SPEECH INTERACTION IN VIRTUAL REALITY

Johannes Müller (1), *Christian Krapichler* (2), *Lam Son Nguyen* (1,2), *Karl-Hans Englmeier* (2), *Manfred Lang* (1)

(1) Institute for Human-Machine-Communication, Munich University of Technology,
Arcisstrasse 21, D-80290 Munich, Germany
mue@mmk.e-technik.tu-muenchen.de

(2) Institute of Medical Informatics and Health Services Research, GSF – National Research Center for Environment and Health,
Ingolstaedter Landstrasse 1, D-85764 Neuherberg, Germany
krapichler@gsf.de

ABSTRACT

A system for the visualization of three-dimensional anatomical data, derived from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT), enables the physician to navigate through and interact with the patient's 3D scans in a virtual environment. This paper presents the multimodal human-machine interaction focusing the speech input. For the concerning task, a speech understanding front-end using a special kind of semantic decoder was successfully adopted. Now, the navigation as well as certain parameters and functions can be directly accessed by spoken commands. Using the implemented interaction modalities, speed and efficiency of the diagnosis could be considerably improved.

1 INTRODUCTION

Research on multimodal interfaces is an emerging area of human-machine communication. It has already been shown for a few applications that the integration of speech and hand gestures improves effectiveness and comfort of interaction (e.g. [6], [11]). Most of them concentrate on speech input, with only a few additional gestures [7]. In the presented approach, users can interact with a system for the visualization of three-dimensional anatomical data derived from MRI or CT by choosing from different modalities, e.g. hand gestures, speech, or additional devices. Depending on the properties of the selected input channel, they can navigate through the virtual scene, use virtual tools for object exploration and manipulation, and control visualization and interaction parameters on-line.

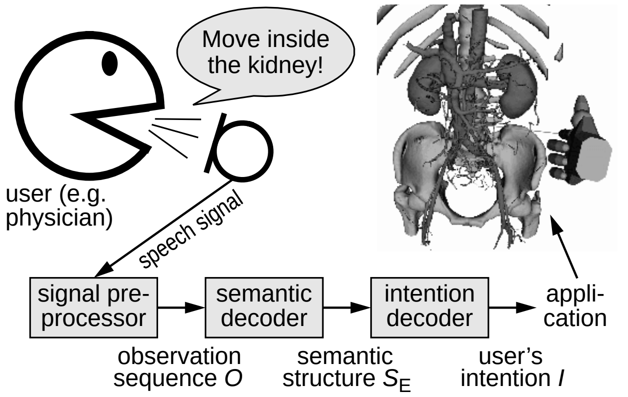

Fig. 1: Architecture of the speech understanding front-end and an example for the speech interaction with the system

2 SPEECH UNDERSTANDING

The speech understanding task is to convert a speech signal into an application-specific code, denoted as user's intention *I*. A signal preprocessing (or 'feature extraction') module creates 64-dimensional observation vectors (or 'feature vectors') in intervals of 10 ms, each of them describing the spectral characteristics of the speech signal contained in a 25 ms-wide window. Similar to current speech recognition systems, the observation sequence *O* is used as input for stochastic pattern matching. As proposed in [14], we use a system architecture with a purely stochastic semantic decoder, as shown in fig. 1.

2.1 Semantic Decoding

The semantic decoder converts a spoken utterance (given as observation sequence *O*) into its semantic representation (denoted as semantic structure *S*). From the set of all possible *S*, that one S_E has to be found which is the most probable given the observation sequence *O*, i.e. which maximizes the a-posteriori probability $P(S|O)$. The resulting term can be transformed using the Bayes formula. Since $P(O)$ is not relevant for maximizing, it can be neglected:

$$S_E = \operatorname*{argmax}_S P(S|O) = \operatorname*{argmax}_S \, [P(O|S) \cdot P(S)] \qquad (1)$$

Due to the high variety of *O* and *S*, it is not possible to estimate $P(O|S)$ directly and additional representation levels are necessary. Clearly defined are the word chain *W* and the phoneme chain *Ph*, which can be used to calculate S_E [12]:

$$\begin{aligned} S_E &= \operatorname*{argmax}_S \max_W \max_{Ph} \left[P(O|Ph) P(Ph|W) P(W|S) P(S) \right] \\ &= \operatorname*{argmax}_S \max_W \max_{Ph} P(O, Ph, W, S) \end{aligned} \qquad (2)$$

Eq. (2) is the maximum a-posteriori (MAP) formula, which is implemented 'top-down' for finding that semantic structure S_E, which is the most likely combination of any semantic structure *S*, any word chain *W*, any phoneme chain *Ph* and the given observation sequence *O*. In the above equations, we assume statistical independence of all probabilities, which are stored in four stochastic knowledge bases (called "models"):

- The semantic model delivers the a-priori probability $P(S)$ for the occurrence of a semantic structure *S*.
- The syntactic model delivers the conditional probability $P(W|S)$ for the occurrence of a word chain *W* given a certain semantic structure *S*.

- The <u>phonetic model</u> delivers the conditional probability $P(Ph|W)$ for the occurrence of a phoneme chain *Ph* given a certain word chain *W*.
- The <u>acoustic model</u> delivers the conditional probability $P(O|Ph)$ for the occurrence of an observation sequence *O* given a certain phoneme chain *Ph*.

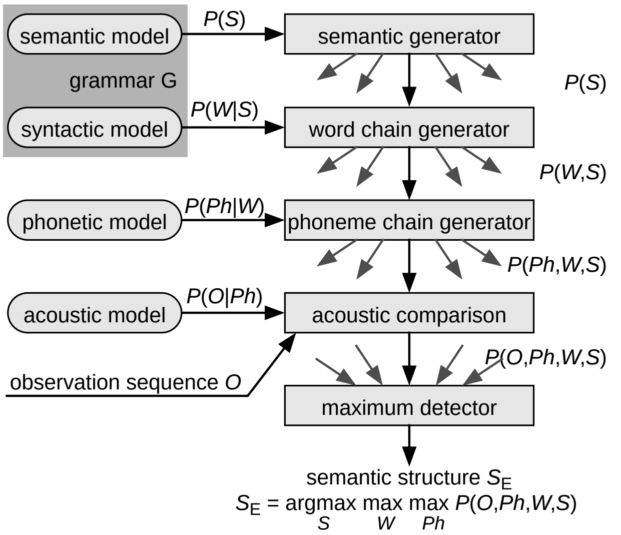

Fig. 2: The ’top-down’ principle for semantic decoding

Phonetic and acoustic models are not necessary to decode written text, since in case of written input (i.e. word chain *W*) eq. (2) can be simplified to

$$S_E = \underset{S}{\operatorname{argmax}} \; [P(W|S) \cdot P(S)] \; . \tag{3}$$

A detailed description of the semantic decoder, which is implemented as an incremental ’top-down’-parser combining a modified Earley-parsing [2] and a Viterbi-beam-search [16] algorithm, as well as the consistent integration of all stochastic knowledge can be found in [13] and in [15].

2.2 Semantic Structure

The *semantic structure S* was introduced as semantic representation of an utterance within a restricted domain [9]. It is hierarchic like a tree, which consists of *n semantic units* (abbreviated *semuns*) s_n :

$$S = \{s_1, s_2, \ldots, s_n, \ldots, s_N\} \tag{4}$$

Each semun s_n can be described by $(X+2)$ components, its type $t[s_n]$, its value $v[s_n]$ and $X \geq 1$ references to its successors $q_1[s_n], \ldots, q_X[s_n] \in \{s_{n+1}, \ldots, s_N, \text{blk}\}$:

$$s_n = \left(t[s_n], v[s_n], q_1[s_n], \ldots, q_X[s_n] \right) \tag{5}$$

- The **type** $t[s_n]$ lays down the number *X* of successors and restricts the set of possible successor-types $t[q_1[s_n]], \ldots, t[q_X[s_n]]$. Furthermore, it makes a selection of the corresponding values $v[s_n]$.
- The **value** $v[s_n]$ shows the exact meaning of s_n.
- Each **successor** $q_x[s_n]$ specifies a certain fact of the semun s_n. If the utterance contains that certain specification, the successor $q_x[s_n]$ is identical with another semun within *S*. In that case, the successor is denoted as *successor semun*

$$q_x[s_n] \in \{s_{n+1}, \ldots, s_N\}. \tag{6}$$

If the utterance does not contain that certain specification, then it is a *blank successor*

$$q_x[s_n] = \text{blk}. \tag{7}$$

For the consistent description of our stochastic approach, it is necessary to allow a type for the blank successor:

$$t[\text{blk}] = \text{blk} \tag{8}$$

The whole semantic structure *S* forms a ’tree’ with the semun s_1 as ’root’ and the blank successors as ’leaves’. All semuns $s_2, \ldots, s_N$ belong to exactly one predecessor semun. Fig. 3 shows an example for a concrete semantic structure S_1 of the described domain in a graphic depiction:

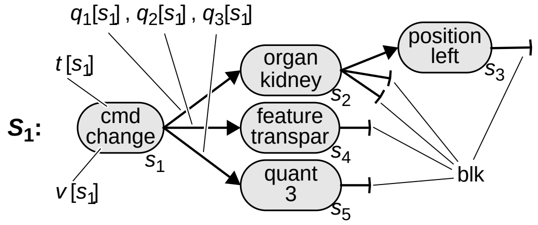

Fig. 3: Semantic structure S_1 corresponding to the utterance *"enlarge the transparency of the left kidney three times"*

2.3 Collection of Training Data

During their multimodal interaction with the visualization system, several subjects (who are specialists for using the system) should clearly speak any commands even if the system would not react on these spoken inputs. In this way, all the subjects are forced to speak within a realistic environment. We thereby recorded 1123 different and authentic spoken utterances in German language by observing eight different subjects.

Due to existing acoustic-phonetic models (trained with spoken German *PhonDat* utterances), only semantic and syntactic parameters are relevant for the calculation the probabilities $P(S)$ and $P(W|S)$. Hence, each utterance was manually converted into the corresponding word chain *W* and the corresponding semantic structure *S*.

In the first initialization step, the probabilities have been estimated by counting the occurring frequencies over the training set. A succeeding iteration step tries to optimize semantic and syntactic ambiguities to improve the probabilities. The iteration is repeated until the maxima of $P(W|S)$ and $P(S)$ are reached [10][15].

2.4 Intention Decoding

Since it is not possible to directly control the application by a semantic structure, the intention decoder has to transform each semantic structure into an application-specific code, denoted as *user's intention I*. The application is controlled by gestures, e.g. generated by hand or, like in our case, by speech. Thus, the command that represents the intention *I* must be a gesture, too. As shown in the next figure, the intention decoder consists of two sub-modules:

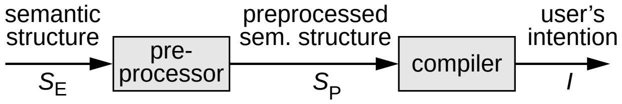

Fig. 4: Block diagram of the intention decoder

The **preprocessor** is necessary to correct inconsistencies in the semantic structure, which occur due to the assumption that each word in the word chain has to be assigned to one single semun [9][10], e.g.

- insertion of missing and necessary information (e.g. the semun of the relevant object),
- deletion of redundant information (e.g. all semuns after an irrelevant "garbage semun").

The task of the **compiler** therefore is to translate a nested semantic structure S_P into a linear code within the application-specific language. Since a main part of a semantic structure's information is held in the topology of the tree, it is not possible to transform each semun individually into one block of the output code. The compiler compounds all pieces of information with the result of a gesture by the use of a-priori knowledge about types, values and the topology of S_P.

As the compiler has to translate ambiguous or incomplete utterances, there are features, which look for the most probable intention or insert missing information. There are two main aspects of context:

- Environment constellation (e.g. *"stop"* can mean *"stop the movement"* or *"stop the running segmentation"*),
- dialogue progression (e.g. the speaker does not repeat information mentioned one or more utterances before).

The contextual situation enormously influences the speaker's behaviour. In certain constellations, it is not necessary to express all information by speech. This aspect is considered by a feedback channel from the application to the compiler. The application is now able to provide relevant information for the compiler. Thus, information can be requested and missing necessary knowledge can be completed.

To gain any important information from dialogue progression, a dialogue history is implemented: Previous utterances are stored in several classes depending on the kind of statement, e.g. "start-command" or "change-command". In this manner, requested information can be returned in a differentiated and effective way.

3 INTEGRATION IN VIRTUAL REALITY

3.1 Visualization

The visual channel is the major output channel for virtual reality (VR). Especially the depth perception is a great benefit of VR displays. To provide a non-flickering representation and to minimize the time delay between user actions and system reactions, a frame rate of about 15 Hz has to be achieved. For the stereoscopic representation, the virtual scene has to be rendered for each eye separately. This means that for the requirements of medicine with very large data sets, even today‘s high performance graphics systems are the bottleneck for image processing and display. But as especially in medical applications no information must be distorted or lost, advanced visualization methods are necessary to fulfil the requirements of speed and accuracy.

There are two different types of data to be rendered: 3D textures (transparent greyscale data from CT or MRI, represented as a stack of 2D slices [1]) and polygonal surface meshes (tools, utilities and segmented medical objects). The ability to apply both volume- and surface-rendering at the same time (hybrid visualization [3][5]) allows to make use of the advantages of both techniques. The user can look at the original tomographic greyscale data at any time. This avoids loss of information due to the previous segmentation and triangulation steps. For purposes like quantitative evaluations, surgical planning or simulation, the surface rendered, segmented objects are necessary.

3.2 Multimodal Interaction

Each interaction device provides the application process with a subset of predefined gestures. In this context, a gesture can be regarded as a user action, which is mapped to application commands. For example, it can be a hand gesture (e.g. making a fist), a speech gesture (e.g. spoken command), or a button gesture (e.g. button click).

Concerning the speech understanding task, this means that the result of the intention decoding process is treated in the same way as a recognized hand gesture from the data glove or a button click of the 3D mouse. Within each frame, all gesture generating sources like hardware devices, speech understanding, hand gesture recognition [8] or virtual menus are polled, and the returned gestures are collected and evaluated. The gesture source makes no difference to its effect on the application. For example, it is possible to accelerate a virtual flight by a spoken utterance, a hand gesture, or a 3D mouse.

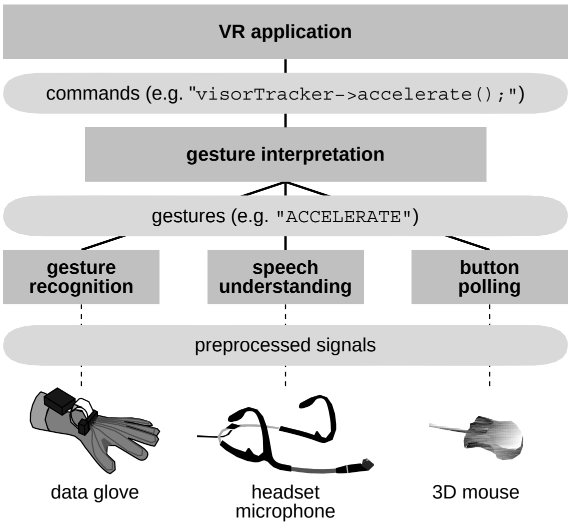

Fig. 5: Various input devices allow multimodal interaction. User actions are translated into uniform gestures.

This multimodal approach allows the user to choose that input channel, which appears to be most appropriate in a given situation. Speech understanding is capable of providing the most extensive subset of defined gestures, covering following areas:

Navigation: The user can start and stop flights through the virtual world, change the flight direction and control velocity and acceleration (e.g. *"slow down"*). Collision detection and a pathfinder utility allowing guided navigation through objects (e.g. in virtual endoscopy) can be switched on and off. Furthermore, it is possible to be placed at a marked position or to mark the current position.

System and application parameters: There is a wide variety of parameters for the adjustment and control of the visualization task. Some system parameters like the desired frame rate or the dynamic video resolution factor of the monitor system can be directly controlled. The display of graphic pipe statistics can be switched on and off for each graphics channel (e.g. *"show me the current frame rate"*). Further functionalities like the visualization of bounding boxes, bounding frames around volume slices, or changing between polygon filling styles (wireframe, filled, scribed) can be chosen. Also, the global accuracy of the polygonal meshes (level of detail) or the 3D textures (voxel resolution, number of interpolated slices) can be influenced. Additionally, some basic application control (e.g. quit, reset, undo, start glove calibration, load/save parameters) can be performed by speech gestures.

Tools and utility functions: There are numerous tools and utilities to explore and manipulate the medical data in the virtual scene. A simple feature is to take snapshots of the scene and save them in a file. Another tool is the virtual probe, a plane to visualize cuts through the tomographic data at any position and in any orientation. This is very helpful to diagnose e.g. the internal structure of a tumour or blood flow in an aneurysm. With spoken commands, it is possible to switch this probe on and off, to adjust its size, and to change its z-buffer representation (permanently in the foreground or being occluded by other objects). Furthermore, segmented objects or data volumes can be switched on and off, and their appearance can be controlled (e.g. size, level of detail, transparency, colour). The ability to switch 3D menus on and off is very helpful, too. The menu structure gives the user further access to parameters and functionalities of the application. For example, sliders allow to control exact values of parameters, buttons can switch tools or values on and off, and displays allow to monitor variables, messages or the user‘s current location in VR.

More sophisticated tools have been developed to support the whole image analysis process, especially the laborious image segmentation [8]. The user can switch to different segmentation modi (automatic, volume growing or model based segmentation [5]) easily by spoken commands like *"I want to use the volume growing segmentation"*, resulting in a gesture "VOLGROW_ON". To reduce data in the 3D texture buffer and thus enable enhancement of the volume data resolution, the tomographic data sets can be reduced to the region of interest using a scalable clipping box, which cuts off all parts of the data volume outside the box. For the volume growing segmentation method [4], it is possible to restrict the area by setting barriers. The algorithm is prohibited to pass those barriers. Their location, orientation, size and shape can be controlled by speech as well as by hand gestures. As next step, a seed voxel can be placed inside the tomogram, marking the start position for the volume growing algorithm. The segmentation can be started by a command like *"start the segmentation now"*.

4 RESULTS

After the training described in chap. 2.3, the semantic and syntactic models contain 26 different types, 112 different values and about 1900 probabilistic parameters. To evaluate the grammar, the semantic structures of the utterances were decoded according to eq. (3). The acoustic-phonetic modelling problem was left aside, so the word chain W was the input of the parser and the factors $P(O|Ph) \cdot P(Ph|W)$ of eq. (2) were omitted. The reclassification (conversion $W \rightarrow S$) of all 1123 training utterances results in 99.8% semantic accuracy (correct semantic structures) and 99.9% semun accuracy (correct semuns). Since we achieved a perfect (rule based) conversion $S \rightarrow I$, the intention decoding accuracy amounts to 100.0%.

REFERENCES

[1] B. Cabral, N. Cam, J. Foran: *Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware*, Proc. ACM/IEEE Symposium, Volume Visualization, 1994, pp. 91-97

[2] J. Earley: *An Efficient Context-Free Parsing Algorithm*, Comm. of the ACM, vol. 13 (1970), no. 2, pp. 94-102

[3] K.H. Englmeier et al.: *Hybrid Rendering of Multidimensional Image Data*, Meth. Inform. Med. 36 (1997), pp. 1-10

[4] R.C. Gonzalez, P. Wintz: *Digital Image Processing*, Addison-Wesley, Reading, 1987

[5] M. Haubner, C. Krapichler, A. Lösch, K.H. Englmeier, W. van Eimeren: *Virtual Reality in Medicine: Computer Graphics and Interaction Techniques*, IEEE Trans. Information Technology in Biomedicine 1(1) (1997), pp. 61-72

[6] A.G. Hauptmann, P. McAvinney: *Gestures with Speech for Graphic Manipulation*, Int. Journal of Man-Machine Studies 38(2) (1993), pp. 231-249

[7] M. Johnston et al.: *Unification-based Multimodal Integration*, Proc. 35th Ann. Meeting Assoc. for Computational Linguistics (Madrid, Spain, 1997)

[8] C. Krapichler, M. Haubner, A. Lösch, K.H. Englmeier: *A Human-Machine Interface for Medical Image Analysis and Visualization in Virtual Environments*, Proc. ICASSP 1997 (Munich, Germany), pp. 2613-2616

[9] J. Müller, H. Stahl: *The Semantic Structure in Comparison with Other Semantic Representations*, Proc. SPECOM 1997 (Cluj-Napoca, Romania), to be published

[10] J. Müller: *Die semantische Gliederung zur Repräsentation des Bedeutungsinhalts innerhalb sprachverstehender Systeme*, Ph.D. thesis, Herbert Utz Verlag, Munich, 1997

[11] S. Oviatt: *Multimodal Interfaces for Dynamic Interactive Maps*, Proc. CHI‘96 (ACM, New York, 1996), pp. 95-102

[12] R. Pieraccini, E. Levin: *Learning how to Understand Language*, Proc. Eurospeech 1993 (Berlin), pp. 1407-1412

[13] H. Stahl, J. Müller, M. Lang: *An Efficient Top-Down Parsing Algorithm for Understanding Speech by Using Stochastic Syntactic and Semantic Models*, Proc. ICASSP 1996 (Atlanta, USA), pp. 397-400

[14] H. Stahl, J. Müller, M. Lang: *Controlling Limited-Domain Applications by Probabilistic Semantic Decoding of Natural Speech*, Proc. ICASSP 1997 (Munich), pp. 1163-1166

[15] H. Stahl: *Konsistente Integration stochastischer Wissensquellen zur semantischen Decodierung gesprochener Äußerungen*, Ph.D. thesis, Herbert Utz Verlag, Munich, 1997

[16] A.J. Viterbi: *Error Bounds for Convolutional Codes and an Asymptotical Optimal Decoding Algorithm*, IEEE Trans. Information Theory, vol. 61 (1973), pp. 268-278