Fully-Automated English Text to Arabic Sign Language Translator

Asmaa Hosni

Assiut University, Electrical Engineering Department, Assiut, Egypt

Abstract

The sign language is considered very important because it is the native language of many deaf people in the world. In this paper, we proposed and implemented a fully-automated and functional translation system that aims to translate from input English Text to Arabic Sign Language (ArSL). Our system is fully-automated, meaning that the system does not need a human operator to apply the signs. In this system, the input English sentence is first translated into a simple Arabic sentence to preserve the grammars of the Arabic language. Then the translator generates a sentence that is fit for the language of Arabic deaf people (ArSL is a language in its own). Then, this deaf Arabic sentence is the sentence to be animated later on the display. Finally, the proposed system uses a combination of hand shapes, positions and movements as well as simple facial expressions in graphically animating the ArSL signs smoothly on the display. This system is mainly implemented to be an education tool for Arabic deaf children. However, it can provide an adequate translation for certain situations like conferences and public places like, restaurants, railway stations and airports, as well as a tool for learning signs for non-signers. Also, it can help deaf Arabic people with less English capability to browse the internet. It could also be useful in emergency situations such as fires and calls in hospitals or police stations, where urgent information could be conveyed without having to wait for a human translator.

Keywords:
Sign language, Deaf, Automatic translator, Computer graphics.

1. Introduction

No doughty, the sign language provides the foundation for self-esteem, educational achievement and social well-being within the deaf community. The sign language makes them take benefits of vast access to information, services and education. The deaf community exists within a wider society of hearing people, many of whom are unaware of sign language and deaf culture. Because of this, deaf people who use sign language experience high levels of social exclusion. Technical facilities that support communication in sign language could be a good chance to social integration of deaf people.

Manuscript received December 5, 2024
Manuscript revised December 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.12.17

The sign language could be considered the first language for deaf people. It is the initial means of communication for them to build their knowledge and education.

On the other hand, the sign language should be considered as any other natural language. It has its own vocabulary, grammar and a large group of speakers. Some people have described sign languages as “gestural” languages, but this is not absolutely correct [1]. Hand gestures are only one component of the sign language; Facial expressions such as eyebrows and mouth-lip movements are also significant components of the sign language. In addition, the sign language makes use of the space surrounding the signer to describe places and persons that are not present [1, 2].

The sign language is not universal. Meaning that, the American Sign Language (ASL) is totally different from the Arabic Sign Language (ArSL). Besides, every sign language has its own grammars and vocabulary. Furthermore, ArSL has different grammars than Spoken Arabic Language (SAL), which makes it a language in its own.

In this paper we present a fully-automated translator system that translates from English text to ArSL without using human operator. We called this system AutoETAST. This system needs first to translate the input English text to Arabic equivalent. Then convert the Arabic Text to its ArSL signs equivalent and finally animates these signs using a graphical representation.

The AutoETAST full system is presented in Section 3. Related work is described in section 2. The evaluation of our proposed system is discussed in Section 4. Finally, conclusions and future directions are summarized in Section 5.

2. Related Work

While the last years have seen an ever increasing development of machine translation systems from major spoken natural languages to sign languages, translation from English to ArSL is virtually ignored. In fact, translation systems can be divided into two categories according to the input: Machine translation from Spoken languages to Sign Language, and Machine translation from text to Sign Language.

Several projects have been done taking as input a spoken language such as English, French, Arabic, British, etc. In [3], a Speech-To-British Sign Language (BSL) translation system that aims to provide a communication aid between a deaf person and a Post Office clerk is presented. The system hired what is called a formulaic grammar approach in which, a set of pre-stored phrases are used and the translation is done via a phrase lookup table. However, using of a predefined set of phrases in the lookup table makes the system a very domain specific. It assumes a very restricted conversation between the users.

In the same context, the framework presented in [4] utilized the power of cloud computing for the complex processing of Arabic speech to Arabic text. The speech processing produces a cartoon avatar showing the corresponding Egyptian Arabic Sign Language on the mobile handset of the deaf person.

Other machine translation systems translate from input text to Sign Language. In the context of translating from non-Arabic input text, the work proposed in [5] translates from English text to American Sign Language (ASL). The authors used Synchronous Tree Adjoining Grammar formalism to represent input text into the ASL syntactic structure. The system holds a bilingual lexicon to identify each valid word-sign pair. The output of the linguistic part was a written ASL gloss notation. The pose and hands information along with the morphological variations, are embedded within the ASL gloss notation. The output of the displaying part uses animated human model (Avatar).

On the same area, authors in [6] implemented a system for translating from English text to British Sign Language (BSL). In their approach, they used the CMU Link Parser to parse the input English text. Then, they used Prolog declarative clause grammar rules to convert this linkage parsed output into a Discourse Representation Structure. During the processing of input English, Head-Driven Phrase Structure rules are utilized to generate a symbolic BSL representation. This representation is then used to display movements required to perform the output Sign Language.

In the context of the translation systems that translate from Arabic text to ArSL, we mention [7, 8]. In [7], the translation strategy is based on the translation rules of the ArSL and the domain ontology to generate the notation called "Sign-Writing" (it is an annotation system for the Sign Language). The "Sign-Writing" is then used as the final output of the system or as an intermediate level for a future animation of an avatar.

In an intermediate step, authors in [8] developed a translation system that has a database for storing Arabic words with their corresponding signs. Each sign is recorded by a sign language human operator in a video sequence. These recorded sign language files representing words in the Arabic dictionary are stored manually on the server in a specific location. Then, when the server receives a request for a particular sign, it searches for the video in that location. The symbols corresponds to each sign are coded in the dictionary via the name of the corresponding file in the video sequence.

In the same context, authors in [9] also presented a system for translating Arabic text to ArSL. It first used knowledge base to solve a number of Arabic language difficulties like synonyms, inflectional, derivational, diacritical, and plural. Besides, the system involves finger spelling translation to overcome the problem of Arabic input words that doesn't exist in the database.

Also, the system presented in [10] is another Arabic text to ArSL translation system which is called "Tawasol". It is used as an educational tool. It has a translator, a dictionary for Arabic words in a set of categories, besides a finger-spelling editor. The dictionary includes different themes which contains a substantial number of words. It offers for the user the ability to find a word, to repeat the word and to control the sound of the word. However, the user have to match the word with the correct video of the sign. In the finger-spelling choice, the system takes the word entered by the user and displays the sign language symbol for each letter.

Recently, authors in [11] implemented a system in which input is a text in Arabic and the output is a real-time, online representation in ArSL Arabic Gloss. They started by a linguistic treatment for the Arabic input sentence (using the MADAMIRA Analyzer system), passing through the definition and the generation of Arabic Annotation Gloss system and coming finally to producing an animated sentence using Arabic Gloss.

Also most recently, a machine translation system (ATLASLang MTS 1) that is based on rule-based Interlingua and example-based approaches is presented in [12]. In this work, a SAFAR Platform and ALKHALIL morpho system are hired to extract the morphological properties of each word of the input Arabic phrase. Then it generates a video sequence representing the sentence in ArSL based on well-established translation rules and a database of signs. They used a sample of a database of gif images. During the translation, if an input word does not have a matching correspondence in the database, it will be displayed using finger spelling. The main shortcoming of this system is that it assumes the input sentence must be completely vowelized.

3. Proposed AutoETAST System

To our knowledge, the system proposed here is the first machine translation system to translate from English input text to graphically animated ArSL.

There are some systems that simply render spoken English as text [13], or as icons [14] or Arabic Gloss such as done in [11]. Such systems are inadequate for Arabic deaf people for two reasons:

1) Many deaf people in the Arabic countries have difficulties with reading and writing English. A text-based systems would make it impossible for these people to follow and understand conversations.

2) If spoken languages is rendered as text or icons, all the information on intonation, pitch and timing is lost, even though this information is very important. In contrast, ArSL is capable of conveying this information by stressing the motion of signing and facial expressions.

As a result, a fully-functional English text/ArSL machine translation system would be superior to a text-based or iconic systems when it comes to conveying all the nuances and tones of spoken language. Besides, a better approach would be using three dimensional (3D) animations of ArSL signs being performed, as suggested in this work. Consequently, the animation approach will make signing more accurate and easier to understand and cancels the presence of human signer.

In the present work a fully Automated English Text to Arabic Sign Language Translator (AutoETAST) is introduced. To our knowledge, this is the first attempt to construct a machine translation system from English to 3D animated ArSL. The system takes into consideration not only linguistics of Arabic language but also visual and spatial information associated with ArSL. The result of using this system demonstrates that, such translator is useful for integrating the deaf people in a productive community.

Our approach in building the proposed system consists of five stages:

1) English Parser: In this stage, the English input sentence is parsed using CMU link grammar rules. The output will be English words and links.

2) Vocabulary Mapping: In this stage, a naïve translation from English words to Arabic ones will be done.

3) Arabic-like sentence construction: The output of this stage will be an Arabic-like grammatically correct sentence.

4) ArSL sentence construction: The purpose of this stage is to produce an ArSL grammatically correct sentence (sequence of signs to be animated), according to the grammar of ArSL.

5) Signs Animator & Emotion Generation: This stage is an interpretation of the ArSL sentence as a motion representation with a set of parameters which actually control the graphical human model to produce the ArSL signs.

Figure (1) shows the overall structure of the proposed system. We will go through details of each part in the next sections.

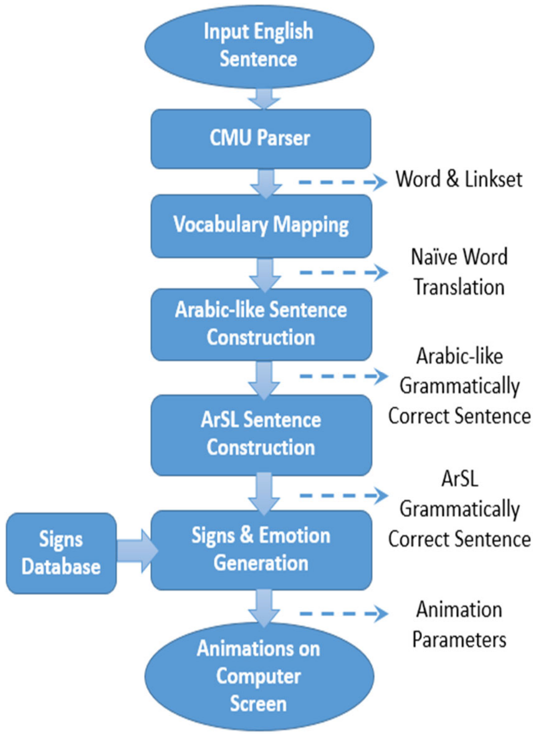

Figure (1): Overall structure of the proposed system

3.1 CMU Parser

There are many methods that can be used in parsing the English input sentence, such as Dependency Grammar [15], Lexicallized Tree Adjoining Grammar [16], Bilexical Grammar [17], and CMU Link Grammar [18]. CMU Link Grammar is used in the present work. We choose the Link Grammar parser because of its simplicity, it proves its robustness over time, its output satisfies our requirements and it is open source.

The Link Grammar is a syntactic structure, which consists of a set of labelled links connecting pairs of words in the sentence. The basic idea is that, think of words as blocks with connectors coming out (or in) which is called links. These links describe the relations between words and should satisfy the following conditions:

i) The links do not cross when drawn over the words.

ii) The links are enough to connect all the words of the sequence together.

iii) The links satisfy the linking requirements of each word in the sequence.

Each link is a positional relation for each sentence components like adjectives, verbs, nouns, etc. As an example of such relations, it is known that, the adjective should be followed by a noun as in the sentence "The BIG DOG chased me". In our system, each link is denoted by English letters (link name) followed by a + or – sign to show the direction of the connection.

A few examples of the used links are:

- The S link is used to connect a noun to its verb.
- The O link is used to connect a verb to its object.
- The A link is used to connect an adjective to its noun.
- The D link is used to connect a determiner to its noun.

Figures (2), (3), (4) and (5) are some examples of English sentences parsed using the previously described links.

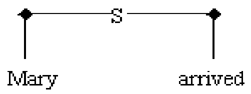

Figure (2): Example of the S link

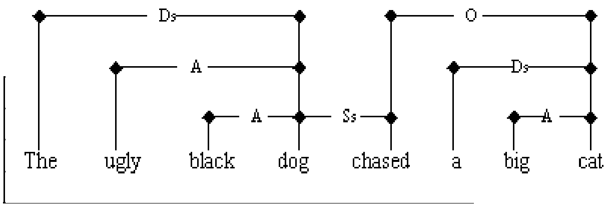

Figure (3): Example of the D link

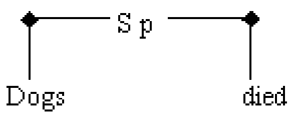

Figure (4): Example of the S link

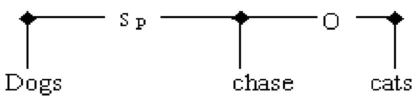

Figure (5): Example of the O link

All possible links of each word used in the translation system are contained in a table. This table consists of a collection of entries, each of which defines the linking requirements of one or more words. To illustrate the utilization of these tables, consider the simple example of parsing the sentence "The cat chased a snake" as shown in Figure (6).

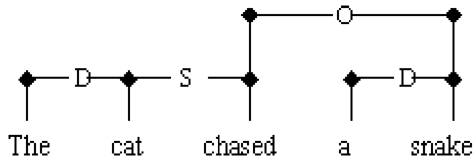

Figure (6): Parsing of the sentence "*The cat chased a snake*"

Table (1) describes the linking requirements of the words used in this example. The linking requirements for each word is expressed as a formula involving the operators &, and ^, parentheses, connector names and the suffix + or -. The + or - suffix after a connector name indicates the direction (relative to the word being defined) in which the matching connector should lie. The sign + means that the link is directed outward from this word and - means that the link is directed inward for this word. The well-known logical & operator of two formulas is satisfied by satisfying both formulas. The ^ of two formulas requires one and exactly one of its formulas to be satisfied (equivalent to the logical XOR).

Table (1) Linking requirements

Words	Formula
a the	D+
snake cat	D- & (O- ^ S+)
chased	S- & O+

The result of this process is the sentence expressed completely as words and links' names. Consequently, the links of the sentences can be viewed as a way of specifying the constituent structure of the sentence.

During the parsing operation, the parser extracts a grammatical information such as sentence type (single or plural) and morphological marks such as tense (past, present and future). Such information could be extracted by analyzing the last few letters of each word plus using look up tables.

2.1 Vocabulary Mapping

After parsing the input English sentence, our translator replaces each English word with its Arabic equivalent. Two tables are used in the translation process: English/Category and Category/Arabic tables. Tables (2 a) and (2 b) show an example of such tables.

Table (2): (a) English/Category

English/Category	
Dog	DOG
Member	MEMBER
Milk	MILK
Mother	MOTHER
Thanks	THANKS
I me	I
He him his	HE
She her	SHE
They them their	THEY
Am are is be being	AM
Chase chases chased chasing	CHASE

The English/Category table consists of lists of the English words and their corresponding categories. Some entries of the table are one to one entries, meaning that, one English word has its own category. For example, the word dog has the category DOG and the word mother has the category MOTHER, see Table (2 a). Other entries in the table are many to one entries, meaning that more than one English words belongs to the same category. For example, the words he, him, his have the category HE, see Table (2 a). All forms of a given verb forms a single category (present, past, future). For example, all forms of the verb chase (chase, chases, chasing, chased) have the same category CHASE. The second table is the Category/Arabic table which has the opposite structure of the English/Category table. It relates each category to an Arabic word (see Table (2 b)).

Table (2): (b) Category/Arabic

Category/Arabic	
DOG	كلب
MEMBER	عضو
MILK	لبن
MOTHER	أم
THANKS	شكرا
I	أنا
HE	هو
SHE	هي
THEY	هم
AM	يكون
CHASE	يطارد طارد مطاردة

To get the category of an English word, the translator looks it up in the English/Category table. Using this category, the translator searches for the corresponding word in the Category/Arabic table. The problem is that, a single category in the Category/Arabic table could corresponds to more than one choice of the Arabic words (one-to-many relation). For example, the category CHASE in the Category/Arabic table corresponds to the Arabic words يطارد، طارد، مطاردة (see Table (2 b)). To start with, the translator just chooses the first word that it finds; this is called 'dumb word replacement'. Unfortunately, the translator may not initially choose the right forms for all of the words that it suites the construction of the sentence. For example, adjectives may be in the wrong case or gender, verbs may be in the wrong form, and so on. Thus to correct the choice, extra information from the CMU link grammar parser should be used.

In Link Grammar, sentence components like noun-verb agreements are specified by subscripts on connectors, singular nouns and verbs have an Ss connector, plural forms have an SP. This subscript information is what our translator uses to choose the correct word from "he" category. This is called 'smart word replacement'. However, the proposed translator, do not need this 'smart word replacement'. The aim of the proposed translator is to translate from English to ArSL, not to SAL. Fortunately, the ArSL is category based, and this is why ArSL does not need the 'smart word replacement'. The 'dumb word replacement' is good enough and this could be considered another point stressing the fact that ArSL is not as complex as SAL as shown in Table (3). The output of this stage is a naïve translation from input English words to Arabic ones. It is missing the right order of the words, which depends on the ArSL grammars.

3.3 Arabic-Like Sentence Construction

While some simple Arabic-like sentences can be reconstructed by simple word-to-word replacement, this is usually not sufficient. In many cases, words need to be added or deleted, or word order must be changed. As an example, for the Arabic sentence to be in the right form, the translator should swap the adjectives if found in the parsed English sentence. For example, if the input English sentence is "I am very happy", after translation without swapping adjectives the Arabic-like sentence will be "أنا يكون جدا سعيد", and this is illegal in Arabic grammars but, after swapping adjectives the Arabic like sentence will be "أنا يكون سعيد جدا". Another example, if we have the sentence "The dog chased the small ugly black cat", which have more than one adjective, the output of the translator after swapping adjectives will be " ال كلب يطارد ال قطة ال سوداء ال قبيحة ال صغيرة ".

As shown in Figure (6), the result of the naïve translation is a sentence with wrong construction that is not accepted in ArSL. The swapping process is done by the help of a set of rules. Each rule catches a specific English construction and provides the corresponding correct Arabic sentence construction.

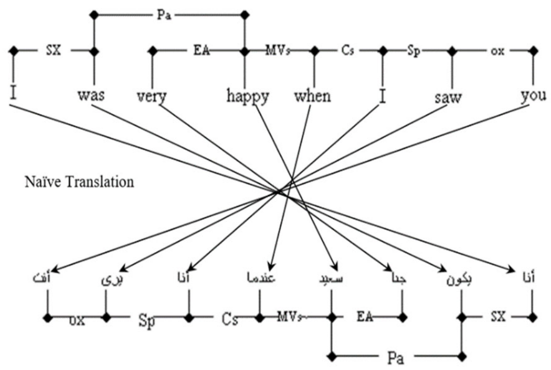

Figure (6): Example of the vocabulary mapping process

3.4 ArSL Sentence Construction

As mentioned previously, ArSL has some differences than SAL (see section 3.3). After producing the Arabic-like sentence, the translator generates the ArSL sentence (sequence of signs to be animated). This is done by applying the ArSL rules on the output of the previous stage, such as removing the Arabic determiner (ال) and adding extra signs indicating the tense and gender. Furthermore, the emotion information are extracted in this stage. Some words are connected to emotions which will be used in the animation stage. Table (4) shows some examples of ArSL sentence constructions. Note that, the determiner ‹ال› is not present in the ArSL. It is used in the examples to clarify any confusion for the Arabic readers. To demonstrate the previous 4 steps, consider

Table (4): Examples of ArSL sentence constructions

Example	Emotion	Description
I will dance in the room. أنا/ (إشارة المستقبل)/ يرقص في الحجرة	Smiling	Future verbs is represented by inserting the sign of the future tense before the verb.
Yesterday, I was sad. بالأمس أنا / (إشارة الماضي) / حزين	Depressed	Past verbs is represented by inserting the sign of the past tense before the verb.
He is a doctor. هو طبيب	Normal	Predicative constructions which have nouns and adjectives as descriptors are mapped to ArSL sentences without verbs.
The boy is not buying a dog. الولد / (إشارة النفي) / يشتري كلب	Normal	Negation in ArSL is represented by inserting the sign of negation in the ArSL sequence.
Did you leave the dog in the house? هل أنت / (إشارة الماضي) / يترك الكلب في المنزل	Normal	The yes/no question is delivered with the sign, ‹هل before the sentence.
The science books are very important. / (إشارة الجمع) / كتاب العلوم مهم	Normal	Plurals are represented by inserting the sign of plurals in the ArSL sequence.
If it rains tomorrow, I will not go to school. لو أنها / (إشارة المستقبل) / تمطر غدا, أنا / (إشارة المستقبل) / (إشارة النفي) / يذهب ألي المدرسة	Normal	More than one sign can be inserted if required, in the ArSL sequence.

the example shown in Figure (7). The example shows the parsed English sentence in the upper line with its links indicated on it. The second line shows the results after the naïve translation. It is clear that, the output of this stage has an illegal Arabic construction. A more appropriate Arabic sentence generated from the ArSL sentence construction stage. The final line shows the sequence of signs (represented by Arabic words) to be animated in the following stage.

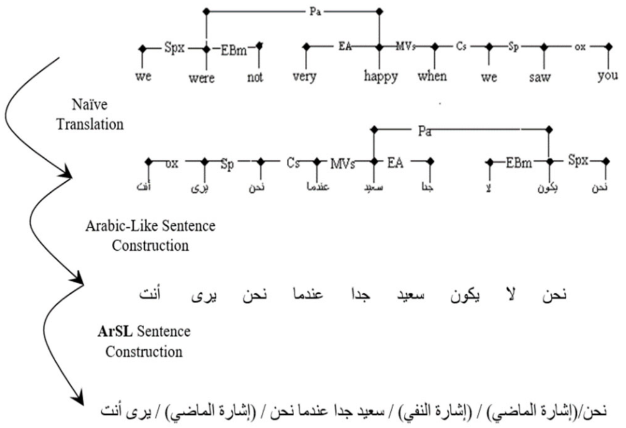

Figure (7): Summary of the first 4 steps in the translation

3.5. Signs and Emotions Generation

It is sufficient for animating ArSL signs on the screen to have an articulated model for the upper body of human being. The model should have fully articulated hands (with all five fingers), arms as well as controllable simple facial expressions. In addition a fast mathematical mechanism is needed to procedurally generate the animation for each ArSL sign.

There are many methods that can be used to achieve the required animation. One of these techniques is to store each sign representation as a set of frames or images to be replayed when needed as movies [10, 12]. A second technique uses storedparametric data (set of angles and orientations) allowing run-time generation of each sign movement through interpolation. These parametric data is used to control the 3D model [19, 20]. Unlike the first method, the second method can generate a smooth interpolation from one sign to another, providing more realistic output.

In the present work, the animator uses the parametric computer-generated sign animation technique. This animator system consists of three parts: The first part is a 3D human graphical model of the whole upper body. The second part is the mathematical animation mechanism to control the graphical model. The third part is the graphical user interface where English sentence to be translated is entered.

3.5.1. 3D Modelling

The biomechanical structure of the human upper body can be well simulated by a kinematic tree as shown in Figure (8) [21]. The kinematic tree consists of nodes, each with a parent and an arbitrary number of children. A node represents a joint in the body, containing a relative angles from the parent node's local axes to its own. The motion is restricted to rotation around the joints. These rotation angles is controlled by the animation system. Each tree-node has an associated polygonal model representing the surface of the body segment directly following the joint. Representing a body part is performed by transforming the associated polygonal model to the node's local coordinate space, then rendering the polygonal in this position. All polygonal model (segments) after a node in the kinematic tree can be rendered by performing a depth-first traversal of the tree from that node.

The proposed upper body model (shown in Figure (9)) has 39 joints with a total of 117 degrees of freedom (DOF). Some of these DOFs is not active (is not used by any of our signs). The kinematic tree used in rendering graphical model is implemented using C++. The choice of C++ is because it

Figure (9): Kinematic tree of the graphical model

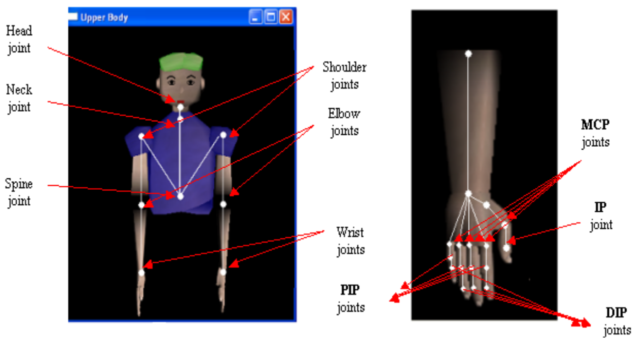

Figure (10)

a) Joints of the upper body b) Joints of the wrist realistically looking images and animations. This technique offers best image quality, but it is too slow. For example, in playing a game at 3,440 x 1,440 resolution with the 'Extreme' preset, around 41 fps can be achieved. However, with a Nvidia GeForce RTX 2080 Ti (GPU card), once ray tracing is enabled with the Ultra setting, the performance collapses to around 23 fps [22]. It needs computation power capabilities that may not exists in the usual PC.

The second rendering method is by using simple solid object rendering with minimum consideration to lighting and reflectivity of surfaces. The tool that used in our system is the popular OpenGL [23, 24]. OpenGL is a programming interface that gives an access to advanced graphical functions offered by the hardware accelerated graphics cards. Most of the contemporary graphics cards are capable of supporting hardware-accelerated graphics. Considering an 800 by 600 image in true color and 25 frames per second, the graphics card should continuously generate 36 megabytes of data per second. This and higher rates are supported by relatively cheap graphics card using OpenGL.

The 3D graphical model is constructed to maintain the proportional dimensions of average human being to be realistic. There are three joints to represent the spine. One for the base of the spine, one for the base of the neck, and one for the base of the head as shown in Figure (10 a). Also in the arm, the shoulder, the elbow, and the wrist are represented by three joints. Furthermore, there are 15 joints per hand as shown in Figure (10 b). The fingers are modeled using joints representing the metacarpophalangeal (MCP), the proximal interphalangeal (PIP), and distal interphalangeal (DIP) joints. The thumb is similarly modeled with joints representing the carpometacarpal (CM), (MCP), and interphalangeal (IP) joints. Thus, the kinematics of our hand closely mimics a real hand.

Each joint allows full three directional motions. Some of the joints are restricted to only one (The PIP and DIP joints) or two (The wrist joint) of these directions. Other joints like the elbow joint, allows the full directional motions. These Physiological constraints on movements are imposed on the animation system. Examples of these constraints, are shown in Table (5).

3.5.2 The Sentence Animator

The sentence animator is the part of the system which is responsible of animating a complete ArSL sentence. The sentence animator takes each word in the ArSL sentence and search for the right parameters in the sign animation parameters table (ArSL/signs table).

The animator should generates signs with a smooth transitions between them. A simple and straightforward approach (not used in the proposed system) for the smooth transitions is to unify the beginning and ending of every sign. The unified frame used in the beginning and ending is called 'neutral position'. While this approach offers smooth continuous transitions, the resulting animations is very unnatural. A more computationally expensive approach is to define transitions between every pair of possible motions. In [25] a technique called motion blending is proved to be robust approach to automatically generate smooth transitions between isolated motions. This approach succeeds in avoiding returning to a required 'neutral position', and succeeds in most cases in generating natural transitions. The last method is the one utilized by the proposed system.

As we mentioned in section 3.4, an emotion tag is attached to each word in the ArSL/signs table. In this stage, rendering this emotions as a facial expressions is performed. The proposed system supports four kinds of facial emotions (normal, smiling, depressed and surprised), see Figure (11).

Figure (11): The 4 supported emotions

a) Normal b) Smiling c) Depressed d) Surprised

The sentence animator is the part of the system which is responsible of animating a complete ArSL sentence. The sentence animator takes each word in the ArSL sentence and search for the right parameters in the sign animation parameters table (ArSL/signs table).

The animator should generates signs with a smooth transitions between them. A simple and straightforward approach (not used in the proposed system) for the smooth transitions is to unify the beginning and ending of every sign. The unified frame used in the beginning and ending is called 'neutral position'. While this approach offers smooth continuous transitions, the resulting animations is very unnatural. A more computationally expensive approach is to define transitions between every pair of possible motions. In [25] a technique called motion blending is proved to be robust approach to automatically generate smooth transitions between isolated motions. This approach succeeds in avoiding returning to a required 'neutral position', and succeeds in most cases in generating natural transitions. The last method is the one utilized by the proposed system.

As we mentioned in section 3.4, an emotion tag is attached to each word in the ArSL/signs table. In this stage, rendering this emotions as a facial expressions is performed.

The proposed system supports four kinds of facial emotions (normal, smiling, depressed and surprised), see Figure (11).

The model would be first in its normal face then, according to the emotion information attached to each word, the system could select the suitable emotion.

3.5.3. The Signs Animator

To animate a sign, the system generates 3 to 5 control points (key-frames) and interpolates between these points to generate the smooth animations. In order to achieve this, the Euler orientation representation method is used [26].

Euler representation describes the orientation of an object using rotations around three axes. The axes of rotation are fixed to the object as opposite to being global axes. This method is sufficient to represent the object in any position. The Euler vector of the three angles at any instant of time is denoted by $q(t)$, where $0 \leq t \leq 1$. This vector could be represented in the in-between key-frames as a linear interpolation of the initial state q_i, and the final state q_f as follows:

$$q(t) = q_i * (1 - t) + q_f * t \qquad (1)$$

The set of parameters needed for the key-frames can be deduced using two methods:

1) In the first method, a set of control points is generated by a human operator. The in-between frames is then automatically generated using a method of dynamics or kinematics [27]. Then tested as a computer graphics animations and refinements and tuning is performed if needed, until a reasonably good animations is obtained.
2) The second method uses a motion capture device such as hardware sensors, or a vision based system where the joint angles are traced automatically [28].

The first method, is used in the present system for simplicity, and a forward kinematics [26] is applied to generate the in-between frames.

3.5.4. Graphical User Interface

The proposed system provides an interactive environment not only to display the Signs of the ArSL, but also it provides an easy way to enter the input English text as shown in Figure (12 a). The user can enter any English text as shown in the figure, then press the ENTER key to see the animations of the corresponding ArSL. If there is no translation for an English word or where the English sentence is too complex to be parsed, the system displays an error message as shown in Figure (12 b).

4. System Evaluation

It is usually hard to find a quantitative evaluation for natural language translations and usually the evaluation is left for human judgments. Furthermore, the evaluation of the quality of the graphical animation usually suffers from the same problem. So, to evaluate the proposed system we tested it with a real deaf children.

Our test consisted of two structured questionnaires:

1) One for testing the ability of the system to translate single English words to its right corresponding Arabic Signs. This questionnaire included 30 English animated signs. The student is asked to provide the corresponding Arabic meaning for each sign These 30 words are chosen arbitrarily from the translator data base.
2) The other questionnaire is for testing the ability of the system to translate a complete English sentence to its correct corresponding meaning in ArSL. This questionnaire included 10 English sentences. We asked their teachers to aid us by letting each student repeats the sentence which he/she understood using ArSL, then the teachers evaluate their understanding on a scale from 1 to 10.

Figure (12) (a) The GUI of the proposed system
b) The error message

The size of the evaluation set is 50 deaf students. The first group consists of 15 students, nearly from age 14 to 18 years old, primary school and the second group represents the rest of the students nearly from age 19 to 22 years old, secondary school. A summary of our results are shown in Table (6) and Table (7).

From these tables we concluded that, the misunderstanding rate was high for students from the 1st group (14-18 old). The main reason is that, the sign language ability of some of these students are below average and their teachers in most cases commented that, this misunderstanding is due to the student's ability but the animated sign on the screen looks fine. An overall performance evaluation criteria of the system could be summarized in one number. This number is the percentage of correctly recognized signs. It was found to be 90%.

Table (6): The results of testing our system using the first questionnaire

No. Unknown Words	No. Students	14-18 Old	19-22 Old	% of 1st Group	% of 2nd Group	% of Total Students
2	27	17	10	63 %	37 %	54 %
3	15	10	5	67 %	33 %	30 %
4	8	5	3	62 %	38 %	16 %

Table (7): The results of testing our system using the second questionnaire

Evaluation of 10 Points	No. Students	14-18 Old	19-22 Old	% of 1st Group	% of 2nd Group	% of Total Students
8	8	3	5	38 %	62 %	16 %
9	14	5	9	36 %	64 %	28 %
10	28	9	19	32 %	68 %	56

5. Conclusion and Future Work

This paper provides a system to translate from English text to ArSL, taking into account not only linguistics but also visual and spatial information associated with ArSL signs.

The system consisted of five stages. The English sentence in the first stage is parsed using the method of CMU Link Grammar. Tables are used in the second stage to replace each English word with the corresponding Arabic ones, producing a naïve Arabic translation. The third and fourth stages are implemented by applying simple Arabic rules and ArSL rules, respectively to the output of the second stage, producing an ArSL sentence that is grammatically correct.

A signs animator is employed for the fifth stage. It assumes that the ArSL sentence is already in correct linguistic order. For each word in the ArSL sentence, the sign animator uses that word as an index to look up a signs database, which stores the parameterized motion templates for all available ArSL signs.

The evaluation of this system with real deaf students shows that the system achieved what it is implemented for.

In a future work, we will make enhancements for the GUI of the system, generate realistic-look graphical representation for displaying the signs, try to generate a mobile application using same strategy, and increase the signs control points database.

References

[1] A. Braffort, "Research on Computer Science and Sign Language: Ethical Aspects", in: Wachsmuth I., Sowa T. (eds) Gesture and Sign Language in Human-Computer Interaction. GW 2001. Lecture Notes in Computer Science, vol. 2298. Springer, Berlin, Heidelberg.

[2] Deaf Web Site: http://www.deaflibrary.org.

[3] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. bbott, "TESSA, a system to aid communication with deaf people", Proceedings of the ACM Conference on Assistive Technologies, ASSETS 2002.

[4] M. M. El-Gayyar, A. S. Ibrahim and M. E. Wahed, "Translation from Arabic speech to Arabic Sign Language based on cloud computing", Egyptian Informatics Journal Vol. 17, Issue 3, 2016, pages 295-303.

[5] L. Zhao, K. Kipper, W. Schuler, C. Vogler, M. Palmer and N.I. Badler, "A Machine Translation System from English to American Sign Language", 2000, pages 54–67.

[6] J. a Bangham, S.J. Cox, R. Elliott, J.R.W. Glauert, I. Marshall, S. Rankov and M. Wells, "Virtual Signing: Capture, Animation, Storage and Transmission", – an Overview of the ViSiCAST Project Face-to-Face Transactions, Gesture, 2000, pages 1–7.

[7] A. Almohimeed, M. Wald and R. I. Damper, "Arabic text to Arabic sign language translation system for the deaf and hearing-impaired community", SLPAT: Speech and Language Processing for Assistive Technologies, Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies, 2011 Pages 101–109.

[8] M. Mohandes and S. Arabia, "Automatic Translation of Arabic Text to Arabic Sign Language", ICGST Int. J. Artif. Intell., 2006, pages 15–19.

[9] A. E. E. El Alfi, M. M. R. El Basuony and S. M. El Atawy, "Intelligent Arabic text to Arabic Sign Language Translation for Easy Deaf Communication", International Journal of Computer Applications, Vol. 92 – No. 8, 2014.

[10] A. Al-Nafjan1, B. Al-Arifi and A. Al-Wabil, "Design and Development of an Educational Arabic Sign Language Mobile Application: Collective Impact with Tawasol".

[11] N. Aouiti and M. Jemni, "Translation System from Arabic Text to Arabic Sign Language", Journal of Applied Intelligent System, Vol. 3 - No. 2, 2018, pages 57 – 70.

[12] M. Broura, and A. Benabboub, "ATLASLang MTS 1: Arabic Text Language into Arabic Sign Language Machine Translation System", Procedia Computer Science 148, 2019, pages 236–245.

[13] F. Vargas, R.D. Fagundes and D. Barros Jr, "A New Approach to Design Reliable Real-time Speech Recognition System", 7th International On-line Testing Workshop, 2001.

[14] Sign Writing Site: http://www.signwriting.org.

[15] N. Fraser, "Parsing and Dependency Grammar", In Carston, Robyn (ed.), UCL Working papers in Linguistics, University college London, 1989, pages 296 - 319.

[16] J. Eisner and G. Satta, "A Faster Parsing Algorithm for Lexicalized Tree- Adjoining Grammars", 5th Workshop on Tree-Adjoining Grammars and Related Formalisms (TAG+5), 2000.

[17] J. Eisner, "Bilexical grammars and a cubic-time probabilistic parser", In Proceedings of the 5th Int. Workshop on Parsing Technologies, 1997.

[18] D. Grinberg, J. Lafferty and D. Sleator, "A Robust Parsing Algorithm for Link Grammars", In Proceedings of the Fourth International Workshop on Parsing Technologies, 1995. Also issued as CMU Technical Report CMU-CS-95-125.

[19] T. Giang, R. Mooney, C. Peters, C. O'Sullivan, and C. Trinity, "Real-Time Character Animation Techniques", Image Synthesis Group College Dublin, Computer Science Department, Technical Report, 2000.

[20] P. Jorissen, M. Wijnants and W. Lamotte, "Using Collaborative Interactive Objects and Animation to Enable Dynamic Interactions in Collaborative Virtual Environments", IEEE Transactions on Visualization and Computer Graphics vol. 11, No. 6, 2005, pages 649 - 60.

[21] Rick Parent, "Computer Animation: Algorithms and Techniques", Morgan-Kaufmann, (Book), 2001.

[22] https://www.techradar.com/news/ray-tracing

[23] OpenGL Web Site: http://www.opengl.org.

[24] J. Kessenich , G. Sellers and D. Shreiner, "OpenGL Programming Guide: The Official Guide to Learning OpenGL", Version 4.5 with SPIR-V, Pearson Education (US), 2016.

[25] A. Feng, Y. Huang, M. Kallmann, and A. Shapiro, "An Analysis of Motion Blending Techniques", In Proceedings of the 5th International Conference on Motion in Games (MIG), 2012, pages 232–243.

[26] R. Parent, "Computer Animation: Algorithms and Techniques", Newnes, Oxford (2012).

[27] M N Risteiu, S D Rosca, and M. Leba, "3D modelling and simulation of human upper limb", IOP Conference Series: Materials Science and Engineering, Vol. 572, International Conference on Innovative Research - ICIR EUROINVENT, 2019.

[28] P Nogueira and V. zborniku, "Motion capture fundamentals: a critical and comparative analysis on real-world applications", 4th International Conference on Information, 2011.