under the control condition (i.e. the video database); next they learned the subset #2 using the experimental condition (the *FAITH* system). The group B, on the other hand, first had to learn the subset #2 under the control condition; then they proceeded to learn the subset #1 under the experimental condition.

Participants in both groups had a 5-minute time-limit for learning each sign. The learning phase lasted about 25 minutes per student. Time limits were needed to prevent fatigue from affecting the results. After finishing the learning phase, participants were asked to demonstrated all the signs they had studied to a experienced signer. Next, they received a score corresponding to the number of correct signs executed. A one-way analysis of variance (ANOVA) was performed on these data.

The tested hypothesis was the following:

H_1: The *FAITH* system would be, at least, as efficient as the traditional method of teaching LIBRAS fingerspelling.

The rationale for this hypothesis is that the hand model employed is anatomically consistent with a real human hand, and the animation associated with signs have been accurately defined by experienced signers. Furthermore, we believe that the interactive aspect and the 3D model would, somehow, improve the learning process.

4.2. Experimental Study B: Testing the Communication Aspect

To evaluate the communication aspect of the proposed tool, a second experiment was held at the rehabilitation center for deaf people (Centro SUVAG-RN). This time we measured the degree of communication by demonstrating several signs via the *FAITH* system and asking volunteers to (a) directly identify the signs being made, and (b) indicate the sign being made from multiple choices of letters.

Data Collected

Before the actual sessions, participants were asked to complete a general background information questionnaire. During the experiment, participants were asked to fill in an answer sheet indicating the signs made by the system.

Participants and Materials

A group of 5 students (all aged 11) and 2 sign teachers (1 male, age 53, with hearing impairment; and 1 female, age 58, presenting normal hearing) took part in the experiment. All volunteers are regular students or staff members of the Centro SUVAG-RN.

For this experiment we used two kinds of hardware configurations: i) a desktop system powered by an AMD Athlon XP1900+ (1.6GHz), with 1Gb of RAM and an onboard video card (64Mb); ii) a notebook powered by an Intel Celeron (1.86GHz), with 512Mb of RAM and an Intel 945gm video card.

The Procedure and the Hypothesis

The first test was the direct letter identification. We defined a set of three letters, chosen according to their progressive difficulty level, as follows: i) letter 'I', which is easily distinguishable from any other alphabet sign; ii) letter 'R', which requires fingercrossing; iii) letter 'H', one of the hardest letters available, since it is only discernible from letters 'K' and 'P' by its secondary animation, and frame of reference (i.e. the signer's body). A random sequence of these letters were shown to the audience. Then, each participant had to indicate on the answer sheet the sign that they thought was being displayed.

The second test was the the multiple choice quiz. The quiz comprised 4 questions, each having 4 options to choose from. The options from each question were made of letters very similar to one another. For example, letters 'A', 'E' and

'S' are quite similar, varying only on how tightly closed the hand is shown, and where the thumb is placed. A total of 16 letters were displayed during this experiment.

The tested hypothesis was the following:

H_2: The communication aspect of the *FAITH* tool is precise and can be used to facilitate communication between people without prior knowledge of a sing language and others who understand fingerspelling.

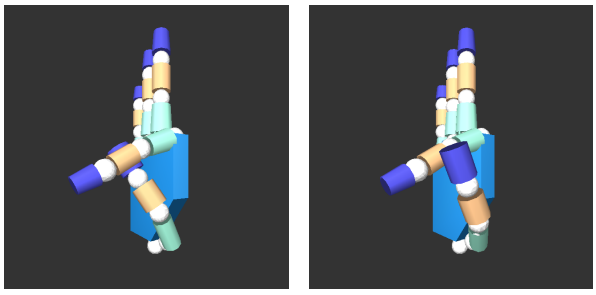
5. Results and Discussion

5.1. Analysis of the Calibration Phase

Observations collected during the calibration phase helped us improving handshapes precision, and, in many cases, adjust camera viewpoint. Calibrators pointed out a total of 6 mistakes in our system. This implies that the acquired key handshapes and the animation generated by the first version of the *FAITH* system were 80% precise.

After refining handshapes, we tested the system with a group of hearing impaired volunteers. That time, only 2 mistakes were noticed, leaving us with an overall 91% approval rate. Despite minor adjustments, the system was considered very precise and accurate.

We notice that rotation control were fundamental in distinguishing letters 'T' and 'F'. Handshapes for these letters are very similar, except for the fact that in the former the thumb is positioned after the indicator (i.e., with no fingercrossing), whereas in the latter it is placed before the indicator (with fingercrossing), as shown in Figure 7.



(a) **(b)**

Figure 7: Comparison of similar handshapes. (a) presents the 'T' sign, while (b) shows the sign corresponding to the 'F' letter.

5.2. The Teaching Experiment

The results of the ANOVA ($F_{1,14} = 5,74$, $p = 0.03$) indicate a significant difference on the type of learning method. Participants that used the *FAITH* system obtained a higher score than those who interacted with the video database.

Observations made during experiment revealed that the ability to stop animation any time and change viewpoint, were the key factors that enabled volunteers to better understand some of the complex signs used in the experiment. In some cases, like the letter 'X', the secondary animation present in the video dataset partially blocked the handshape; the same did not happen with *FAITH* due to the rotations.

The smooth transformation from one sign to the next was also an important feature, allowing users to learn how to sign in a continuous flow, rather than just doing a sing, stopping, and doing the next sign.

Finally, the participants who first interacted with the video database spent, on average, five minutes (the time limit per sign); whereas the participants who started off with the *FAITH* system spent, on average, two minutes. Consequently, the total time for those who used *FAITH*, then consulted the video database was reduced in roughly 70%, if compared to the time spent by those who first consulted de video database and then the *FAITH* system.

5.3. The Communication Experiment

The group of deaf volunteers who took part in this test was able to correctly guess the sign displayed by the *FAITH* system in 98% of the times, considering both the direct letter identification and the multiple choice questions.

The single mistake observed was made by the hearing rehabilitation teacher, who mistook the letter 'O' for the letter 'C'. She obtained an 85% score.

This results indicates that, as a communication tool, the primarily group of interest has found our system to be satisfactory.

6. Conclusion and Future Work

Traditional unsupervised teaching methods used to learn fingerspelling present some limitations. They often show the signs from a single viewpoint, and do not support a smooth transition from one handshape to the next.

In this paper we presented an alternative solution that tackles these issues: a desktop virtual reality system called *FAITH*. The proposed system provides an interactive fingerspelling mechanism aimed at assisting the learning of a sign language. *FAITH* offers natural transition between sequences of signs and lets users observe signs from different viewpoints, by means of rotations around coordinated axes.

Besides the *FAITH*'s primary objective, i.e. a teaching tool, the system may be used to facilitate communication among those who do not know a sign language and those who do comprehend.

We evaluated our system via two experiments. The first showed that the process of learning alphabet letters with the *FAITH* system is faster and more accurate than the traditional method of using a database of pre-recorded videos. The second experiment demonstrated that the tool was, approximately, 95% accurate when presenting signs to an experienced audience. Hence, it is possible to say that this type of communication tool is important in occasions where a certified translator is not available, for example in emergency situation at a hospital.

In terms of future work, we are currently pursuing two aspects: i) to extend the system so that we could use a full body avatar, thereby fully supporting the execution of the LIBRAS sign language; ii) to develop a web version of this system and, possibly, an adaptation that would function as a personal translator, running on personal digital assistants or even mobile phones.

Acknowledgment

The authors acknowledge financial support from Fundação de Apoio à Pesquisa do Estado do Rio Grande do Norte(FAPERN/CNPq/PPP, grant number 610031/2006-6), and CCET/UFRN.

References

[1] G. Burdea. *Haptic Rendering: Algorithms and Applications*, chapter The Role of Haptics in Physical rehabilitation. AK Peters, Wellesley, MA, USA, first edition, 2008.

[2] G. Fiedler. Gaffer on games: Fix your timestep!, `http://gafferongames.com/game-physics/fix-your-timestep/`, last access, 19/February 2009.

[3] N. Frishberg, S. Corazza, L. Day, S. Wilcox, and R. Schulmeister. Sign language interfaces. In *Conference on Human Factors in Computing Systems*, pages 194–197, 1993.

[4] A. Lécuyer, P. Mobuchon, C. Mégard, J. Perret, C. Andriot, and J. Colinot. HOMERE: a multimodal system for visually impaired people to explore virtual environments. In *Proceedings of the IEEE Virtual Reality 2003 (VR'03)*, pages 1–9, 2003.

[5] N. Lowe, J. Strauss, S. Yeates, and E.-J. Holden. Auslan jam: A graphical sign language display system. In *DICTA 2002 Conference Proceedings of Digital Image Computing Techniques and Applications*, pages 1–6, 2002.

[6] S. M. Malfatti, S. R. dos Santos, L. M. Fraga, J. C. Oliveira, and C. Justel. The design of a graphics engine for the development of virtual reality applications. *Revista de Informática Teórica e Aplicada*, XV(3):26–45, December 2008.

[7] G. Riva, C. Botella, P. Lgeron, and G. Optale, editors. *Cybertherapy: Internet and Virtual Reality as Assessment and Rehabilitation Tools for Clinical Psychology and Neuroscience*. Ios Press, Amsterdam, first edition, 2004.

[8] A. Rizzo and G. J. Kim. A SWOT analysis of the field of virtual reality rehabilitation and therapy. *MIT Journal of Presence: Teleoperators and Virtual Environments*, 14(2):119–146, November 2005.

[9] F. D. Rose, B. M. Brooks, and A. A. Rizzo. Virtual reality in brain damage rehabilitation: Review. *CyberPsychology & Behavior*, 8(3):241–262, June 2005.

[10] E. Sedgwick, K. Alkoby, M. J. Davidson, R. Carter, J. Christopher, B. Craft, J. Furst, D. Hinkle, B. K. a. Lancaster, S. Luecking, A. Morris, J. McDonald, N. Tomuro, J. Toro, and R. Wolfe. Toward the effective animation of american sign language. In V. Skala, editor, *WSCG 2001 Conference Proceedings*, 2001.

[11] D. Shreiner, M. Woo, J. Neider, T. Davis, and A.R.B. OpenGL. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R), Version 2*. Addison-Wesley Professional, fifth edition, August 2005.

[12] A. Watt. *Advanced Animation and Rendering Techniques*. Addison Wesley (ACM Press), New York, NY, USA, 1992.

[13] R. Wolfe, N. Alba, S. Billups, M. J. Davidson, C. Dwyer, D. G. Jamrozik, L. Smallwood, K. Alkoby, L. Carhart, D. Hinkle, A. Hitt, B. Kirchman, G. Lancaster, J. McDonald, L. Semler, J. Schnepp, B. Shiver, A. Suh, and J. Young. An improved tool for fingerspelling recognition. In *Technology and Persons with Dsabilities Conference 2006*, pages 20–25, 2006.

[14] S. Yeates, E.-J. Holden, and R. Owens. Real-time 3d graphics for human modelling and teaching sign language. In Zakopane, editor, *ICCVG 2002 Conference Proceedings*, 2002.

[15] E. Zimand, P. Anderson, G. Gershon, K. Graap, L. Hodges, and B. Rothbaum. Virtual reality therapy: Innovation treatment for anxiety disorders. *Primary Psychiatry*, 9(7):51–54, November 2003.