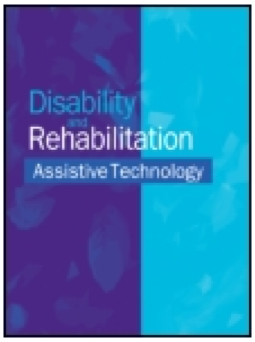

Taylor & Francis
Taylor & Francis Group

Disability and Rehabilitation: Assistive Technology

ISSN: 1748-3107 (Print) 1748-3115 (Online) Journal homepage: https://www.tandfonline.com/loi/iidt20

A non-expert-user interface for posing signing avatars

Nicoletta Adamo-Villani, Voicu Popescu & Jason Lestina

Disability and Rehabilitation: Assistive Technology, 2013; 8(3): 238–248

ISSN 1748-3107 print/ISSN 1748-3115 online
DOI: 10.3109/17483107.2012.704655

informa
healthcare

PRODUCTS AND DEVICES

A non-expert-user interface for posing signing avatars

Nicoletta Adamo-Villani[1], Voicu Popescu[2] & Jason Lestina[3]

[1]Department of Computer Graphics Technology, Purdue University, West Lafayette, IN, USA, [2]Department of Computer Science, Purdue University, West Lafayette, IN, USA, and [3]Department of Computer Graphics Technology, Purdue University, West Lafayette, IN, USA

We describe a graphical user interface designed to allow non-expert users to pose 3D characters to create American Sign Language (ASL) computer animation. The interface is an important component of a software system that allows educators of the Deaf to add sign language translation, in the form of 3D character animations, to digital learning materials, thus making them accessible to deaf learners. A study indicates that users with no computer animation expertize can create animated ASL signs quickly and accurately.

Keywords: American Sign Language, character animation, Graphical User Interfaces

Implications for Rehabilitation

- Deaf education, especially in science, technology, engineering and math (STEM), is a pressing national problem in the U.S. Deaf individuals are significantly underrepresented in STEM fields.
- An important underlying cause of the educational lag is that young deaf students have limited access to grade-level curriculum materials (because of their low English literacy level).
- Computer animation of American Sign Language (ASL) has the potential to improve learning outcomes by making educational content deaf accessible, thus providing deaf children with the same learning opportunities as hearing students.
- The work reported in the paper shows that it is possible for deaf educators, who are not animators, to annotate digital lessons with ASL translation in the form of animated avatars, thus making them accessible to deaf children.

Introduction

Deaf education, especially in science, technology, engineering, and math (STEM), is a pressing national problem in the U.S. Deaf individuals are significantly underrepresented in STEM fields and historically have had difficulty entering higher education leading to STEM careers [1,2]. An important underlying cause of the educational lag is that deaf students have limited access to grade-level curriculum materials. Computer animation of American Sign Language (ASL) has the potential to improve learning outcomes by making educational content deaf accessible, thus providing deaf children with the same learning opportunities as hearing students. Computer animation provides a low-cost and effective means for adding signed translation to any type of digital content.

Compared to video, animation technology has two fundamental advantages. The first one is *scalability*. Animated signs are powerful building blocks that can be concatenated seamlessly using automatically computed transitions to create new ASL discourse. By comparison, concatenating ASL video clips suffers from visual discontinuity. The second advantage is *flexibility*. Animation parameters can be adjusted to optimize ASL eloquence. For example, the speed of signing can be adjusted to the ASL proficiency of the user, which is of great importance for children who are learning ASL. The signing character can be easily changed by selecting a different avatar, hence the possibility of creating characters of different age and ethnicity, as well as cartoon characters appealing to young children.

However, ASL animation currently falls short of reaching its potential in deaf education. There is no easy-to-use public domain authoring system that allows educators to create learning materials annotated with animated ASL. An important piece of functionality that such a system has to provide is to *allow educators to animate new ASL signs*. This

Correspondence: Nicoletta Adamo-Villani, Purdue University, Department of Computer Graphics Technology, 401 N. Grant Street, West Lafayette, IN, 47907, USA. E-mail: nadamovi@purdue.edu

(Accepted June 2012)

is important for several reasons. There are thousands of ASL signs and an initial database can realistically only cover basic signs; a given ASL sign might need to be animated in several ways to reflect stylistic preferences; like any other language, ASL evolves continually and new signs enter the language all the time; outsourcing the animation task to an expert animator every time a new sign is needed is a solution that does not scale (time and remuneration costs, slow feedback loop).

We are developing a software system for allowing educators to author deaf-accessible math and science digital learning materials for grades 1–3 (Figure 1). One major challenge is the interface for posing the signing character. A preliminary user study revealed that educators find interfaces similar to those used in commercial animation systems very difficult to use. Many users gave up, and those who succeeded took over 20 min for a single ASL sign.

In this paper, we describe a novel interface that allows users knowledgeable in ASL but with only basic computer literacy and with no computer animation expertize to pose characters to create new signs (Figure 2, Figure 8). We also refer the reader to the accompanying video, which can be downloaded from http://www2.tech.purdue.edu/cgt/i3/DR_AT.

We differentiate between *non-expert* users targeted by our work and *novice* users. Our users are expert educators and it is not intended that they ever become expert animators. Consequently the goal is to provide a near-zero learning curve interface that is rapidly adopted by a large number of educators, and not to train educators to become expert animators. The interface design incorporates the following principles.

First, the users' knowledge of ASL is leveraged to make the interface more efficient. Instead of always starting from the neutral pose, the user has the option of loading a hand shape similar to the one targeted. The hand shapes available to the user are the digits and letters of the English alphabet. These hand shapes are well known to ASL users, they span the space of possible hand shapes, and they can be easily invoked by pressing the corresponding key.

Second, all selection operations are performed unambiguously in 2-D, which avoids problems with occlusion or poor separation between selection targets, without requiring view adjustments.

Third, the tens of degrees of freedom (DOFs) of the character are decomposed hierarchically such that the user manipulates only a single DOF at a time. Once the bottom of the hierarchy is reached, individual DOFs are selected using buttons labeled with an animation that previews the effect of manipulating that particular DOF. This way the user is more likely to select and manipulate the correct DOF, avoiding a trial and error approach.

We have conducted a user study to evaluate the proposed interface. The subjects were asked to pose the avatar to form the "I love you" (Figure 2) and "Apple" (Figure 8) ASL signs. All subjects completed the tasks successfully with no prior training other than a tutorial given right before the experiment begun. The average task completion times for the two signs were 97 and 181 s, respectively, a substantial improvement over the 20 min or more needed with the conventional interface. These results indicate that the proposed non-expert-user interface could remove one of the major barriers precluding ASL animation from becoming a widespread solution for making digital materials accessible to deaf learners.

Prior work

Although our work is supported by results in deaf education research that suggest visualization, interaction, and engagement as key prerequisites for effective learning for young deaf children, a comprehensive review of learning theories, pedagogical approaches, and prior interventions is beyond the scope of this paper. We limit the discussion of prior work to research aimed at generating and using computer animation of sign language, and at simplifying computer animation interfaces.

Sign language animation

Computer animation of sign language is a valuable tool for improving deaf education [3,4]. Several groups have been focusing on research, development and application of

Figure 1. First-grade math learning activity annotated with ASL animation using our prototype system.

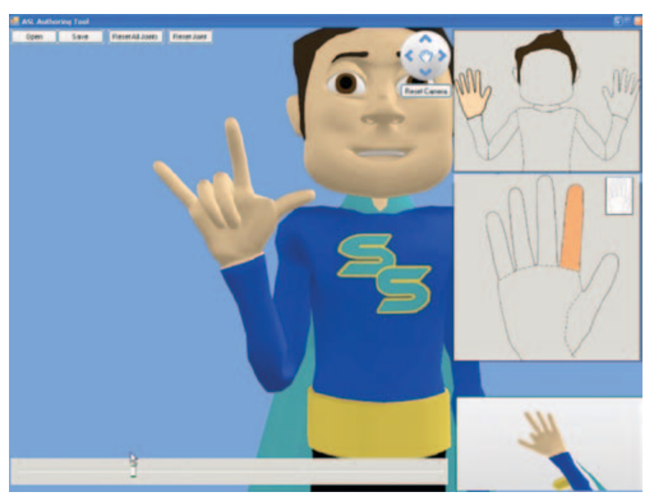

Figure 2. Proposed interface for posing singing avatar. Here the user animates the "I love you" sign.

computer animation technology for enhancing deaf accessibility to educational content, including ViSiCAST [5], Vcom3D [6], the Technical Education Research Center (TERC [7]), and our group at Purdue University [8].

The ViSiCAST project [5], later continued as the eSIGN project [9], aims to provide deaf citizens with improved access to services, facilities, and education through animated British Sign Language. The project is developing a method for automatic translation from natural-language to sign-language. The signs are rendered with the help of a signing avatar. A website is made accessible to a deaf user by enhancing the website's textual content with a signed translation encoded as a series of commands. The eSIGN software installed on the user's computer interprets these commands, retrieves previously animated signs from an online sign database, and applies them to a local avatar. The system relies on the creation of a large library of animated signs. The interface for producing animated signs is text-based, requiring users to be familiar with the eSIGN notation system and scripting language, as well as with character animation. Our work is complementary to this effort – our interface can help achieve and maintain a large and up-to-date sign database.

Vcom3D [6] commercializes software for creating and adding computer animated ASL translation to media. The SigningAvatar® software system uses animated 3-D characters to communicate in sign language with facial expressions. It has a database of 3500 English words/concepts and 24 facial configurations, and it can fingerspell words that are not in the database. The Sign Smith Studio® and Gesture Builder® [6] systems enable the creation of new ASL signs to overcome the database limitations. The system is intended for digital content creators and requires experience with character animation.

TERC collaborated with Vcom3D and the National Technical Institute for the Deaf (NTID) on the use of SigningAvatar® software to annotate the web activities and resources for two Kids Network units [10]. Recently, TERC has developed a Signing Science Dictionary [11]. Both the Kids Network units and the science dictionary benefit deaf children confirming again the value of animated ASL. However, the animated signs were produced by experienced programmers and animators, rather than by ASL educators, and thus the content cannot be easily extended.

The Purdue University Animated Sign Language Research Group [8], in collaboration with the Indiana School for the Deaf, is focusing on research, development, and evaluation of 3-D animation-based interactive tools for improving math and science education for the Deaf. The group developed Mathsigner™, a collection of animated math activities for deaf children in grades K-4, and SMILE™, an educational math and science immersive game featuring signing avatars [12]. As this content proves to be engaging and effective in the classroom, the team sets out to develop the ASL system, an authoring tool that allows all educators of deaf children to produce ASL annotated learning activities. In an initial formative evaluation of the system the ASL animations were perceived as accurate and fluid, but the system was perceived as difficult to use [13].

Many research efforts target automated translation from text to animated sign language to give signers with low reading proficiency access to written information. In the U.S., English to ASL translation research systems include those developed by Zhao et al. [28] and continued by Huenerfauth [29], and by Grieve-Smith [30]. The eSIGN project [9] provides text to sign language animated translation in the U.K. Translation to Greek Sign Language (SL) is pursued by Efthimiou's group [31], to German SL by Bungeroth [32], and to Irish SL by Morrissey and Way [33], to name just a few. Text to sign language translation is a problem orthogonal to that addressed by our work. Our interface will help create sufficient intelligible, expressive, and appealing signs that can be used to translate text. Moreover, we target the annotation of learning materials for young learners, where the amount of text is small and translation is not a bottleneck.

Computer animation interfaces

Animating a character is a challenging task that has been approached from several directions.

One approach is to give up on trying to *synthesize* animation and rather focus on *recording* the animation data using a variety of motion capture technologies. Motion capture cannot be surpassed when it comes to recording high quality motion (e.g. dance, sports). However, in our context of ASL animation, motion capture has important limitations. First, the approach does not scale to thousands of educators due to equipment costs. Second, motion capture systems excel at recording body and limb poses and are weak when it comes to recording hand shapes. We tried using a glove [14] to capture hand shapes but the low level of accuracy required lengthy post processing. A glove is well suited for tracking basic gestures but not well suited for recording crossing fingers and contact or near contact between finger tips.

In order to alleviate the hardware bottleneck of motion capture, extensive research has been directed towards extracting animation poses from a single image and without the use of markers. Since a 2-D image does not fully define a 3-D pose, disambiguation is attempted using probabilistic modeling and learning [e.g. 16]. The lack of markers implies having to solve the challenging problem of segmentation. Single camera methods focus on estimating body poses and do not handle hand shapes.

Another approach is to rely on sketching as an indirect interface. The user generates a 2-D drawing of the character, which is then converted to values for the many DOF defining the pose of the character. The approach works well for articulated figure animation [17–20], but requires that users have the ability to draw. Like in the case of single image pose capture, sketching has the problem of difficult disambiguation between the many poses matching a given 2-D drawing. One option is to let the user select the desired pose from the multiple candidates [20], another option is to require the user to provide additional data through the 2-D drawing using sketching conventions [17], and a third option is to resort to multiple passes [18]. In the context of ASL the main challenges of animation by sketching are the need for artistic talent and the difficult applicability to hand shapes which are hard to sketch and suffer from occlusions.

A third approach for simplifying the task of animating a computer character is to rely on databases of prior poses. One system provides a database with 2.8 million prerecorded poses, which is queried with a direct interface based on sketching or with an indirect interface based on constraints [21]. Querying by sketching suffers of the disadvantages described above. Defining constraints that uniquely identify a pose requires overcoming a non-negligible learning curve. We do adopt the pose database idea. However, as described below, we employ a small database with a trivial querying mechanism.

A fourth approach is to devise novel input devices or modalities that allow the animator to specify the pose more intuitively than by using conventional graphical user interfaces manipulated with mouse and keyboard. One system allows a puppeteer to control a virtual puppet in real time using trackers embedded into cylindrical physical handles [22]. Another system enables the creation of 2-D animations with an ingenious use of a sketching surface that allows the user to position and time objects [23]. A third system defines a series of hand gestures (e.g. "pinch", "zoom", "rotate") that are captured with a camera. Unlike in motion capture where the gestures are the desired animation [15], here the gestures allow the user to interact with the 3-D model and to pose it for each key frame [24]. An important advantage of all these systems is that they engage the user's motor and cognitive skills (e.g. affective connection to character, sense of space and time), which makes the animation task more intuitive, with a lower learning curve. In our context, a major concern is achieving an input device and/or modality that are robust and inexpensive to enable scalability to thousands of educators.

Considerable research has been devoted specifically to simplifying animation for those who are not expert animators [18,23,24]. The consensus is that for such users interface simplicity is of paramount importance. Simplicity is a primary design concern in our work as well. Most efforts target novice users, whereas the K-Sketch project [23] has the merit of making a clear and important distinction between animators in training, i.e. novice animators, and users who have animation needs but do not intend to become animators, i.e. non-animators. As noted in the introduction our work targets the latter. Non-animators cannot afford a lengthy time investment for learning the craft of computer animation, as they are typically attempting to animate outside of their main charge. This emphasizes the requirement of a low learning curve. Moreover, non-animators want to apply computer animation in a specific domain, and pertinent knowledge and simplifying assumptions should be leveraged.

In conclusion, although computer animation of ASL has improved significantly over the past few years and shows strong potential for enhancing accessibility to digital media, its effectiveness and wide-spread use is precluded by the absence of an easy-to-use public domain authoring system that allows users without animation expertize to create ASL animations that can be embedded into digital content.

Interface challenges in character animation

In this section, we analyze the process of character animation in order to isolate potential challenges faced by non-expert animators and to motivate the choices made in the design of our user interface. The challenges and proposed solutions are identified based on direct observation during formative evaluation with educators of deaf children and based on our experience with 3-D animation software and with the challenges faced by students trying to learn animation. The process of animating a 3-D character involves three steps: (1) setting key poses, (2) interpolating between poses, and (3) refining the poses and interpolation to perfect the motion and timing of the character. Steps (1) and (3) are usually performed by the animator, whereas step (2) is carried out automatically by the animation software.

The character posing challenge

Posing the character as required by steps 1 and 3 is the most complex and time-consuming task, as a human character model is defined by a large number of skeletal joints, many of which have multiple DOFs. Current commercial 3-D animation systems, such as Maya, 3DS Max, Poser, SoftImage [25,26], provide users with two approaches for posing avatars: Inverse Kinematics (IK) and Forward Kinematics (FK). IK allows users to manipulate only the extreme joints of a character (e.g. the wrist joint to pose the entire arm), and then the software interactively updates the position of the intermediate joints using a variety of IK algorithms. With FK the user rotates each individual skeletal joint to attain the desired pose.

The advantage of IK is that it allows users to create a rough pose fairly quickly. The main disadvantage is that it is difficult to refine the position of the intermediate joints. These joints cannot be manipulated directly and non-expert animators find it challenging to achieve the exact desired pose by only manipulating the extreme joints. On the other hand, FK is more time consuming but it allows the user to pose a character very accurately. Character manipulation is constrained to a single joint at a time. The user makes slower but steady progress from the root to the leaves of the dependency tree defined by the skeleton.

However, a joint can have up to three (rotational) DOFs and the user needs to select and manipulate the correct one. The typical solution adopted by animation interfaces is to annotate the joint with a 3-D construct (i.e. a gizmo) that suggests the DOFs of the joint and allows the animator to select and manipulate individual DOFs (Figure 3). Such gizmos are often perceived as confusing and difficult to use by inexperienced animators.

The selection challenge

To facilitate the selection and manipulation of the character's components, the majority of animation software packages offer Graphical User Interfaces (GUIs) that can be customized by experienced users to speed up the selection and transformation of the avatar's parameters. For example, it is possible to write scripts, expressions, or use reactive animation to create custom character controls and IK/FK switches that facilitate the process of posing the character. However, customizing the interface is not an easy task for a non-expert user, as it requires knowledge of the animation system scripting language and familiarity with advanced animation methods.

Regardless of whether an IK or an FK approach is taken, an important problem in the context of a non-expert user is the selection of a character control for manipulation. An intuitive solution is to let the user click on the character control directly in the main window. Due to the 3-D nature of the character, it might be challenging to select a given control: the control could be difficult to separate from other controls that project to the same part of the screen, or the control could not be visible at all due to occlusions. For the approach to work, the user has to have the ability of defining an adequate view where the control is visible and well separated from other controls (Figure 4).

The starting pose challenge

An obvious way of accelerating the posing of the character is to not always start from a neutral pose, but rather to start from an existing pose that is as close as possible from the desired new pose. The benefits of a better starting point are particularly important for a non-expert user. Animation systems (e.g. Poser [26]) do allow the user to select a character configuration from a pose library and use it as a starting point. One problem is that such libraries contain a limited set of poses, whereas an articulated human model can assume millions of different configurations. Therefore there is a high probability that a similar pose does not exist in the library. Another problem is that accessing and searching the vast libraries of pre-made poses is challenging for a non-expert user, requiring either learning how to formulate textual queries that describe the desired pose or the painstaking examination of hundreds of thumbnails towards identifying the best starting pose.

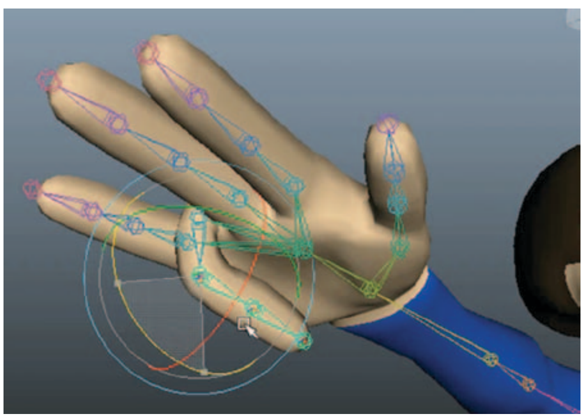

Figure 3. Gizmo provided by commercial animation system interface for rotating finger joint.

The ASL animation context

Posing the character is a particularly challenging process when animating a signing avatar. Animation of sign language requires very precise configuration of hands, which are one of the most complex parts of the human body. A hand 3-D model is usually rigged with a skeletal deformation system comprised of 24 to 27 joints and some of these joints have multiple DOFs. For example, the metacarpophalangeal joints have two rotational DOFs – pitch and yaw – to produce finger flexion and abduction. After selecting the joint, the animator needs to control both DOFs and understand the effect that each one has on the motion of the finger. The complexity of the hand exacerbates the challenges non-expert animators have with gizmos for DOF selection.

Furthermore, ASL hand gestures can be very complex and the position of some of the fingers can occlude the view of the rest of the hand, as, for example, in the hand shapes of the ASL signs for the letters "N" and "M". This requires changing the view constantly in order to reveal and separate the joints, a difficult task for non-expert users. The starting pose challenge is somewhat simpler in the ASL context since the animation system knows that the animator is only interested in creating an ASL sign, which reduces the size of the database of pre-made poses and presents opportunities for simplifying the query formulation.

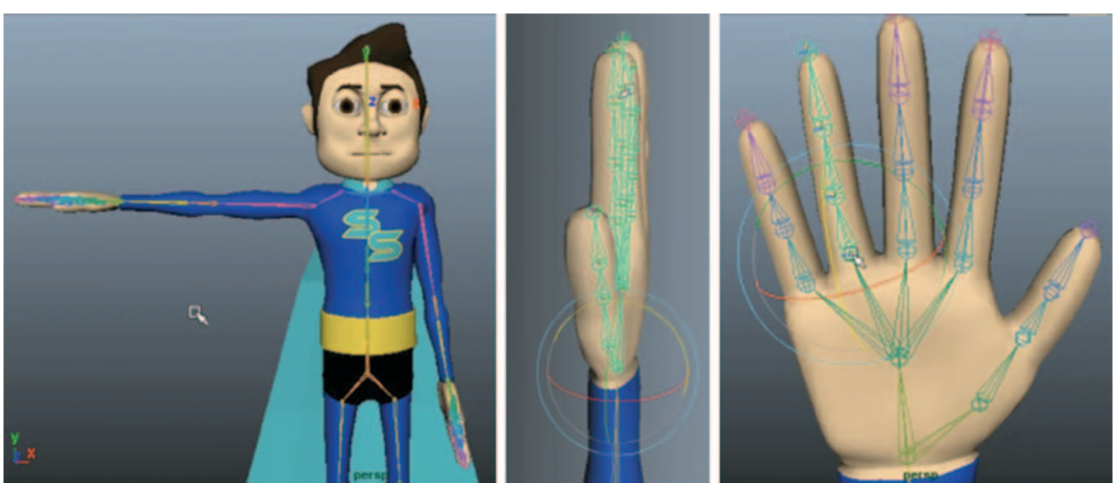

Figure 4. Inadequate (*left* and *middle*) and adequate (*right*) view for selecting the finger joints in commercial animation system.

In conclusion, the process of posing 3-D human-like models in general and signing avatars in particular can be a difficult and frustrating endeavor for inexperienced users if given an interface designed for expert computer animators. What is needed is an interface specifically designed for non-expert users.

Non-expert user interface

The proposed interface for posing signing avatars addresses the challenges discussed above as follows.

Selection

We avoid the problem of occlusions and poor separation between selection targets using 2-D selection maps. The selection maps show the character in a fixed pose that reveals all selection targets well. The selection maps do not change as the pose of the character changes, so the user can always select any selection target with one or two mouse clicks. To achieve an adequate level of detail on hands, selecting a hand displays a selection map of just the hand (Figure 5, top). Selection maps eliminate the need to change the view to reveal the selection targets. Another reason for changing the view is to better see the part of the model that is being posed. We reduce this need for changing the view by framing the selection – the view changes automatically to show the selection in detail.

Character posing

Our interface allows posing the character through forward kinematics (FK), which, as discussed above, is more suitable for non-expert users. In order to avoid the confusion about which joint needs to be selected to move a particular segment of the character body, the interface allows selecting body segments instead of joints. When a body segment is selected, FK manipulation acts on the upstream joint. For example, when a user attempts to pose the arm of a character there might be confusion as to whether to select the shoulder or the elbow. Whereas the shoulder is the joint that rotates to position the arm through FK, some users attempt to translate the elbow in IK fashion. Instead of burdening the user with the distinction between FK and IK, we simply place the selection targets on body segments and not on joints (e.g. right arm and last index segment in Figure 5).

A body segment can have 1, 2, or 3 DOFs according to the DOFs of the underlying upstream joint. Once a body segment is selected, the interface assists the user in the process of choosing the DOF to be manipulated by displaying one button for each DOF. The button is labeled with an animation that illustrates the effect of manipulating that particular DOF. In Figure 6, *top*, the 3 buttons at the bottom right of the image show how the hand rotates by manipulating each of the 3 DOFs. The animated labels preview in parallel all possible manipulations of a body segment without committing any pose change to the character. The user is more likely to select and manipulate the appropriate DOF without the frustration of having to decipher cryptic gizmos and of sequentially trying multiple DOFs and undoing unwanted effects.

Once the user decides which DOF needs to be manipulated, the DOF is selected by pressing the corresponding button which reveals a simple horizontal slider that commits the change to the character pose (Figure 6, *bottom*).

To accelerate the posing of the hand, the interface allows the user to curl a finger by manipulating a single slider. To

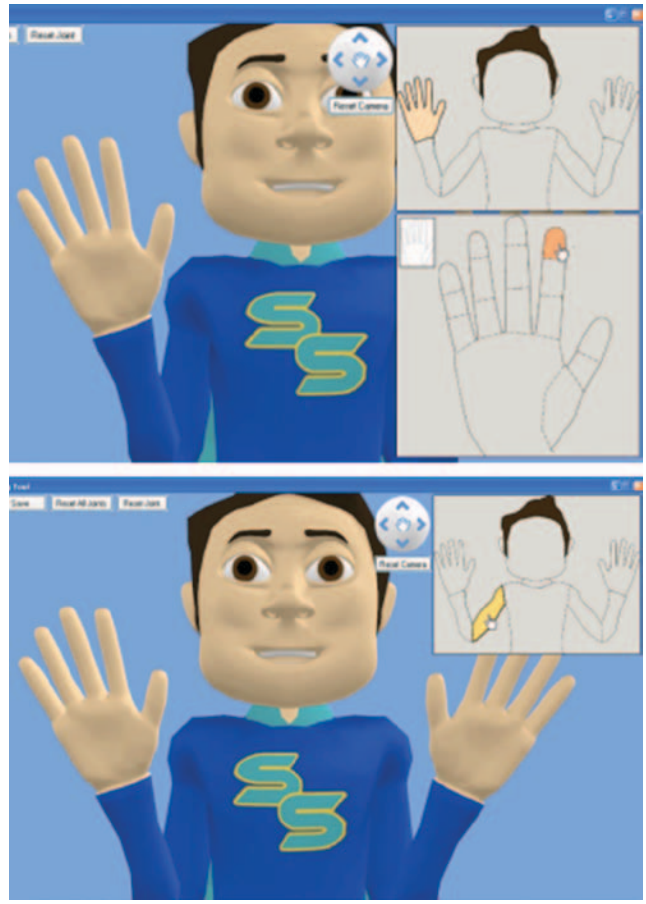

Figure 5. 2-D map selection of tip of right index (*top*) and of right arm (*bottom*) with proposed interface.

Figure 6. Buttons with animated labels previewing the effect of manipulating the 3 DOFs of the hand (*top*) and manipulation of lateral hand rotation using slider (*bottom*).

enable finger curling, the interface provides two selection maps for the hand. One is the selection map seen in Figure 5, *top*, which allows selecting individual finger segments, and one is the selection map seen in Figure 6, which allows selecting entire fingers. When an entire finger is selected the slider curls the finger. The user can toggle between the two selection maps for the hand by simply clicking on an icon of the other map.

Starting pose

A good set of starting poses is a set that samples the space of all possible poses well, that has as few poses as possible, and that can be easily queried by the user. In our context of ASL animation, the biggest challenge comes from posing hands so instead of defining a set of starting poses we define a set of starting hand shapes. This reduces the dimensionality of the starting set while still providing most of the posing efficiency gains.

The ASL linguistics literature does not define a canonical set of hand shapes, i.e. a small set of hand shapes from which all other hand shapes can be easily created.

The process of defining a canonical set of hand shapes has to start from defining the space of all possible hand shapes in ASL, which is difficult since there are thousands of signs and new signs are added continually. The second step is to define a metric for quantifying the difference between two hand shapes. One possible metric is the sum of corresponding angle differences between the two hand shapes over all joints and all DOFs. Another possibility is an interface specific metric that measures the average time it takes to convert the starting hand shape (i.e. the hand shape in the canonical set) into the destination hand shape (i.e. the new hand shape), using a given interface. Given a set of starting hand shapes and a new hand shape, the distance between the set and the new hand shape is defined as the smallest distance between a hand shape in the set and the new hand shape. In a third step, a set of N hand shapes has to be found that minimizes the sum of the distances between the set and all other hand shapes. This amounts to a complex optimization problem with hundreds of DOFs.

Moreover, the suitability of a canonical set also depends on how easy it is for the user to retrieve the hand shape from the set that is closest to a given hand shape. Sequential inventorying of the canonical set is slow, and a set of hand shapes that can be quickly invoked should be preferred.

After consulting with ASL linguists and users we have converged on a canonical set defined by the 36 ASL hand shapes for the letters and digits of the English alphabet (Figure 7). The set covers a great range of hand shapes, all well familiar to ASL users. Given a new hand shape, an ASL user can quickly identify mentally the closest letter or digit, which is invoked intuitively by pressing the corresponding key.

User study

The interface described in the paper is an important component of an authoring system that enables ASL educators to create sign language animations to annotate digital learning content. We compared our UI against an earlier version of our system, which had an interface modeled after conventional interfaces such as those of Maya or Gesture Builder (e.g. selection in 3-D, rotations via gizmos, direct manipulation of multiple degrees of freedom, and the option of inverse kinematics). The earlier system and its formal evaluation are described in detail in a publication [13]. In summary, only 1 out of 5 deaf educators was able to pose an ASL sign, and this in over 20 min. Also the subject survey indicates that the earlier system was perceived as very difficult to use. One of the subjects gave up and left during the study. The main complaints pertained to difficulty in selecting joints via gizmos, in predicting the effect of manipulation, and in adjusting the view to focus on the character part currently posed. This preliminary study provides the baseline for the present study.

Subjects

10 participants, 8 hearing ASL signers and 2 deaf ASL signers, age 22–53 years, 2 males, 8 females. 5/10 participants were very fluent in ASL; the other 5 participants were students in an advanced ASL class and had a good knowledge of ASL, as they had taken 3 ASL courses prior to participating in the experiment. All participants had basic familiarity with computers and standard input devices (i.e. mouse and keyboard). None of the participants had any prior experience with 2D or 3D animation software. Although hearing condition and age were not the same for all subjects, this was intentional to ensure that the subjects tested are a representative sample of the population targeted by our software: 2 subjects were deaf, 3 subjects were hearing CODAs (i.e. Children of Deaf Adults), and 5 were hearing ASL students.

Procedure

Participants were first given instructions (in ASL and spoken English) and a 15-min demonstration on how to use the system and posing interface with mouse and keyboard. Then subjects were asked to perform five tasks: (1) Import a 3-D character into the system and load the database of signs, (2) Use the existing database of signs to create an ASL animated sentence, (3) Reproduce a facial expression represented in a provided image (4) Use the script editor to animate a specific ASL sentence, and (5a and 5b) Create the "Apple" and "I love you" signs using the user interface described in the paper. We focused on posing the character because the transitions between poses are provided automatically by the system, with collision detection, and with a variety of interpolation types.

Subjects worked alone at a personal computer. The task completion time was recorded by a key press which started and stopped the timer and video screen capture. The video recordings were later analyzed to determine the nature and number of mistakes and possible "bugs" in the system and user interface. Subjects were instructed to create accurate and legible signs as quickly as possible. Some subjects chose to refine their signs whereas others did not. After the experiment the signs were evaluated for accuracy and legibility by two deaf ASL signers.

After completing the 5 tasks, participants were directed to fill out a web survey. The first part of the survey included questions related to gender, age, ASL fluency, familiarity with computers,

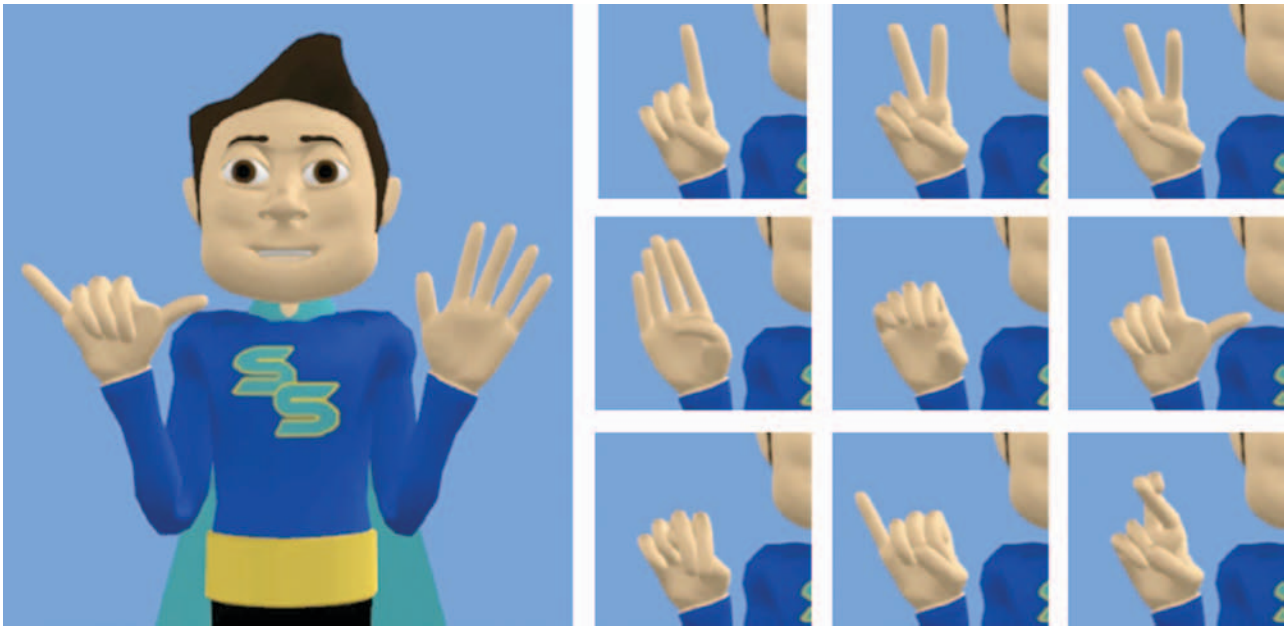

Figure 7. Hand shape for letter Y used as a starting point for posing the "I Love You" sign shown in Figure 2 (*left*), and other examples of starting hand shapes (*right*).

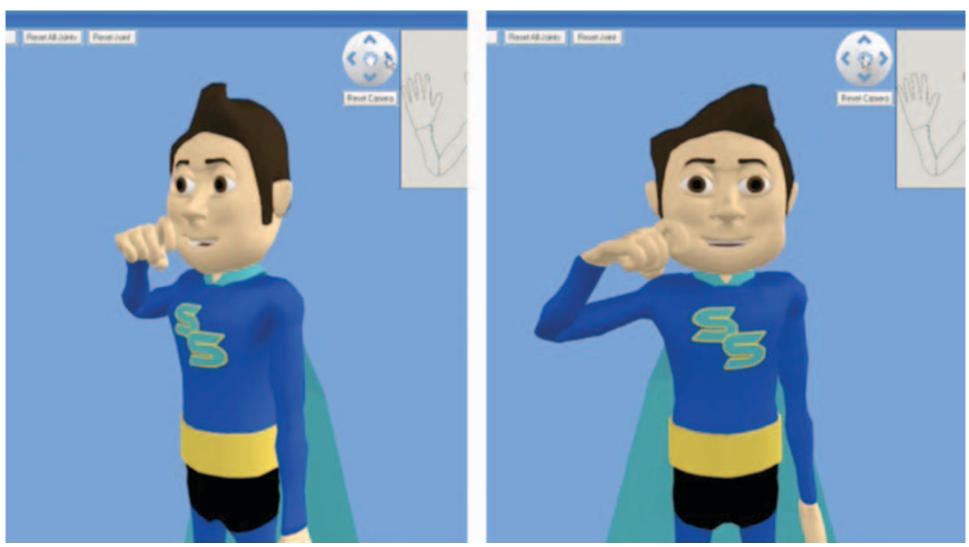

Figure 8. "Apple" ASL sign. The pose is challenging because the right index has to touch the cheek.

and experience in 3D animation. The second part of the survey included rating questions focusing on system usability.

Findings

We report the results for tasks 5a and 5b as they pertain to the effectiveness of the proposed interface. All subjects were able to produce all the signs correctly; all signs were rated accurate and legible. For task 5a (posing the "Apple" sign), the subjects' times were: MEAN = 182 s; MIN = 31 s; MAX = 285 s; STDV = 79 s. For task 5b (posing the sign for the "I love you" sign) the subjects' times were: MEAN = 97 s; MIN = 51 s; MAX = 182 s; STDV =38 s. There were no significant differences with age and hearing condition.

The results of the survey show that subjects found the posing interface easy-to-use (MEAN = 4, on a scale of 1 "not easy" to 5 "very easy"), and familiarity with computers was not a significant correlate. Regarding the proposed user interface, participants commented: "*I liked the easy manipulation of the hands and arms to make each sign as perfectly as possible; it was really easy to create signs with minimal instructions; I liked the premade poses, more premade poses would be useful… also for the arms*".

The analysis of the video screen recordings shows that all 10 subjects were able to identify and load without hesitation the Y and X hand shapes as starting poses for the creation of the "I Love You" and "Apple" signs, respectively. This confirms the usefulness of a library of canonical ASL hand shapes that ASL users are familiar with.

Participants did not experience any difficulty selecting the character components and manipulating the sliders. However, 3 out of 10 subjects did not seem to understand the effect of manipulating some of the sliders, as they repeatedly selected the incorrect ones. This finding suggests that the animated preview of the effect of manipulation of a specific DOF might not be clear to all users. Some users were able to use the interface extremely efficiently – for example to pose the "Apple" sign in 31 s.

One interesting finding is that the majority of the subjects spent 30–40% percent of the total time for making the sign posing the left arm, which does not actually sign, and only has to be brought down from the neutral elevated position (see for example Figure 7, *left*) to a resting position parallel to the body (see for example Figure 8). The time for creating a sign could be significantly shortened by providing a library of commonly used ASL *arm* poses that the user can load as starting positions.

Lastly, subjects took much longer to make the "Apple" sign than the "I Love you" sign. Observation of the subjects' screen interaction shows that many participants experienced difficulties in positioning the index finger such that it touches the character's cheek (Figure 8). We believe that this difficulty is due to the subjects' inexperience with 3-D spatial representation and 3-D view manipulation. Assisted view changing, which we plan to implement in the future, might alleviate this problem.

The results of the study are promising as they demonstrate that the proposed UI is usable by non-expert users and functional – all subjects were able to create the signs. The time required to produce a new sign is acceptable (i.e. 30 s–285 s) and the interface does not appear to have a steep learning curve. Descriptive statistics including all subjects' performance data are available at: http://www2.tech.purdue.edu/cgt/i3/DR_AT/performance_data.htm

Comparison to commercial animation software

As anyone who has used a commercial animation system such as Maya can attest, a user with no animation expertize cannot pose a character with such a system without training. The video accompanying our submission opens with an illustration of some of the complex steps necessary to pose a character in Maya.

We have also compared our UI to that of Gesture Builder, a commercial computer animation software system specifically designed for posing and animating gestures. We have conducted a thorough literature search for Gesture Builder user study data, without success. We then proceeded to conducting a comparison between our interface and that of Gesture Builder (GB) at a design principle level. Here are our findings:

- GB does not allow creating hand shapes, so the user is limited to preexisting hand shapes (see Figure 9). We tackle the problem of posing individual fingers and we allow creating any new hand shape. Figure 5 shows the selection of individual finger segments in the proposed interface; once a finger segment is selected, it can be rotated using a simple slider (Figure 2, bottom) in order to attain any hand pose.
- GB performs joint rotations via gizmos with multiple degrees of freedom (see Figure 10) whereas we select and affect individual rotations via animated button labels that preview and 1-D sliders that apply the rotation (see Figure 6, right and bottom). Gizmos were found very hard to use in our preliminary user study, as well as in our experience in teaching animation.
- GB uses IK by asking the user to define pairs between sets of tens of possible contact points (see Figure 11); the rough pose is then refined using FK. Our system uses an FK approach exclusively. As mentioned in the section "Non-expert user interface", the main disadvantage of IK is that it is difficult to refine the position of the intermediate joints of the IK chain. These joints cannot be manipulated directly and non-expert animators find it challenging to achieve the exact desired pose by only moving the extreme joints.

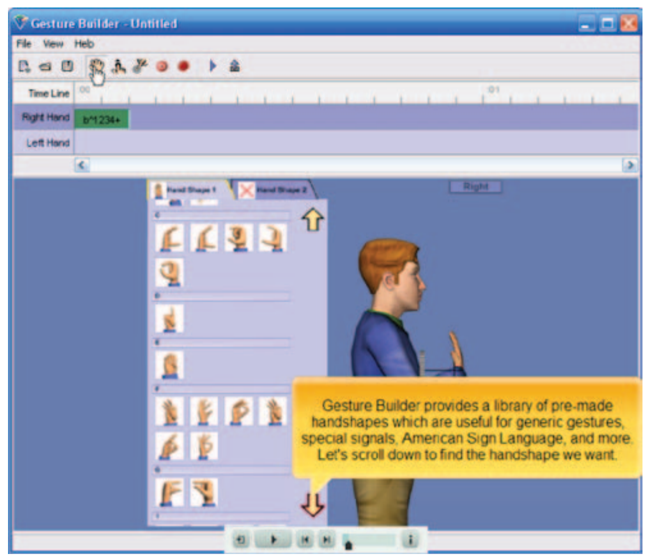

Figure 9. Screenshot from Gesture Builder software showing the library of pre-made hand shapes.

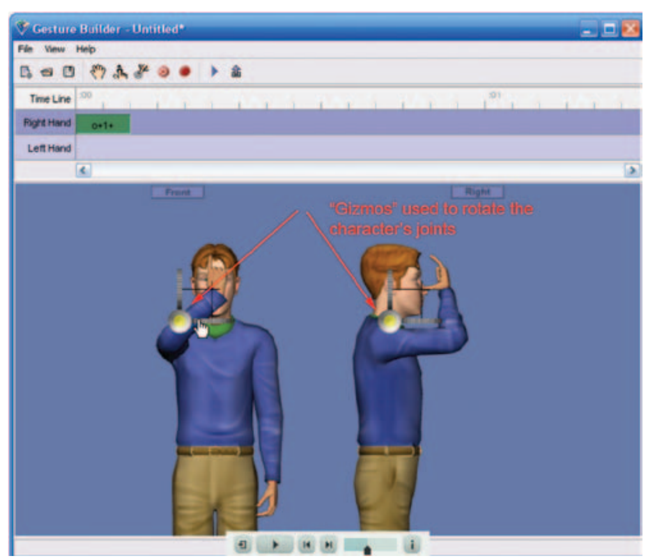

Figure 10. Screenshot from Gesture Builder software showing the joints' rotation "Gizmos".

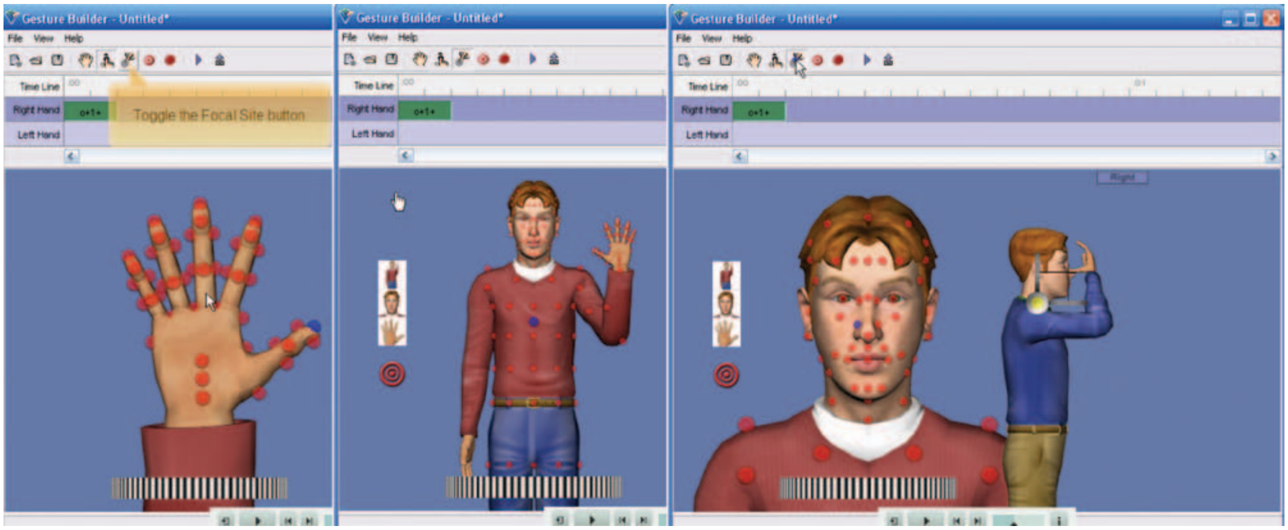

Figure 11. Screenshot from Gesture Builder software showing tens of contact points for the face, hands and torso.

- GB restricts the user to a set of predefined orthographic views, and it could happen that none is adequate for some signs. Our system automatically frames the character part currently being modified and also allows the user to change the view freely (see figure 8), including to create a top view.

Conclusions and future work

We have demonstrated an interface that allows users with no animation expertise to pose a signing avatar. We refer the reader to the accompanying video that further illustrates our interface and the learning activities that it enables.

So far we have focused on users with no prior experience in ASL animation. Our user study showed that the interface was usable with no training. As future work we will investigate whether interface efficiency can be improved as users become familiar with ASL animation and with our specific interface. Does the interface need to evolve and how? We foresee that there are opportunities for further reducing the time required to pose an avatar as the user gains experience, but we believe that it is unlikely that the interface will have to evolve to resemble the interfaces of commercial animation systems – the user does not evolve into an expert animator.

The interface will be integrated into a software system for annotating computer learning activities with ASL translations. The interface promises to make the system accessible to all educators, an important prerequisite for a definitive solution to the accessibility problem faced by deaf education in our society. We plan to leverage the society level deaf-accessible content creation effort through an online community where ASL signs and ASL-translated materials are contributed, found, rated, and used by deaf learners, their parents, and their educators.

A plausible important additional benefit of our work is in the context of teaching ASL and of improving English literacy. Young deaf learners are faced with the daunting challenge of learning ASL, written English, and subject matters such as mathematics, all at the same. The combination of realistic ASL discourse delivered by the engaging avatar, of English subtitles, and of a graphical description of the content could prove to be an essential tool for overcoming this triple challenge. Since many deaf children have at least one hearing parent, who is typically not a signer [27], ASL animation could also prove to be useful in the context of teaching parents ASL, opening an additional, natural, communication channel with their young child.

Finally, facilitating the creation of ASL translation is important beyond the education domain and could be used for example in entertainment and social networking to remove communication barriers between hearing and non-hearing members of our society.

Declaration of interest

This work was supported in part by the National Science Foundation (award # 0622900) and by the Dr. Scholl Foundation. We thank Marcus Oania for his help with the project, and all the subjects who participated in the study.

References

1. Lang HG. Higher education for deaf students: research priorities in the new millennium. J Deaf Stud Deaf Educ 2002;7:267–280.
2. Burgstahler S. (Ed.) 2010. Making Math, Science, and Technology Instruction Accessible to Students with Disabilities. Seattle: DOIT, University of Washington.
3. Vesel J. Signing Science. Learn Leading Techno 2005; 32: 30–31.
4. Roush D. Providing Sign Language Access to the Science curriculum for deaf students. Proc. of 2004 Technology and Persons with Disabilities Conference, Cal. State U. (2004).
5. Bangham JA, Cox SJ, Elliott R, Glauert JRW, Marshall I. (UEA), Rankov S, Wells M. An Overview of ViSiCAST. IEE Seminar on "Speech and language processing for disabled and elderly people", London, 2000.
6. Available at: Vcom3D. http://www.vcom3d.com/
7. Available at: TERC. http://www.terc.edu/
8. Adamo-Villani N, Wilbur R. Software for math and science education for the deaf. Disabil Rehabil Assist Technol 2010;5:115–124.
9. Available at: eSIGN http://www.visicast.cmp.uea.ac.uk/eSIGN
10. Available at: EnviSciNetwork. http://www.enviscinetwork.com/signed_vers_toc.cfm
11. Available at: Signing Science. http://signsci.terc.edu/
12. Adamo-Villani N, Wilbur R. Two Novel Technologies for Accessible Math and Science Education. IEEE Multimedia -Special Issue on Accessibility, (2008), 38–46.

13. Hayward K, Adamo-Villani N, Lestina J. A computer animation system for creating deaf-accessible math and science curriculum materials. Proc. of Eurographics 2010 – Education Papers, (2010), Sweden. EG Digital Library.
14. Immersion Cyberglove. Available at: http://www.immersion.com.
15. Ishigaki S, White T, Zordan V, Liu K. Performance-based control interface for character animation. ACM Transactions on Graphics, 28: 3 (August 2009).
16. Rosales R, Sclaroff S. Specialized mappings and the estimation of human body pose from a single image. IEEE Workshop on Human Motion, 2000.
17. Mao C, Qin SF, Wright DK. A sketch-based gesture interface for rough 3D stick-figure animation. Proc. of Eurographics Workshop on Sketch-Based Interfaces and Modeling, (2005). EG Digital Library.
18. Jeon J, Jang H, Lim SB, Choi YC. A sketch interface to empower novices to create 3D animations. Computer Animation and Virtual Worlds, 21, (2010), 423–432.
19. Mao C, Qin SF,Wright DK. Sketching-out virtual humans: from 2D storyboarding to immediate character animation. Proc. of ACE 06, Hollywood, CA (2006).
20. Davis J, Agrawala M, Cuang E, Popovic Z, Salesin D. A sketching interface for articulated figure animation. Proc. of EG/SIGGRAPH Symp. on Computer Animation (2003).
21. Wei X, Chai J. Intuitive interactive human character posing with millions of example poses. IEEE Computer Graphics and Applications, 10 (Nov. 2009).
22. Oore S, Terzopoulos D, Hinton G. A desktop input device and interface for interactive 3-d character animation. Proc. of Graphics Interface, (2002), 133–140.
23. Davis RC, Colwell B, Landay JA. K-Sketch: a "kinetic" sketch pad for novice animators. Proc. of CHI 2008, Florence, Italy (2008), 413–422.
24. Chu E, Loidl K, Rosenbaum S. Animation for novices. (2007). Available at: http://www-cs.stanford.edu/people/ rosenbas/animator.pdf.
25. Autodesk Maya, 3DS Max, Softimage. Available at: http://usa.autodesk.com/
26. SmithMicro Poser. Available at: http://poser.smithmicro.com/dr/poser-left.html
27. Blanchfield BB, Feldman JJ, Dunbar JL, Gardner EN. The severely to profoundly hearing impaired population in the United States: Prevalence and demographics. Bethesda, MD: Project HOPE Center for Health Affairs. (A shorter Policy Analysis Brief [Series H, Vol. 1, No. 1, Oct. 1999]).
28. Zhao L, Kipper K, Schuler W, Vogler C, Badler N, Palmer M. (2000) A machine translation system from English to American Sign Language. AMTA Conference, LNCS 1934.
29. Huenerfauth M. 2004. A multi-path architecture for machine translation of English text into American Sign Language animation. In Proc. of the Workshop at the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL).
30. Grieve-Smith A. 2002. SignSynth: A sign language synthesis application using Web3D and Perl. In: Gesture and Sign Language in Human-Computer Interaction, I. Wachsmuth and T. Sowa, Eds. Vol. 2298. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 134–145.
31. Kouremenos D, Stavroula-Evita F, Efthimiou E, Ntalianis K. 2010. A prototype Greek text to Greek Sign Language conversion system, Behaviour & Information Technology, 29: 5, 467–481.
32. Bungeroth J, Ney H. 2004. Statistical sign language translation. Workshop on Representation and Processing of Sign Languages, 4th International Conference on Language Resources and Evaluation (LREC 2004). Portugal, 105–108.
33. Morrissey S, Way A. 2005. An example-based approach to translating sign language. Workshop on Sample-based Machine Translation (MT X-05). Thailand, 109–116.