

Multimodal tools and interfaces for the intercommunication between visually impaired and “deaf and mute” people

Konstantinos Moustakas, Georgios Nikolakis, Dimitrios Tzovaras, Benoit Deville, Ioannis Marras and Jakov Pavlek

Abstract— The present paper presents the framework and the results of Project 2: “Multimodal tools and interfaces for the intercommunication between visually impaired and “deaf and mute” people”, which has been developed during the eNTERFACE-2006 summer workshop in the context of the SIMILAR NoE. The developed system aims to provide alternative tools and interfaces to blind and deaf-and-mute persons so as to enable their intercommunication as well as their interaction with the computer. All the involved technologies are integrated into a treasure hunting game application that is jointly played by the blind and deaf-and-mute user. The reason for choosing to integrate the multimodal interfaces into a game application is that it serves both as an entertainment and as a pleasant education tool to its users. The proposed application integrates haptics, audio, visual output as well as computer vision, sign language analysis and synthesis, speech recognition and synthesis, in order to provide an interactive environment where the blind and deaf and mute users can collaborate in order to play the treasure hunting game.

Index Terms—Multimodal interfaces, Rehabilitation technologies, Virtual Reality.

I. INTRODUCTION

DURING the latest years there has been an increasing interest in the Human-Computer Interaction society for multimodal interfaces. Since Sutherland's SketchPad in 1961 or Xerox' Alto in 1973, computer users have long been

K. Moustakas is with the Informatics and Telematics Institute Centre for Research and Technology Hellas, 1st Km Thermi-Panorama Str. 57001 Thermi-Thessaloniki, Greece and the Aristotle University of Thessaloniki (moustak@iti.gr)

G. Nikolakis is with the Informatics and Telematics Institute Centre for Research and Technology Hellas, 1st Km Thermi-Panorama Str. 57001 Thermi-Thessaloniki, Greece (gniko@iti.gr)

D. Tzovaras is with the Informatics and Telematics Institute Centre for Research and Technology Hellas, 1st Km Thermi-Panorama Str. 57001 Thermi-Thessaloniki, Greece (tzovaras@iti.gr)

B. Deville is with Computer Vision and Multimedia Lab, University of Geneva, Geneva, Switzerlandn (e-mail: Benoit.Deville@cui.unige.ch).

I. Marras is with the Aristotle University of Thessaloniki, Informatics Scool AIIA Lab. Thessaloniki Greece (e-mail: imarras@aia.csd.auth.gr)

J. Pavlek is with the Faculty of Electrical Engineering and Computing, Zagreb, CROATIA (e-mail: jakov.pavlek@fer.hr)

acquainted with more than the traditional keyboard to interact with a system. More recently with the desire of increased productivity, of seamless interaction and immersion, of e-inclusion of people with disabilities, as well as with the progress in fields such as multimedia/multimodal signal analysis and human-computer interaction, multimodal interaction has emerged as a very active field of research (e.g. [1], [2]).

Multimodal interfaces are those encompassing more than the traditional keyboard and mouse. Natural input modes are put to use (e.g. [3], [4]), such as voice, gestures and body movement, haptic interaction, facial expressions and more recently physiological signals. As described in [5] multimodal interfaces should follow several guiding principles: multiple modalities that operate in different spaces need to share a common interaction space and to be synchronized; multimodal interaction should be predictable and not unnecessarily complex, and should degrade gracefully for instance by providing for modality switching; finally multimodal interfaces should adapt to user's needs, abilities, environment.

A key aspect in multimodal interfaces is also the integration of information from several different modalities in order to extract high-level information non-verbally conveyed by users. Such high-level information can be related to expressive, emotional content the user wants to communicate. In this framework, gesture has a relevant role as a primary non-verbal conveyor of expressive, emotional information. Research on gesture analysis, processing, and synthesis has received a growing interest from the scientific community in recent years and demonstrated its paramount importance for human machine interaction (see for example the Gesture Workshop series of conferences started in 1996 and since then continuously growing in number and quality of contributions; a selection of revised papers from the last workshop can be found in [6]).

The present work aims to make the first step in the development of efficient tools and interfaces for the generation of an integrated platform for the intercommunication of blind and deaf-mute persons. It is obvious that while multimodal signal processing is essential in such applications, specific issues like modality replacement and enhancement should be addressed in detail.

In the blind user's terminal the major modality to perceive a

virtual environment is haptics while audio input is provided as supplementary side information. Force feedback interfaces allow blind and visually impaired users to access not only two-dimensional graphic information, but also information presented in 3D virtual reality environments (VEs) [7]. The greatest potential benefits from virtual environments can be found in applications concerning areas such as education, training, and communication of general ideas and concepts [8]. Several research projects have been conducted to assist visually impaired to understand 3D objects, scientific data and mathematical functions, by using force feedback devices [9].

PHANToM™ is one of the most commonly used force feedback device; it is regarded as one of the best on the market. Due its hardware design, only one point of contact at a time is supported. This is very different from the way that we usually interact with surroundings and thus, the amount of information that can be transmitted through this haptic channel at a given time is very limited. However, research has shown that this form of exploration, although time consuming, allows users to recognize simple 3D objects. The PHANToM™ device has the advantage to provide the sense of touch along with the feeling of force feedback at the fingertip. Another device that is often used in such cases is the CyberGrasp that combines a data glove (CyberGlove) with an exoskeletal structure so as to provide force feedback to each of the fingers of the user (5DoF force feedback, 1DoF for each finger). In the context of the present work we used the PHANToM™ desktop device to enable haptic interaction of the blind user with the virtual environment.

Deaf and mute users have visual access to 3D virtual environments; however their immersion is significantly reduced by the lack of audio feedback. Furthermore effort has been done to provide applications for the training of hearing impaired. Such applications include the visualization of the hand and body movements performed in order to produce words in sign language as well as applications based on computer vision techniques that aim to recognize such gestures in order to allow natural human machine interaction for the hearing impaired. In the context of the presented framework the deaf-mute terminal incorporates sign-language analysis and synthesis tools so as to allow physical interaction of the deaf-mute user and the virtual environment.

The paper is organized as follows. Section II describes the overall system architecture and the objectives, Sections III and IV describe the SeeColOr and the haptic interaction modules respectively. In Section V the sign synthesis and sign recognition systems are briefly described. In Sections VI and VII the automatic grooved map generation and partial matching algorithm are described, respectively. The entertainment scenario is described in Section VIII. Finally, in section IX the conclusions are drawn.

II. OVERALL SYSTEM DESCRIPTION

The basic development concept in multimodal interfaces for the disabled is the idea of *modality replacement*, which is the

use of information originating from various modalities to compensate for the missing input modality of the system or the users.

The main objective of the proposed system is the development of tools, algorithms and interfaces that will utilize modality replacement so as to allow the communication between blind or visually impaired and deaf-mute users. To achieve the desired result the proposed system combines the use of a set of different modules, such as

- Gesture recognition,
- Sign language analysis and synthesis,
- Speech analysis and synthesis,
- Haptics,

into an innovative multimodal interface available to disabled users. Modality replacement was used in order to enable information transition between the various modalities used and thus enable the communication between the involved users.

Figure 1 presents the architecture of the proposed system, including the communication between the various modules used for the integration of the system as well as intermediate stages used for replacement between the various modalities. The left part of the figure refers to the blind user's terminal, while the right refers to the deaf-mute user's terminal.

The different terminals of the treasure hunting game communicate through asynchronous TCP connection using TCP sockets. The following sockets are implemented in the context of the treasure hunting game.

- SeeColor terminal: Implements a server socket that receives queries for translating color into sound. The code word consists of the following bytes, "b;R;G;B", where b is a boolean flag and R, G, B the color values.
- Blind user terminal: Implements three sockets.
 - A client socket that connects to the SeeColor terminal.
 - A server socket to receive messages from the deaf-mute user terminal
 - A client socket to send messages to the deaf-mute user terminal
- Deaf-mute user terminal: Implements two sockets
 - A server socket to receive messages from the blind user terminal
 - A client socket to send messages to the blind user terminal

Also file sharing is used to ensure consistency between the data used in the various applications.

III. SEECOLOR

SeeColOr is meant to be a mobility aid for visually impaired and blind people. Its main interest resides in the fact that colours will henceforth be accessible to these disabled people using their hearing. As you will see later, this is done using an association between colours and musical instruments. The project #2 being an implementation of different multimodal interfaces to make possible the communication

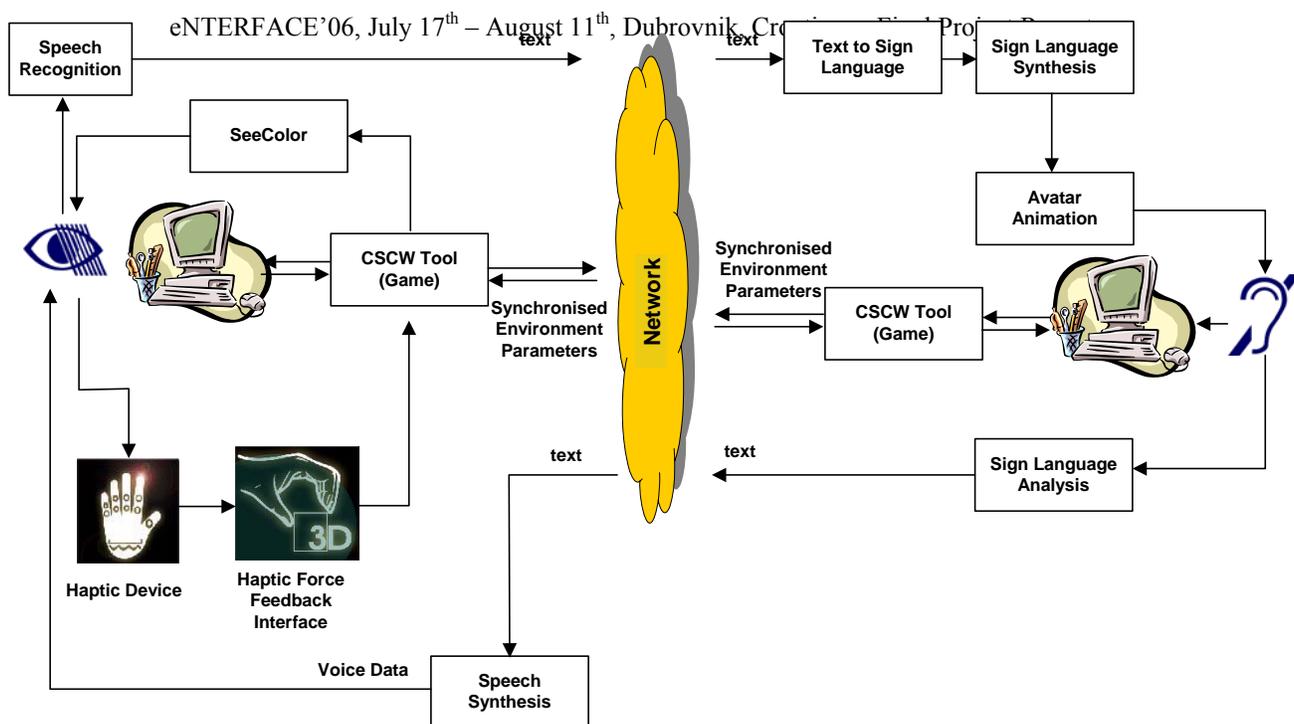


Figure 1. Block diagram showing the intercommunication between the various modalities used to provide communication between visually impaired (left part of the diagram) and deaf-mute users (right part).

between two different communities of disabled people, one of them being the visually impaired community, the interest of the integration of SeeColOr was obvious.

We will first describe the SeeColOr system from a global point of view, and its objectives. Then we will go into details, explaining how works the colour to instrument mapping. Finally, the way it has been integrated into the treasure hunt game for blind and deaf-and-mute people will be pictured, as for the adaptations we made to simplify and adapt it to the game.

A. SeeColOr system

Unlike other systems, SeeColOr's colour to sound mapping includes musical instruments. In fact, most of actualonification systems use either artificial sounds [10], [11], [12], [13] or meaningful sounds [12], [14], i.e. particular sounds or texts that are supposed to describe an object. For example, a car could be represented by the sound of a motor engine, or a horn sound; toilets could be well described by a flush sound [12]; and the leaves moving in winds are easily understood as being trees.

Each approach has qualities and drawbacks. For example, it is easy to describe edges using artificial sounds, but constantly listening to them is tiring, even annoying. There is no evidence that these sounds should not be used in a sensory substitution system, but it was intuitively thought that such a system should only produce sounds that are common to users. These sounds should be pleasant to hear. This is why common musical instruments were chosen.

The meaningful sound approach is really useful for known environments, and searching particular objects. Although easier to learn, such a system is useless in a new environment,

or with new objects. This is a classification problem, and in an unknown situation, one never knows how the system would react. This is the reason why we prefer to let the human brain makes its own deductions, and only sonify colour information.

B. Colour-instrument mapping

The system uses the HSL (Hue - Saturation - Lightness) colour system to define colours. The hue represents the pigment of the colour. Its value is expressed in degrees, from 0° to 360°. The variations of saturation from 0 to 1 make the colour go from grey to really intense colour. The lightness expresses the quantity of light, from black to white, and it is also a decimal value in [0,1]. Figure 2 illustrates this colour system.

Each dimension of this colour space is mapped to an auditory dimension. First, the hue is quantified into seven colours: red, orange, yellow, green, cyan, blue, and magenta. Each colour is then mapped as a linear combination of two particular instruments, according to the table 1. A pure colour is of course played by only one instrument timbre.

The pitch of the played note depends on the colour saturation, which is divided into four intervals. The four notes are C, G, B \flat , and E, thus making a dominant seventh chord.

Finally, the luminance –or brightness, or lightness– of the colour is represented by two different instruments with varying pitch. Dark and clear colours will be played respectively by the double bass and synthesized human voice. Here the pitch depends on the lightness value. The lightness interval is divided into eight parts; the four darker ones are played by the double bass and the four other by the synthesized human voice. The notes are the same as for the saturation, therefore a dominant seventh chord.

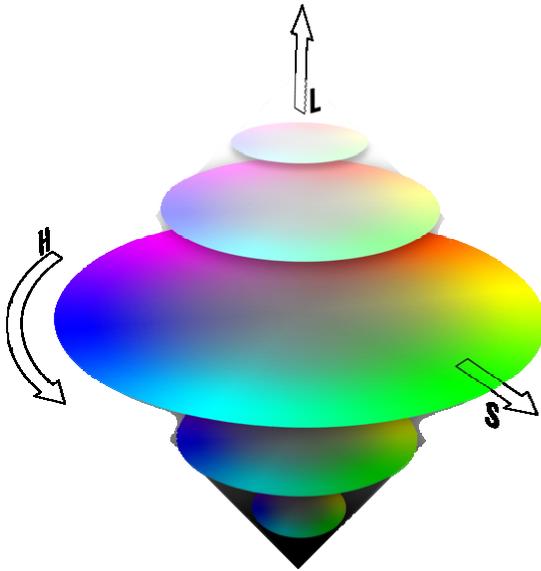


Figure. 2. HSL Double Cone.

TABLE 1
SEECOLOR'S COLOUR-TO-INSTRUMENT MAPPING

Colour	Instrument
red	oboe
orange	viola
yellow	violin (pizzicato)
green	flute
cyan	trumpet
blue	piano
magenta	Saxophone

This system has been tested on five sighted persons, and some useful conclusions have been made. First, some instruments are difficult to differentiate. Most people never clearly heard the viola, or had problems to distinguish all or part of wind instruments, i.e. saxophone, oboe, flute or trumpet. However, it was obvious that it was easy to say if the colour was either dark or bright. On the other hand, pitch modulations remained abstruse to them, both for saturation and lightness. All these observations will lead to the creation of a new orchestra in the near future.

C. Integration

In the context of the eNTERFACE'06 project, we decided to simplify the SeeColOr system. First, the haptic device gives us one pixel, not a window. Actually, SeeColOr's configuration computes pixel values over a 17 x 9 pixel window. This is done in order to sonify more colours at the same time, and to spatialize them in 3D using a virtual AmbiSonic approach, so that different sources of colour can be distinguished. Since this part has not been completely finished and validated, we have decided to leave it away.

We then have an adaptation of SeeColOr playing only one instrument at the same time. To achieve this, we decided to

quantize the hue wheel into the six primary colours –red, yellow, green, cyan, blue, and magenta–, and grey, from black to white. Thus, learning the correspondences will be easy for blind users. Indeed, too much instruments would be difficult to learn and recognize, since we only play on the timbre of the instrument. Furthermore, the choice of instruments was reduced, due to different causes.

TABLE 2
CONDITIONS LEADING THE GAME'S COLOUR-TO-INSTRUMENT MAPPING

Condition	Colour	Instrument
$S < 0.05$ or $L < 0.15$ or $L > 0.9$	grey	double bass
$H = [0^\circ; 15^\circ] \cup]325^\circ; 360^\circ]$	red	oboe
$H =]15^\circ; 70^\circ]$	yellow	violin (pizzicato)
$H =]70^\circ; 155^\circ]$	green	flute
$H =]155^\circ; 210^\circ]$	cyan	trumpet
$H =]210^\circ; 255^\circ]$	blue	piano
$H =]255^\circ; 325^\circ]$	magenta	saxophone

- The original database was limited to a certain amount of instruments.
- The recording quality was not good enough for some of the instruments.
- Some instruments were too close to be distinguished from one another.

Table 2 shows the choice we made for the colour to instrument mapping, and the condition for each mapping. The grey colour's condition is prior to any other one, because for some values of saturation and lightness (presented in this table), the hue is not important anymore, the visual feeling remains grey. Note that the values were chosen empirically, as for the choice of instruments. In fact, the colour to instrument mapping was made, imagining the feeling effect of colour and instrument. For example, the green reminds nature, and then, singing birds, which leads to flute. Cyan is a powerful colour, like the sound of the trumpet. In jazz, you can hear about the blue note, and piano is one of the most known instruments in jazz music. Experiments done during the last months showed that this approach should be reconsidered.

With this sonification system defined, we can now talk about the technical integration into the game. The sonifier is an independent server, which is waiting for the haptic device to send the colour to map. This has been implemented in C/C++ through a socket server, which works as follows.

- Launch server
- While not the end of the game do
 - Server listens to client
 - If Client connection then
 - $string \leftarrow buffer$
 - if $string[0] = '1'$ then
 - $R \leftarrow rrr$ part of string
 - $G \leftarrow ggg$ part of string
 - $B \leftarrow bbb$ part of string
 - $hsl \leftarrow rgb2hsl(R,G,B)$

- sound ← sonify(hsl)
- play(sound)

The pixel string send by the haptic device part is normalized according to this code: “n;rrr;ggg;bbb”. The first value, n, is 0 or 1, to say if this value has to be sonified or not. The rrr, ggg, and bbb are the red, green, and blue values respectively, coded on three characters, and are filled with zeros if necessary.

IV. HAPTIC INTERACTION

Haptic rendering is performed at every time step of the haptic loop using the extensively used spring dumper model. The force feedback calculation is performed using directly the GHOST SDK [15], [16] library for PHANToMTM device. PHANToM desktop has 6 DOF for input (provides position and orientation) and 3 DOF for output (provides force feedback along the three axes). In particular, the force fed onto the haptic device is evaluated through the following formula:

$$F = k_s d - k_d v$$

where k_s , k_d are the spring and dumping coefficients and d, v the penetrating distance of the haptic probe into the grooved line map and its velocity.

In order to provide realistic force feedback it is important to ensure that force feedback loop runs at frequency equal or higher than 1KHz. As a result simplified models of the 3D visual objects are used for the collision detection and the calculation of force feedback in the system.

V. SIGN LANGUAGE RECOGNITION AND SYNTHESIS

A. Sign Language Recognition

Sign language recognition used in the application has been developed in cooperation with Project 3 within the eNTerFACE'06 workshop and was integrated to the system.

Figure 3 illustrates the steps in sign recognition. The first step in hand gesture recognition is to detect and track both hands. This is a complex task because the hands may occlude each other and also come in front of other skin colored regions, such as the arms and the face. To make the detection problem easier, we have used differently colored gloves worn on two hands (see Figure 4).

Once the hands are detected, a complete hand gesture recognition system must be able to extract the hand shape, and the hand motion. We have extracted simple hand shape features and combined them with hand motion and position information to obtain a combined feature vector [17].

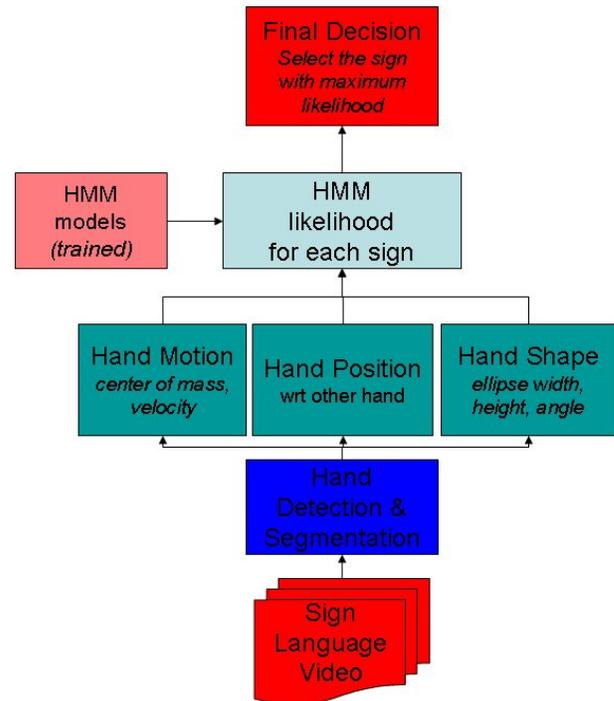


Figure 3. Sign language recognition system block diagram

Our sign database consists of five signs from ASL: *map*, *exit*, *start*, *town*, and *cave*. For each sign, we recorded 15 repetitions from two subjects. The video resolution is 640*480 pixels and the frame rate is 25 fps.

A left-to-right continuous HMM model with no state skips is trained for each sign in the database. For the final decision, likelihoods of HMM for each sign class are calculated and the sign class with the maximum likelihood is selected as the base decision.



Figure 4. The user wears colored gloves

B. Sign Language Synthesis

The system [18] utilizes H-ANIM models to provide the animation and creates the animations using as input Sign Writing Markup Language (SWML). The avatar used was provided by EPFL [19].

Currently, symbols from the 1995 version of the Sign Symbol Sequence (SSS-1995) are supported. This sequence comprises an "alphabet" of the SignWriting notation system, while true images (in gif format) of each symbol contained in this sequence are available in [20].

The input for the sign synthesis system consists of the SWML entries of the sign boxes to be visualized. For each sign box, the associated information corresponding to its symbols is parsed.

The proposed technique first converts all individual symbols found in each sign box to sequences of MPEG-4 Face and Body Animation Parameters. The resulting sequences are used to animate a H-anim-compliant VRML avatar using MPEG-4 SNHC BAP and FAP players, provided by EPFL. The system is able to convert all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box. Manual (hand) gestures and facial animations are currently supported. The proposed technique has significant advantages:

- Allows almost real-time visualization of sign language notation, thus enabling interactive applications,
- Avatars can easily be included in any virtual environment created using VRML, which is useful for a number of envisaged applications, such as TV newscasts, automatic translation systems for the deaf, etc.

1) Generation of BAP key-frames

The shape number field of movement description symbols, which indicates the symbol shape, indicates the type of movement. First, the total number of key-frames to be produced is determined, based on the number and nature of the available movement, movement dynamics, contact, and synchronization symbols. More specifically, a look-up table is used to define an initial number k of key frames for each movement symbol. Furthermore, the fill parameter specifies whether the motion is slow, normal or fast. In addition, some symbols explicitly specify the movement duration. For this reason, a classification of such symbols into three categories has been defined and a different duration value D is defined for each category:

- Slow motion ($D=3$)
- Normal motion ($D=2$)
- Fast motion ($D=1$)

Synchronization (Movement Dynamics) symbols (180,181 and 182) are handled in a similar way as movement symbols. An exception is the "Un-even alternating" symbol, where first one hand moves, while the other hand is still and then the opposite. Thus, in this case, the total number of key frames is doubled ($N=2kDP$), since kDP frames are generated for the first hand, while the second hand remains still then and vice versa.

After the generation of the key-frames related to the available hand, contact, movement and synchronization symbols, it is checked whether more palm postures for the right or both of the hands exist. If there are more than one

palm symbols for one or both hands, additional key-frame(s) are generated containing the values of the BAPs, which represent the final palm position(s).

2) Use of Inverse Kinematics

Our technique is applied only to rotational joints, whose configuration is specified by one or more scalar values, describing the angle values (degrees of freedom) of a rotational joint. The complete configuration of the multibody is specified by n unknown scalars $\theta_1, \dots, \theta_n$ (joint angles) describing the configuration of all joints ([21]). The positions of the k end effectors are denoted by a vector $s=(s_1, \dots, s_k)$. The (desired) target positions are also expressed by a vector $t=(t_1, \dots, t_k)T$, where t_i is the target position for the i th end effector, and $e_i = t_i - s_i$ is the corresponding error. If $\theta = (\theta_1, \dots, \theta_n)$, T is the column vector of the unknown joint angles, the position of the j th end effector is given by a function $s_j(\theta)$, $1 \leq j \leq k$ of the joint angles. In vector notation, this can be expressed as $s=s(\theta)$, where $s_i = s_i(\theta)$. According to the IK problem we seek values for the θ_j 's such that $t_i = s_i(\theta)$, for all i . These equations can be solved by iterative local search based on the $m \times n$ Jacobean matrix J whose elements are defined by: $J_{i,j} = \frac{\partial s_i}{\partial \theta_j}$. According to that iterative method,

the current values of θ , s and t are used for the computation of a value $\Delta\theta$ and the incrementing of the joint angles θ by $\Delta\theta$. Since $s' = J(\theta)\theta'$, the resulting change in end effector positions can be estimated as $\Delta s \approx J\Delta\theta$. The angle update may be performed either once per frame so that the end effectors only approximately follow the target positions, or iteratively until the end effectors are sufficiently close to the targets.

The entries in the Jacobean matrix are usually easy to calculate. If p_j is the position of the joint, v_j is a unit vector pointing along the current axis of rotation for the joint, and the i th end effector is affected by the joint, then the corresponding

entry in the Jacobean is $\frac{\partial s_i}{\partial \theta_j} = v_j \times (s_i - p_j)$, where angles are

measured in radians with the direction of rotation given by the right rule. If the i th end effector is not affected by the j th joint,

then $\frac{\partial s_i}{\partial \theta_j} = 0$. The update value $\Delta\theta$ is computed using the

Selectively Damped Least Squares (SDLS) method ([21]), where the damping constants depend not only on the current configuration of the articulated multibody, but also on the relative positions of the end effector and the target position as well as on the difficulty of reaching the target rather than just the distance to the target.

The generation of the FAP frame sequence is performed after the generation of the BAP frame sequence, so that the total number of generated FAP frames is exactly the same as the total number of BAP frames. For each sign-box, the FAP

key-frames are determined, based on the existing facial expression/animation symbols, from predefined lookup tables for each symbol. The number of FAP key-frames, $N_{FAP_keyframes}$, is generally much smaller than the total number of BAP frames N_{BAP} that have been already generated using the procedures described in the previous Subsections. If $FAP(k), k = 0, \dots, N_{BAP} - 1$ denotes the vector of FAPs corresponding to frame k , the FAP keyframes are first positioned every $s = N_{BAP} / (N_{FAP_keyframes} - 1)$ frames:

$$FAP(i \cdot s) = FAP_keyframe(i), i = 0, \dots, N_{FAP_keyframes} - 1$$

Then, each of the remaining FAP frames is determined using linear interpolation between the two closest available FAP keyframes.

VI. GROOVED LINE MAP GENERATION

In the context of the treasure hunting game, a tool for generating grooved line maps out of interactively sketched 2D drawings is developed. A grooved line map is a 3D terrain that is grooved in specific areas that represent streets or other meaningful areas that the blind user is able to perceive through a haptic device. Recently, a system for converting conventional 2D maps to haptic representations for the blind has been developed by the ITI-CERTH team [22].

The method presented in [22] has been extended so as to convert drawings that are interactively sketched by the user into haptic representations. Figure 5 illustrates the sketched image, while Figure 6 depicts the 3D grooved line map. Since haptic rendering is a very sensitive process and demands in every time step to perform the computationally intensive procedure of collision detection that performs slower for larger 3D meshes, the grooved line map is further processed so as to generate a multiresolutional grooved line map as illustrated in Figure 7. It is obvious that this map is more detailed in the areas close to the path thus reducing the redundant complexity of the initial 3D map.

VII. HEIGHT FIELD BASED PARTIAL MATCHING

Another important aspect of developed framework is its partial matching utility using height fields. In particular, the user is initially drawing with the one hand 3D gestures that correspond to the shape of a 3D object. A stereo camera captures the gestures and generates a 3D point cloud.

The point cloud is subsequently filtered so as to remove the 3D visual tracking noise. To achieve that, a low-pass filter is used, that erases points with high value of deviation after checking their distribution in space.

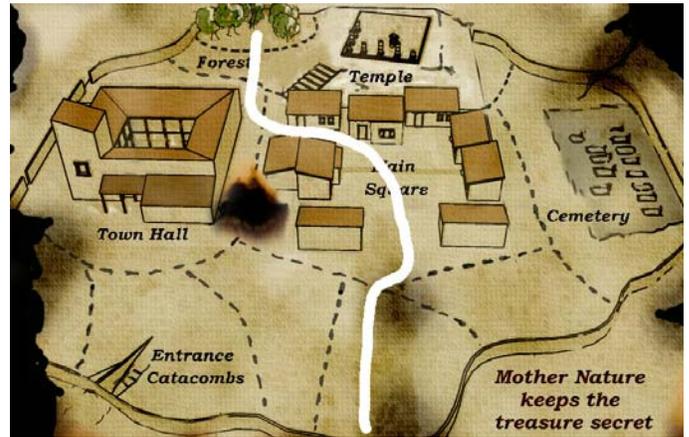


Figure 5. Example of sketched image map

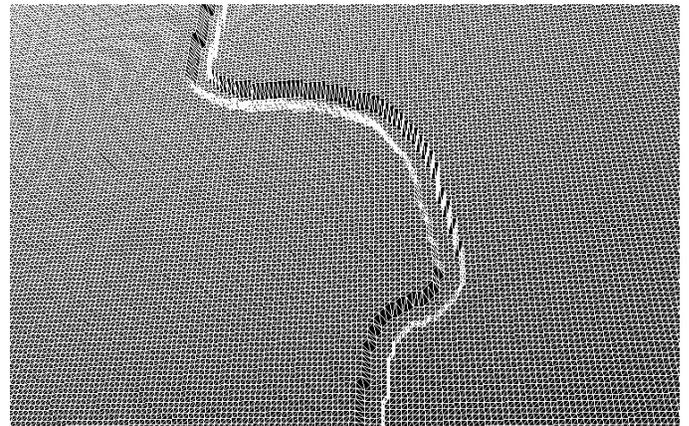


Figure 6. The 3D grooved line map.

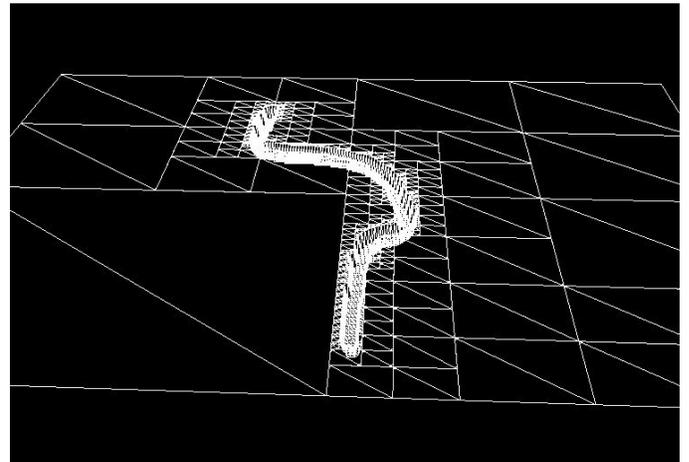


Figure 7. 3D grooved map after polygon reduction

A. Creating Height Fields

Height fields are fast, efficient structures that are generally used to create terrains or other raised surfaces out of hundreds of triangles in a mesh. A height field is, as implied, a 2D array that stores in its entries values about the height of the specific point in the 3D structure [23]. Graphics applications usually create a polygon mesh so as to render a height field [24], [25].

The values of the height field are normalized in the interval $[0,1]$. In the context of the present framework, we initially generate a height field that corresponds to the entire scene. During runtime, the user sketches a 3D point cloud that is also converted to a height field. The aim of the system is to identify in the scene the object that the user has drawn.

The first step is to generate the height field from a triangulated surface. In the case of the sketched point cloud, Delaunay triangulation [26], [27] is used in order to convert the point cloud to a 3D mesh. It is obvious that from a 3D triangulated mesh direct information about the shape's height is obtained only for the vertices of the mesh. Thus, in order to generate a dense height field with height values at every entry of the height map, proper interpolation procedures have to be implemented. Moreover, the generated dense height field is invariant on the sampling resolutions of the scene and the sketched object.

The dense height field is generated by performing interpolation on the triangulated mesh that it models. In the beginning, the user can determine the density of the resulting height field that influences its quality. The height field of the complete 3D scene of the game is shown in Figure 8. Also, the height field for the point cloud is shown in Figure 9. Brighter values correspond to higher areas in the scene.

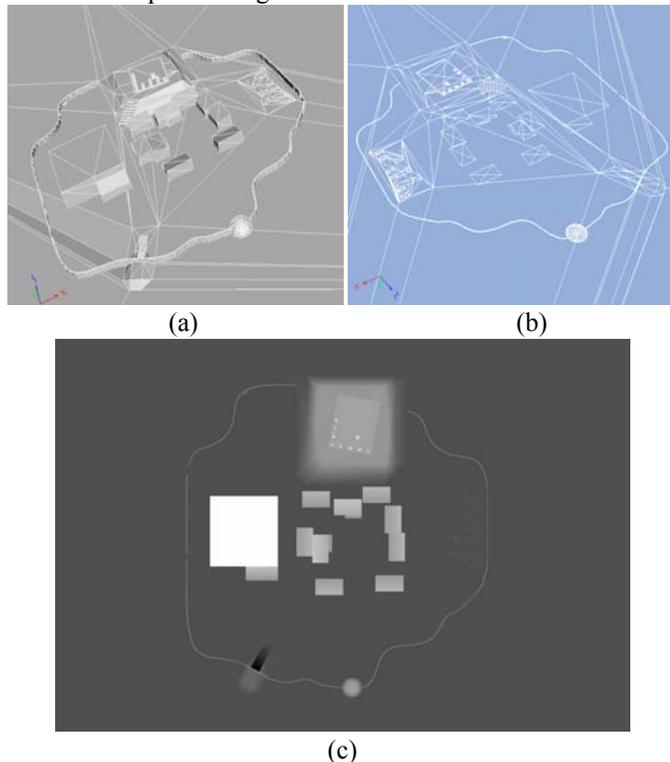


Figure 8: (a) The initial 3D scene, (b) 2D mapping of the 3D scene, (c) The produced height field.

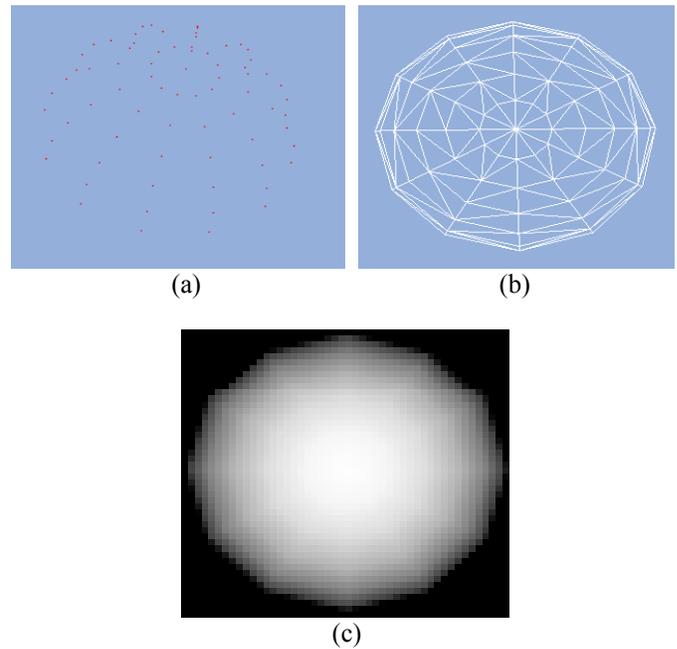


Figure 9: (a) The initial point cloud, (b) Triangulated 2D map of the point cloud, (c) The produced height field.

A height field is in general an efficient way to describe a terrain or a landscape. Moreover, as a 2D structure, it can be processed much faster and state-of-the-art image processing techniques can be applied so as to obtain, e.g. shape descriptors, local features, etc.

On the contrary, it features a serious limitation. In particular, in order to generate accurately a height field out of a 3D mesh, the mapping of the mesh onto the 2D height field surface has to be injective (one-to-one), i.e. each entry of the height field should correspond only to one element of the 3D mesh. This restriction inhibits the modeling of folded and other complex structures with height fields. Although, structures for modeling those kinds of models exist, e.g. vectorial height fields, their processing becomes more complex.

B. Height Field Matching

As mentioned before, the values of the entries of a normalized height field range in the interval $[0,1]$. When trying to match a height field with only one part of another, then a Z-scaling (scaling in the Z dimension) issue obviously arises. In the context of the present framework, this problem is dealt with by renormalizing only the part of the height field that is currently processed. As a result, assuming that the correct 2D part of the height field is chosen, the renormalization would solve a possible scaling mismatch of the source and target height fields.

The matching process proceeds as follows: The source height field is convolved with the target height field by moving the source height field window across the target height field. Notice that the size of the window is variable so as to also consider the 2D scaling mismatch. Moreover, correlation of the input height field and the target height field

part is also considered for different orientations [28], [29] so as to create a rotation invariant matching scheme. The correlation is calculated using the following equation:

$$C(x_0, y_0, a_k, \theta_l) = \sum_{i=x_0}^{x_0+a_k N} \sum_{j=y_0}^{y_0+a_k N} (\mathbf{H}_o(i, j) - \mathbf{H}_s(i-x_0, j-y_0, \theta_l))^2$$

$$\nabla(x_0, y_0),$$

$$\forall a_k \in \{a_i, a_i + a_s, a_i + 2a_s, \dots, a_i\},$$

$$\forall \theta_l, \theta_l = \frac{2\pi k}{N_r} \text{ for } k \in \{0, 1, \dots, N_r - 1\}$$

where (x_0, y_0) refers to the current processing position in the scene height field, a_k is the 2D scaling factor, N is the default size of the sketched height field, a_i and a_t the marginal values of the scaling factor, a_s the scaling space search step, θ_l the current rotation angle and N_r the rotation space resolution. Functions \mathbf{H}_o and \mathbf{H}_s correspond to the scene height field and the sketched object's height field respectively.

Finally, the algorithm outputs a variable vector that maximizes the correlation between the source height field and a target area in the scene height field. The variables in the variable vector refer to X position, Y position, 2D scaling and rotation angle.

VIII. APPLICATION SCENARIO

The aforementioned technologies were integrated in order to create an entertainment scenario. The Scenario consists of seven steps. In each step one of the users has to perform one or more actions in order to pass successfully to the next step.

The storyboard is about an ancient city that is under attack and citizens of the city try finding the designs in order to create high technology war machines.

A. 1st Step



Figure 10. House with the red closet. The white sphere corresponds to the position of the Phantom probe

The first step involves the blind user. The user receives

audio message informing him/her to find a red closed. The user starts from the initiating point at the entrance of the village and using the haptic device explores the village in order to find in one of the houses a red closet. In this step the blind user has to use the haptic device in order to explore the 3D Workspace.

Furthermore audio modality replaces color modality, using SeeColor module, and allows the blind user select the correct closet and thus receive an audio message. The audio message is then sent to the second step of the application.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input*: Audio message “Find the red closet”
- *Action*: Search for the red closet in one of the city houses
- *Output*: Audio message “Town hall”
- *Modality replacement*: Color is converted into sound using the SeeColor utility

B. 2nd Step

The second step involves the deaf and mute user. The user receives audio message. The message is converted from audio to text using the speech recognition tool and then to sign language using the sign synthesis tool. The user finally receives the message as a gesture through a 3D interactive avatar.



Figure 11. The 3D avatar is performing a sign language phrase.

The message guides the blind user to the town hall of the city where the mayor (Figure 12) of the city assigns them a task.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input*: Sign language synthesized phrase “Go to the town hall”
- *Action*: Go to the town hall and talk to the king
- *Output*: Audio message “Go to the temple ruins”
- *Modality replacement*: The input audio message is recognized and converted to text, which is finally

converted to the corresponding sign language phrase.



Figure 12. Town Hall.

C. 3^d Step

The third step involves the blind user, who hears the message said by the Mayor and goes to the temple ruins. In the temple ruins the blind user has to search for an object that has an inscription written on it.



Figure 13. The temple ruins and the inscription.

One of the columns in the destroyed temple has an inscription written on it that states, “The dead will save the city”. The blind user is informed by an audio message whenever he finds this column and the message is sent to the deaf-mute user’s terminal.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input:* Audio message “Go to the temple ruins”
- *Action:* Go to the temple ruins and find oracle inscription
- *Output:* Inscription (text) “The dead will save the city”

D. 4th Step

The fourth step involves again the deaf and mute user. The user receives the written text in sign language form. The text modality is translated to sign language symbols using the sign synthesis tool. Then the deaf and mute user has to understand the meaning of the inscription “The dead will save the city” and go to the cemetery using the mouse.



Figure 14. The cemetery scene.

There he/she should search for a key that lies in one of the graves. The word “Catacombs” is written on the key. The deaf and mute user has to perform a sign in sign language in order to allow the blind user understand that the key opens a box in the catacombs. The deaf user has to perform the sign “Cave”. This sign is recognized by the sign language recognition tool and the text result is sent to the next step of the scenario.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input:* Sign language synthesized phrase “The dead will save the city”
- *Action:* Go to the cemetery and find the key
- *Output:* Key labeled “Catacombs”
- *Modality replacement:* The input inscription text is converted to sign language and the deaf-mute user sketches the word “cave” that is recognized by the system and sent to the blind user’s terminal.

E. 5th Step

In this step the blind user receives the text, which is converted to audio using the text to speech tool. This step involves haptic and audio information. The user has to search for the catacombs enter in them and find the box that contains a map (Figure 15). The map is then sent to the next level.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input:* Audio message “Cave”
- *Action:* Go to the cave (catacombs) and search for a hidden map.
- *Output:* Map

- *Modality replacement*: Text is transformed to synthesized speech.



Figure 15. In the catacombs.

F. 6th Step

The deaf user receives the map, and has to draw the route to the area where the treasure is hidden (Figure 5). The route is drawn on the map and the map is converted to a grooved line map, which is send to for the last level to the blind user.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input*: Map with riddle (Figure 5)
- *Action*: Solve the riddle and sketch the path to the treasure area.
- *Output*: 2D sketch on the map
- *Modality replacement*: Visual information (sketched map) is transformed into haptic representation (grooved line map).

G. 7th Step

The blind user receives the grooved line map and has to find and follow the way to the forest where the treasure is hidden. Although the map is presented again as a 2D image the blind user can feel the 3D grooved map and follow the route to the forest. The 2D image and the 3D map are registered and this allows us to visualize the route that the blind user actually follows on the 2D image. The blind user is asked to press the key of the PHANToM device while he believes that the PHANTOM cursor lies in the path. Finally, after finding the forest he obtains a new grooved line map (Figure 16) where the blind user has to search for the final location of the treasure.

The input-output of this step as well as actions that should be performed are summarized in the following:

- *Input*: Two grooved line maps
- *Action*: Find the forest following the first grooved line map and finally explore the second grooved line map and

- find the treasure.
- *Output*: Treasure



Figure 16. The forest grooved line map.

After searching in the forest streets the blind user should find the treasure that is illustrated in Figure 17.

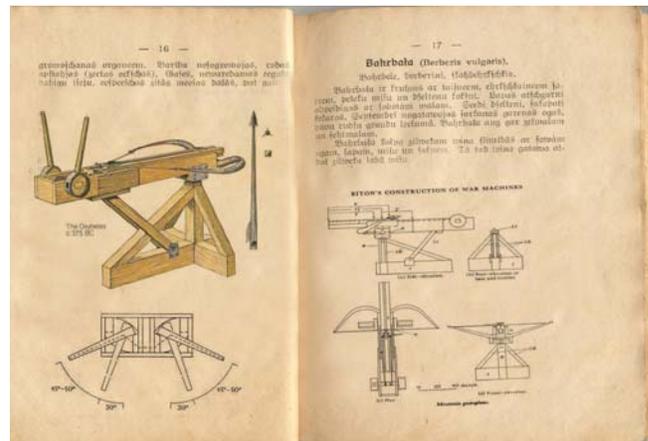


Figure 17. The treasure book.

IX. CONCLUSIONS

The initials tests performed within eNTERFACE have shown that the application is user friendly and is integrated into a feasible for the users scenario. Moreover, the tests the two users are asked to perform have various levels of difficulty.

The proposed system is actually the first attempt of generating a system for the intercommunication of blind and deaf-mute users. Although in the current version the users are somehow limited in their intercommunications, this initial implementation has shown that with proper user-centered design, the development of such a system is feasible.

Further a structured usability evaluation of the system involving both visually and hearing impaired users is a necessary in order to identify the weaknesses of the proposed

methodologies for the intercommunication between the blind and deaf mute users.

ACKNOWLEDGEMENTS:

We would like to thank the group 3 working on '**Sign Language Tutoring Tool**' for providing software, infrastructure and effort to integrate the sign language recognition tool to the developed system. This work was supported by the EU funded **SIMILAR** Network of Excellence.

REFERENCES

- [1] W3C Workshop on Multimodal Interaction, 19/20 July, 2004, Sophia Antipolis, France (<http://www.w3.org/2004/02/mmi-workshop-cfp.html>)
- [2] Special Issue: Interacting with emerging technologies, J. Strickon, Guest Ed., IEEE Computer Graphics and Applications, Jan-Feb 2004.
- [3] I. Marsic, A. Medl and J. Flanagan, "Natural communication with information systems", Proc. of the IEEE, pp. 1354-1366, vol. 88, no.8, August 2000.
- [4] J. Lumsden and S. A. Brewster, "A paradigm shift: Alternative interaction techniques for use with mobile & wearable devices", Proc. 13th Annual IBM Centers for Advanced Studies Conference CASCON'2003, pp. 97-100, Toronto, Canada, 2003.
- [5] T. V. Raman, Multimodal Interaction Design Principles For Multimodal Interaction, CHI 2003, pp. 5-10, Fort Lauderdale, USA, 2003.
- [6] A. Camurri and G. Volpe (Eds.), "Gesture-based Communication in Human-Computer Interaction", Lecture Notes in Artificial Intelligence, no. 2915, Springer Verlag, February 2004.
- [7] C. Colwell, H. Petrie, D. Kornbrot, A. Hardwick, and S. Furner, "Haptic Virtual Reality for Blind Computer Users", in Proc. of Annual ACM Conference on Assistive Technologies (ASSETS '98), pp 92-99, 1998.
- [8] C. Sjostrom, "Touch Access for People With Disabilities", Licentiate Thesis, in CERTEC Lund University, Sweden, 1999.
- [9] V. Scoy, I. Kawai, S. Darrah, F. Rash, "Haptic Display of Mathematical Functions for Teaching Mathematics to Students with Vision Disabilities", Haptic Human-Computer Interaction Workshop, 2000.
- [10] C. Capelle, C. Trullemans, P. Arno, and C. Veraart., "A real-time experimental prototype for enhancement of visionrehabilitation using auditory substitution", IEEE Transactions on Biomedical Engineering, 40(10), 1998.
- [11] G. Iannizzotto, C. Costanzo, P. Lanzafame, and F. La Rosa, "Badge3d for visually impaired", In CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, pages 29-36, 2005.
- [12] Patrick Roth, "Représentation multimodale d'images digitales dans des systèmes informatiques multimédias pour utilisateurs non-voyants", PhD Thesis, Geneva, Switzerland, 2002.
- [13] L. Kay, "A sonar aid to enhance spatial perception of the blind: Engineering design and evaluation." The Radio and Electronic Engineer, 44, pages 605-627, 1974.
- [14] P.B.L. Meijer, "An experimental system for auditory image representations", Transactions on Biomedical Engineering, 39(2), pages 112-121, 1992.
- [15] Sensable Technologies Inc, "PHANToM™ Haptic Device", http://www.sensable.com/products/phantom_ghost/phantom.asp.GHOST
- [16] T. Massie and K. Salisbury, "The PHANToM Haptic Interface: A Device for Probing Virtual Objects", ASME Winter Annual Meeting, DSC-Vol. 55-1, ASME, New York, pp. 295-300, 1994.
- [17] O. Aran, L. Akarun "Recognizing two handed gestures with generative, discriminative and ensemble methods via Fisher kernels", International Workshop on Multimedia Content Representation, Classification and Security, (MRCS'06), Istanbul, September 2006.
- [18] M. Papadogiorgaki, N. Grammalidis, L. Makris, and M. G. Strintzis, "Gesture synthesis from sign language notation using MPEG-4 humanoid animation parameters and inverse kinematics", 2nd International Conference on Intelligent Environments (IE06), July 5-6, 2006, Athens, Greece
- [19] SNHC. ISO/IEC JTC1/SC29/WG11 N2802, <http://coven.lancs.ac.uk/mpeg4/>
- [20] Official Site of SWML, <http://swml.ucpel.tche.br/>
- [21] S.R. Buss (2004), 'Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods', University of California, San Diego, Typeset manuscript, available from <http://math.ucsd.edu/~sbuss/ResearchWeb>.
- [22] K. Moustakas, G. Nikolakis, K. Kostopoulos, D. Tzovaras and M.G. Strintzis, "The Force Field Haptic Rendering Method: Application in the Haptic Access to Visual Data for the Training of the Visually Impaired", IEEE Multimedia Magazine, accepted for publication.
- [23] Fabio Policarpo, Manuel M. Oliveira, "Relief Mapping of Non-Height-Field Surface Details".
- [24] J. Peng, D. Kristjansson and D. Zorin, "Interactive modeling of topologically complex geometric detail". ACM Transaction of Graphics - Proceedings of SIGGRAPH 2004 23, 3 (August), 635–643.
- [25] S. Porumbescu, B. Budge, L. Feng and K. Joy, "Shell maps", ACM Transaction of Graphics - Proceedings of SIGGRAPH2005 24, 3 (July), 626–633, 2005.
- [26] B. Delaunay, Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793-800, 1934.
- [27] P.Cignoni, C.Montani, R.Perego and R.Scopigno. Parallel 3D Delaunay Triangulation. Eurographics 93: 129-142, 1993.
- [28] F. Policarpo, M.M. Oliveira and J. Comba, "Realtime relief mapping on arbitrary polygonal surfaces", In Proceedings of ACM Symposium on Interactive 3D Graphics and Games 2005, ACM Press, 155–162.
- [29] X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, and H.Y. Shum, "Generalized displacement maps", In Eurographics Symposium on Rendering 2004, 227–233.