

INTERACTIVE VISUALIZATION AND SONIFICATION FOR MONITORING COMPLEX PROCESSES

Thomas Hermann, Christian Niehus and Helge Ritter

Neuroinformatics Group · Faculty of Technology · Bielefeld University
Bielefeld, Germany
thermann@techfak.uni-bielefeld.de

ABSTRACT

This paper introduces AVDisplay, a versatile auditory and visual display for monitoring, querying and accessing information about modules or processes in complex systems. In the context of a collaborative research effort (SFB360, *artificial communicators*) at Bielefeld University, a cognitive robotics system for human-machine interaction is being developed. The AVDisplay provides the central interface for monitoring and debugging this system, currently involving about 20 computers hosting more than 30 complex processes. The display is designed to provide a summary over the system's activities combining visualization and sonification techniques. The dynamic visualization allows inference of correlated activity of processes. A habituation simulation process automatically sets a perceptual focus on interesting and relevant process activities. The sonification part is designed to integrate emotional aspects – if the system suffers from poor sensory quality, the sound conveys this by sounding uncomfortable.

1. INTRODUCTION

The digital revolution during the last decade has led to an explosion of data obtained from complex systems and to a growth of complexity in technical systems. The massive amount of data available from systems like electronic trading systems, network traffic or medical data is beyond the means of classical information display techniques. In many systems, however, insight can only be gained by a thorough exploratory data analysis. The human brain and perceptual system is extremely suited to process vast amounts of sensory data in a short time and is capable of performing extraordinary pattern detection tasks. Since in complex processes that evolve in time, patterns usually take the form of rhythmical structures, our auditory system is particularly suited to detect such regularities. Besides that, auditory display offers the advantage of being eyes-free, leaving the visual modality free for other tasks, e.g. interaction with a complex system. Related displays were considered for program debugging [1, 3].

In a collaborative research effort at Bielefeld University (SFB 360, situated artificial communicators), a system for multi-modal human-machine interaction for instructing grasping robots is being built [2]. The system combines a robot arm with a hand, a vision system for gestural instruction and visual attention with an intelligent interface for speech recognition and linguistic interpretation. About 30 modules running on more than a dozen computers actively participate to realize the complex system behavior. The modules themselves are often very complex and the distributed system communicates using a multi-threaded distributed architecture communication system (DACS).

A common problem is that failure or inappropriate behavior of individual modules is sometimes noticed too late, which makes it difficult to run and maintain the system. For that reason we developed an audio-visual display (AVDisplay) for monitoring the processes and their inter-process communication. Similar to a “stethoscope” that reflects a patient's state, normal operation of all processes leads to a “normal” auditory pattern of the working system. Any interruptions and failures are easily and early detected from changes in the system diagnosis soundscape. An additional visual display may be accessed if a detailed investigation is wished. It allows to retrieve messages from specific modules and to control an auditory focus on specific modules of interest. We think that our approach, here demonstrated with the SFB robotics system, is quite generic and can easily be extended to be used in other application fields, like monitoring network traffic or debugging multi-threaded code.

The paper is structured as follows: Section 2 briefly presents the multi-modal human-machine instruction system. Section 3 introduces the message protocol. Section 4 illustrates the setup of the AVDisplay, concerning the sonification and visualization parts. In Section 5, different sonification strategies are motivated and contrasted for a typical system interaction and sonification examples are provided. The paper closes with a conclusion and prospects for future work.

2. THE SYSTEM

The SFB 360 aims towards the development of situated artificial communicators, that allow user interaction in a natural “human-like” fashion, including verbal and gestural communication. For the interaction scenario, constructing a toy plane from wooden building blocks (BauFix) was chosen. Limitation of the domain severely facilitates many subtasks, e.g. speaker independent speech recognition due to the limited corpus, visual scene analysis, grasping strategies for the robot. Fig. 1 shows an instructor interacting with the robot system using natural language and pointing gestures.

The high degree of flexibility of the system and its need for robust communication exceeded the possibilities of conventional programming. In fact it prohibits a conventional programming paradigm. Instead, the system architecture is a distributed network of many heterogenous modules (C, Tcl, NEO/Nst, etc.) that exchange information using a common communication platform DACS (Distributed Architecture Communication System), which can be regarded as the “neural pathways” for information exchange.

A coarse overview over the main information processing units is given in Fig. 2. The modules can roughly be divided into four groups, namely robot arm and hand, visual attention, speech pro-

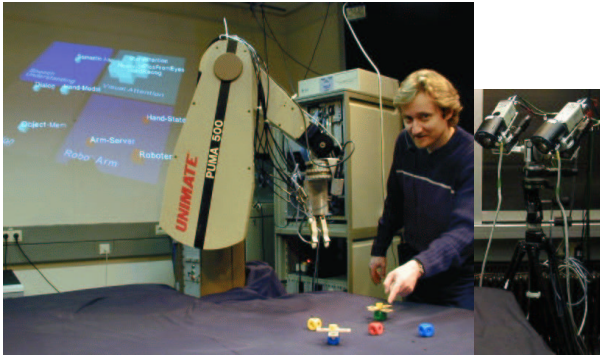


Figure 1: The interaction scenario: A human instructs the robot (hand left, eyes right) how to build a plane from the wooden building blocks. In the background a wall projection of the AVDisplay can be seen.

cessing and integration (see [2] for a detailed description).

3. MESSAGE PROTOCOL STANDARD

The AVDisplay extends the system by a component for merging and interactive monitoring of system-specific messages and module behaviors. It is itself realized equitable to the other modules as a process running on any available machine. In our current implementation, the AVDisplay is a passive system, in the sense that it just collects, filters and displays information about the system and so far has no means to ask certain modules for more verbose information or to control directly the communication between modules. The limitation was made to keep the first prototype library simple. Extension to an active control unit are aimed at with a future version. So far, a C library is provided that allows modules to inform the AVDisplay about their state, actions and results including a qualitative valuation using simple function calls such as AVDSend(). The arguments are used to compile a message string that is sent to the display module via a common DACS stream.

Analysis of typical module behavior led to the development of the following message definition standard, described in Table 1. A typical message string is composed as follows:

```
modulename runtime msg-type [ARGS]
```

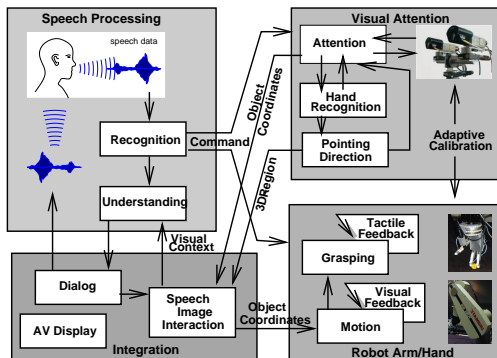


Figure 2: Illustration of the modular system setup.

message type	description
init	process is initialized
exit	a proces is destructed or terminated
start	execution of a service has begun
stop	execution of a service is completed
loop	looped service begins new iteration
action	performed action, e.g. moving camera
result	process likes to inform about a result
state	state machines report new state
info	reserved for additional information
error	process reports an error conditions
display	process orders a display service

Table 1: Message types of the AVDisplay protocol.

The run_time argument specifies the local time measured since module initialization. This allows to sort messages sent by one module. Messages arriving at the AVDisplay module are prefixed with an arrival time stamp allowing to protocol and replay the log file for the system behavior.

Some of the message types specify additional arguments like e.g. "state", where the argument is a name of a state entered by the state machine. The first argument of the "result" message is declared to be a float number between 0 and 1, describing the quality of the result. The meaning of this number depends on the module at hand. For object classifiers, for instance, this will correspond to distance from rejection threshold. If related modules report results of different quality, this can indicate vagueness or inconsistency. If several modules report results of bad quality, this indicates bad general conditions (e.g. lighting, noisy environment, etc.). Thus aspects of a qualitative "emotional" system state can be computed from the incoming result messages. Communication via text strings does not severely reduce the overall efficiency or communication bandwidth - most modules send only about 4 messages per second. Besides readability, the text format is so generic, that it can easily be extended and adapted for monitoring other processes or systems. Our concrete plans are to use the AVDisplay component as a core for debugging and monitoring C++-object creation and modification in our multi-threaded object-oriented sound synthesis engine *rats* [7], that is also used for sound synthesis in this auditory display.

4. DISPLAY DESIGN

The AVDisplay is designed along a model-based framework, leading to audiovisual entities that are able to express both visual patterns and auditory patterns in a coherent way. Thus, modules are identified with concrete 3D-objects in a model space whose configuration and dynamical behavior is related to the incoming messages. Concerning visualization and sonification, the display shows a rendered view on the model space, whereas the sound offers an "auditory aspect" on the modules activities/reactions. According to the groupings observed in our concrete application, the representing modules are clustered in model space as described next.

4.1. Visualization

The visualization used for our application is shown in Fig. 3. Modules are represented as colored objects on a surface. The surface itself is divided into four different plates to respect the grouping of the modules into four different functional divisions, namely a plate

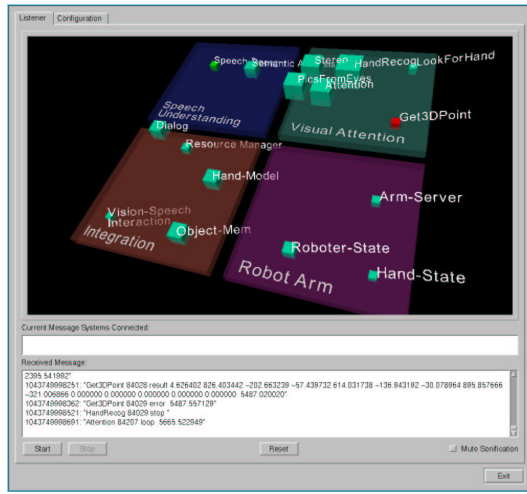


Figure 3: Screenshot of the AVDisplay Visualization. Modules are represented by colored objects on the group plates. Activities of any module is represented by representative object actions causing an auditory and visual change.

for visual attention, integration, speech processing and robotics modules.

For the visualization, many different object attributes are available to a mapping from data. The following selections try to evoke an intuitive association of module activity to activities within the visualization. A module object is instantiated at a free location on its respective plate when an init message is received. Any message of a module that was started before the AVDisplay is launched is treated the same way, so that after a while all modules are represented with 3D objects. The assignment from modules to the plates is achieved by an internal database in the AVDisplay system – unassigned modules are lined up in front of the plates. Color, size, position and orientation have functions within the visual display. Modules perform a blinking action for every arriving message. The blinking color represents the message type – error messages are represented in red signal color. This facilitates associating auditory module representations to objects on the plates. The size of a module represents its frequency in relation to other modules – more active modules are represented as bigger objects. A maximum size is provided to keep the visual display concise. Start messages and loop messages trigger a 180° rotation of the module object around the horizontal axis. This mapping enables to quickly achieve an overview over the active modules. Result messages provide a quality rating, which is used to determine the object color between green (good) and red (bad).

The AVDisplay is able to group modules automatically from an analysis of the pattern of incoming messages. If two modules show highly correlated activity over time, their function is assumed to be related. The correlation of module dependent frequency of messages $f_m(t)$ is used as a similarity measure to determine attractive forces acting on the modules in the 3d-visualization. To avoid module collisions, a pairwise rejection force $F_j^{push}(\vec{r}_i) = -F_i^{push}(\vec{r}_j)$ is added. An additional penalty term is added to avoid disorder. A panel force $\vec{F}_{panel}(\vec{r})$ repulses all module objects from the edges to the center of the plates. This prevents the modules from leaving their plates. Figure 4 illustrates the force

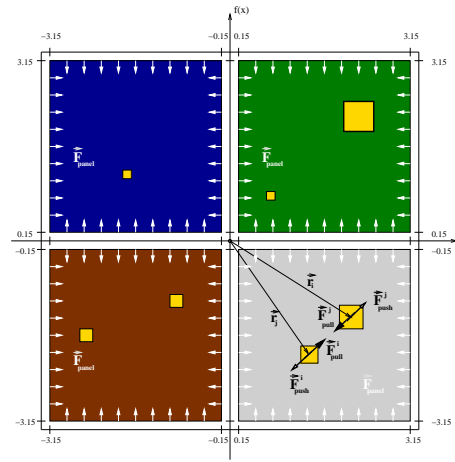


Figure 4: Illustration of force terms used to dynamically update the object positions in the visualization.

terms. The total force on a module is given by

$$\vec{F}(\vec{r}) = \vec{F}_{panel}(\vec{r}_i) + \sum_{j \neq i} (\vec{F}_j^{push}(\vec{r}_i) + \vec{F}_j^{pull}(\vec{r}_i)) \quad (1)$$

Currently, the user can interact with the visualization by using the computer mouse: modules can be muted or focused by clicking the mouse on the objects. Interactive navigation on the display can later be used to influence a sound spatialization using a multi-channel audio system. We further plan to integrate an *aura*, represented by an ear icon in the display. By moving or scaling the aura, a very intuitive means is given for selecting objects to contribute to the sonification.

4.2. Sonification

The sonification is designed according to a model-based paradigm [4], using the proposed identification of acoustic objects in model space as visualized and mapping activities and messages of the modules to interactions and potentially acoustic processes in the auditory display. Where it showed practical, however, the model-based approach is left in order to attain a certain perceptual goal. Three different sonification strategies were followed and two of them will be presented with sound examples in the next section.

4.2.1. Simple Sonification

As a first proof-of-principle, to demonstrate basic operation of the prototypical auditory display, a simple sonification was realized which emphasizes module identification as the most important information of a message by neglecting all other details. A simple decaying sine tone is used to represent the modules, mapping the module ID to pitch. Even this simple sonification suffices to draw the attention rapidly to any failing module, which is perceived as a change of acoustic texture and rhythm. It is practical to divide the pitch space in groups (e.g. octaves) and use pitches within one group for modules on one plate. Sound examples are provided on our web site [5].

4.2.2. Musical Sonification

This sonification aims at a perceptually oriented approach for displaying the available information. Grouping relations in the space of available modules are intended to be perceived as related sounds in the sonification. As a second objective, this sonification aims at integrating all available information about the modules (like message type, density, run-time, results, state) by sound. To limit perceptual irritations, a domain was searched where human listeners show a high degree of sensitivity in discriminately perceiving sounds. For that reason, musical instrument sounds were considered.

Practically, each module is identified with an musical instrument (timbre). The grouping is provided by using four different instrument classes, namely (a) plucked string instruments, (b) hammered instruments, (c) wind instruments and (d) bowed string instruments. Each instrument plays a diatonic interval with the tonic chosen from a pentatonic scale such that all sounds merge to a pleasant harmony. For 'result'-messages, the interval is chosen such that its quality value corresponds to consonance/dissonance. Thus vagueness is heard by unpleasant musical elements. Frequently occurring messages, and those occurring with a high temporal regularity are likely to dominate the sonification and can mask the other modules' contributions. For that reason, we adopted a level mapping based on the "interestingness" of an event. The interestingness is computed from the periodicity of its rhythmical structure and of its absolute frequency [6]. It is used to determine sound level so that interesting events automatically stand out.

4.2.3. Model-Based Sonification

The model-based approach associates material properties with the objects and a material with the different plates (e.g. glass, metal, wood, plastic). Messages are used to excite the objects, resulting in acoustic feedback. Material properties like damping, stiffness, resonances are mapped from variables of the data domain. The resulting sounds merge the same way as individual raindrop sounds merge to the soundscape of raining, allowing to perceive different operation modes operation in different acoustic textures. The implementation is work in progress and sound examples as well as theoretical results can be found in [8] and on the web site [5].

5. SOUND EXAMPLES

Some example interactions are used to illustrate the different sonifications. The "blue cube" interaction refers to an interaction where the instructor asks the system to take a cube. The visual system recognizes several cubes and initiates the dialog module to ask what cube is meant. The answer "take the blue cube" is processed and the robot starts to move and the hand grasps the blue cube. The "red cube" interaction is another similar interaction. Two further examples are being used where either a module fails or the results are of bad quality due to bad lighting conditions. This can be clearly perceived from the sonifications. The sound examples are provided on our web site [5]. The sonifications are used by "non-sonification people" in the robotics lab. Their subjective evaluation is summarized in Table 2, and gives us a clear indication that the musical sonification is better suited than the simple sonification, and that the additional information is accepted.

aspect		pleasantness	reaction time	concentration requirement	risk of distraction	information port selectivity	customization time	risk of information confusion	usefulness as monitoring tool	pitch range	sound quality	sound duration	error detection	
2.2	1.6	2.4	2.8	3.2	2.6	2.6	3	2.8	1.2	2	1.2	3.8	mean	Simple Sonification
0.8	0.6	0.5	0.4	0.4	0.5	0.7	0.6	0.4	0.4	0.9	0.4	0.4	standard deviation	
2.2	1.4	1.6	3	2.2	3.2	1.9	2.4	1.8	1.6	2	1.6	2.4	mean	Musical Sonification
0.8	0.6	0.5	0.6	1.0	0.8	0.9	0.5	0.4	0.8	0.6	0.8	1.0	standard deviation	

Table 2: Subjective evaluation of two different sonification strategies averaged over 5 subjects. The most obvious changes (marked grey) indicate superiority of the musical sonification (1=good, 4=bad).

6. CONCLUSIONS

We presented the AVDisplay, a system for monitoring processes both by visualization and sonification. The auditory display was designed to respect the users' perceptual skills. As an interesting approach, "emotional" information was computed and communicated by using sounds with a similar emotional effect. Interestingness of events was used to determine the level so that changes stand out in the auditory ecology. The AVDisplay can easily be adapted to changing contexts and such generalizations are subject of ongoing research.

7. REFERENCES

- [1] J. L. Alty and P. Vickers, "The caitlin auralization system: Hierarchical leitmotif design as a clue to program comprehension," in *Proc. Int. Conf. Auditory Display (ICAD '97)*. ICAD, 1997.
- [2] P. McGuire, J. Fritsch, J. J. Steil, F. Röthling, G. A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter, "Multi-modal human-machine communication for instructing robot grasping tasks," in *Proc. IROS. 2002*, pp. 1082-1089, IEEE.
- [3] D. H. Jameson, "Sonnet: Audio-Enhanced Monitoring and Debugging," in *Auditory Display*, G. Kramer, Ed., Addison-Wesley, 1994.
- [4] T. Hermann, *Sonification for Exploratory Data Analysis*, Ph.D. thesis, Bielefeld University, Bielefeld, Germany, 6 2002.
- [5] T. Hermann, "Sonification for exploratory data analysis - demonstrations and sound examples," <http://www.techfak.uni-bielefeld.de/~thermann/projects/index.html>, 2002.
- [6] T. Hermann, M. H. Hansen, and H. Ritter, "Sonification of Markov Chain Monte Carlo Simulations," in *Proc. of 7th Int. Conf. on Auditory Display, ICAD*, pp. 208-216, 2001.
- [7] J. Krause, "A haptic interface for real-time control of sonification models" M.S. thesis, Bielefeld University, Bielefeld, Germany, 8 2002.
- [8] C. Niehus, "AVDisp++ - Sonifikation und Visualisierung von Prozessdaten" M.S. thesis, Bielefeld University, Bielefeld, Germany, 4 2003.