HAVE'2006 - IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa, Canada 4-5 November 2006

Multi-Modal Interface for a Real-Time CFD Solver

M. Kasakevich [1], P. Boulanger [1], W. F. Bischof [1], and M. Garcia [2]
[1]Department of Computing Science University of Alberta,
Edmonton, Alberta, T6G 2E8, Canada,
[2]Department of Mechanical Engineering, EAFIT University
Medellín, Colombia

Abstract *– Advances in computer processing power and networking over the past few years have brought significant changes to the modeling and simulation of complex phenomena. Problems that formerly could only be tackled in batch mode, with their results visualized afterwards, can now be monitored whilst in progress using graphical means. In certain cases, it is even possible to alter parameters of the computation whilst it is running, depending on what the scientist perceives in the current visual output. This ability to monitor and change parameters of the computational process at any time and from anywhere is called computational steering. Combining this capability with advanced multi-modal tools to explore the data produced by these systems are key to our approach. In this paper, we present an advanced multi-modal interface where sonification and 3D visualization are used in a computational steering environment specialized to solve real-time Computational Fluid Dynamics (CFD) problems. More specifically, this paper describes how sonification of CFD data can be used to augment 3D visualization.*

Keywords *– Visualization, Sonification, CFD, Haptic Interface, HPC*

I. INTRODUCTION

Advances in computer processing power and networking over the past few years have brought a significant change to the modeling and simulation of complex phenomena. Problems that formerly could only be tackled in batch mode, with their results visualized afterwards, can now be monitored whilst in progress using graphical means, and, in certain cases, it is even possible to alter parameters of the computation whilst it is running, depending on what the scientist perceives in the current visual output. This ability to monitor and change parameters of the computational process at any time and from anywhere is called computational steering [1]. Combining this capability with advanced multi-modal tools to explore the data produced by those systems are key to our approach. In this paper, we present such an advanced multi-modal interface where sonification and 3D visualization are used in a computational steering environment specialized to solve real-time Computational Fluid Dynamics (CFD) problems. More specifically, this paper describes how sonification of CFD data can be used to augment 3D visualization. Figure 1 shows a general overview of the real-time CFD processing environment. The present paper is not concerned with how a real-time CFD solver works (see [1]), but instead with how one can convey useful information of CFD data through a combination of visual and sound modalities.

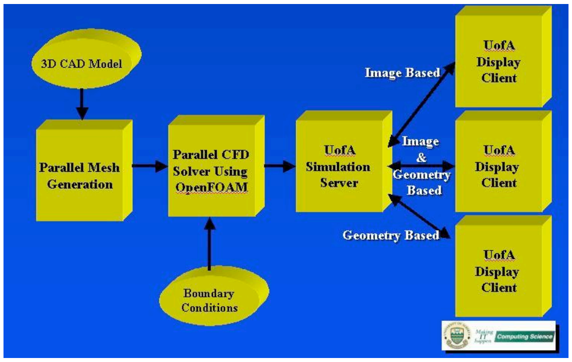


Figure 1: Real-Time CFD Solver Architecture.

Today most computational fluid dynamists exclusively use 3D visualization tools such as ParaView [2], AVS Express [3], and AMIRA [4] for viewing the results of CFD simulations. However, visualization techniques may not be sufficient because the user might miss some important details. As simulation datasets become larger and have larger dimensionality, it is becoming increasingly difficult to visualize them using simple visual encoding schemes such as color, length of arrows, icons, and others. In the scientific visualization literature, this problem is referred to as the dimensionality curse. Scientific sonification comes as a possible solution to this problem because it can be used either as an alternative or as a complement to visualization and possibly other modalities such as haptic rendering. Adding sonification to data exploration increases information bandwidth, thus reinforcing the clues given through other senses. Sonification can help in the recognition of features not obvious from other sources. For example, the visual sense is best at processing spatial information, whereas sound is better for conveying sequential or temporal information [5]. This also gives the user the possibility to concentrate on the dataset using two senses instead of one, creating a multi-modal sensory experience, which is known in the psychophysics literature to improve perception [5,6]. For example, global events could be presented through sound, while local details could be presented through vision. One can also distribute different data attributes to different senses, reducing the effect of the dimensional curse by adding more sensory modalities. For example, the temperature characteristics at each data vertex of a CFD solution could be presented through visual clues, pressure through a haptic stimulation, and flow or vorticity through sound. One of the goals of this paper is to describe the basic ideas of real-time CFD data sonification and its implementation in the context of a virtual wind tunnel [1]. In Section II, we discuss the relevant literature on sonification and in Section III the mapping functions developed for real-time CFD data. In Section IV, we describe the various implementation details, and then conclude by analyzing the results obtained so far.

II. PREVIOUS WORK

Attempts to use sonification as a display channel can be found throughout the literature. One can classify the literature into two basic categories, generic and data-specific algorithms. The category of *generic algorithms* (*category I algorithms*) consists of algorithms implemented as programmable toolboxes that are used to convert generic datasets into sound with predefined mapping functions. These algorithms can sonify all sorts of datasets but the user needs to define how the dataset is mapped to the synthesized sound. The category of *data specific* algorithms (*category II algorithms*) consists of algorithms that are concerned only with sonification in a specific area and are trying to create meaningful sounds (in relation to that area) by mapping data and sounds in a specific way. These algorithms try to take advantage of the data properties specific to the domain. Each of two main categories of sonification algorithms can be further classified by the *sound generation techniques* they use. Some algorithms modify pre-recorded sounds. Others vary physical properties of the sound field: here, sound particles are created for each sound source, then pressure, density and particle velocity are modified. Yet another possibility is to modify the sound wave itself, introducing variations in pitch, duration, timbre, the envelope, and other sound wave properties. Each category can be further classified by indicating whether the mapping function is done in real time or not.

Walker [7] takes temporal data sets of dimensionality M as an input and creates a linear mapping function that modifies N sound parameters such as pitch, timbre, volume, or pan. The linear mapping function is defined by an M x N matrix, where the weights in the matrix are controlled by user inputs. Because of the linear mapping, each data point is played in linear time fashion, creating a sound that is easy to understand, but that is not always descriptive. In this category I algorithm, creating sounds that are descriptive for vector flow field would require non-trivial tuning of the linear mapping function, making this simple scheme unimportant to the rendering of CFD data.

Kaper [8] introduces various mappings from the properties in the dataset to the sound wave parameters. The sound is created through a summation of simple sine waves called partials. Static and dynamic control parameters are used at the levels of the partials to create a meaningful sound. Even though the idea is fairly straightforward, the parameter mappings are not simple and require heavy offline calculations. In this category I algorithm, producing meaningful sound is not obvious because the mapping function could be complex and non-linear, making it hard to tune the sound synthesis process to be descriptive.

An example of a simple category II algorithm is the work by Noirhomme-Fraiture [9] where scalar 2D and 3D time-dependent graphs are sonified in real-time. The main idea here is to map data values to frequencies, creating an easily interpretable sound. In this algorithm, the curves are pre-smoothed before sonification to create a nicer sound.

Metze [10] tries to take advantage of the data properties to produce a sonification algorithm that allows easy distinction between normal and malignant cells images. In this algorithm, a Fast-Fourier Transform (FFT) is first applied to the microscopic cell images. Then amplitude and frequency of the sound are modulated for each pixel by the FFT amplitude and its frequency modulus in the x and y directions. Using this algorithm, every pixel in the FFT image represents a sound with defined frequency and amplitude. In addition to FFT sonification, the authors introduce a temporal playing rule, where a vector in the 2D frequency domain is moving clockwise, like a hand of a clock, in 30 seconds from the twelve-hour to the six-hour position. The sound mapped at each pixel is played, creating a time sequence defined by the angle between the vertical line and the spatial direction of each pixel. The claim here is that lower frequencies predominate for malignant cells, making it possible to distinguish between two types of cells. Metze provides no indication how well this algorithm performs for cell discrimination.

Another example of a category II sonification algorithm is the work of Ballora on heart rate sonification [11]. Here, fairly complicated mapping functions are defined between data and sound characteristics. Data properties, like inter-beat interval characteristics, are used to control the mapping function. The resulting sound is very complex to interpret but, after some training, can be used for discriminating heart anomalies. The paper claims that, with practice, the sound produced by the system become easier to discriminate between normal and abnormal heart rates. Although the paper does not provide supporting evidence for this claim, it illustrates the possibility of interactive, real-time sonification of data.

Bovermann's work [12] also uses a very complicated mapping function, which is defined by a physical model of heat, where local heat causes interaction between data items in space, virtually causing them to create short sound grains. The mapping to sound grain characteristics is quite complicated and the final sound is not always descriptive.

Most of the work described here play with modifying a simple or complex sound wave. A different type of sonification algorithm is presented in Shin's work [13], where a sound field is created at each sound source and the physical properties of the sound field itself are modified. The data is represented as several sound sources, which might or might not be reaching the user in different parts of his/her travel path through the dataset. Sonification is not performed in real time, and the mapping function is quite complicated. One advantage of this algorithm is that it produces a sound that is very clear and self-explanatory.

A bit closer to our CFD sonification problem is Child's work [14]. The main idea behind this sonification algorithm is that one needs to be able to listen to a CFD solver progress and to identify if it is converging or if the solver needs to be stopped and re-started with different parameters. The basic idea of this sonification algorithm is that if the produced sound is converging towards a known sound then so is the solution. The mapping used in this example is fairly simple and based on an algorithm that modifies sound frequency and envelope.

The most interesting work and the on closest to ours is described in Klein [11]. A sphere, representing a user's head is interactively moved around the data field, and only the data samples inside that sphere affect the sound wave reaching a virtual microphone. In this system, sonification is performed interactively and in real-time. Each selected vector's direction and magnitude is mapped to the sound location, level and pitch. If all of the samples in the area are of roughly the same magnitude and direction, then a constant and smooth sound is synthesized, indicating low vorticity in the area. If flow vectors vary widely, sound appears to shift more, giving the impression of higher turbulence. The paper also indicates that the sound produced by this algorithm is very helpful and easy to understand but again without any proof or perceptual evaluation.

In general, one of the rules of data sonification is that careful analysis of specific data properties as well as of the purpose of the required sonification is very important for creating a clear sound that is easy to interpret. Some interesting ideas from the literature can be used. Basic physical ideas, like the fact that the apparent magnitude of a sound varies with the inverse square of the distance, are obvious and simple. Another good idea is Klein's idea of a virtual microphone that can be positioned in the flow field, recording sound sources defined inside a sphere of influence. One of the problems that Klein mentions in his paper is how to choose the right sphere diameter in order to preserve intelligibility. To explore this idea, one could give the user control over the sphere diameter, which could go from a single point to the whole field. A usability study can then be performed to determine the optimal radius. In many ways, Klein's [15] work is very close to the current project, with a number of differences. The field he is analyzing is defined on a rectilinear grid of vectors thus simplifying the connections and relative locations between data points. Choosing a more complicated field grid without such a nice structure introduces a more complicated relationship between user position and the selected data region. Further, Klein is only concerned with finding a good representation of the data in the selected region. In his system, the sphere of influence of the virtual microphone cannot be modified, making the system difficult to test in perceptual studies.

The proposed sonification interface described here belongs to the real-time, data-specific sonification categories, which modifies sound wave properties such as frequency, amplitude, and timing based on CFD parameters. In the next sections, we define a mapping function between the CFD data fields and a sound wave to create an easy-to-understand set of audio stimuli that can be used by fluid dynamists.

III. SOUND MAPPING FUNCTIONS FOR CFD

In this section, we discuss possible mapping ideas for CFD data. One of the problems of sound mapping is to determine at which scale one needs to operate. We can have an ambient, global sound associated with the whole CFD domain or a local sound specific to a particular point, area or line of interaction.

For a global sound, every node contributes to the sonification process, either equally or dependent on the distance to a virtual microphone located in the 3D domain. The first rendering mode gives a uniform change in the field sound from one time-step to the next. The second mode allows for a more interactive exploration of the dataset, where, by moving the virtual microphone, one can hear getting closer to or further away from a point of interest, with a distinguishable sound for vortices or for regions with laminar flow.

For local sonification, only the field values interpolated at the virtual microphone position are used to synthesize the sound. In CFD, the field value at time-step t at the microphone position $\vec{r}_m = (x_m, y_m, z_m)^T$, is defined by a tensor $\mathbf{p}_m(t) = (\rho(t), p(t), T(t), \vec{v}(t), \vec{\Omega}(t))^T$ where $\rho(t)$ is the fluid density, $p(t)$ the pressure, $T(t)$ the temperature, $\vec{v}(t) = (v_x(t), v_y(t), v_z(t))^T$ the fluid speed, and $\vec{\Omega}(t)$ is the vorticity of the fluid, which is proportional to the rotational speed. In this scheme, the field tensor values $\mathbf{p}_m(t)$ are interpolated for sonification by first determining which grid cell the virtual microphone is located in and then, using Schaeffer's interpolation scheme, the flow parameters $\mathbf{p}_m(t)$ at that position are computed using the following equation:

$$\mathbf{p}_m(t) = \frac{\sum\limits_{ForallNodes} \mathbf{p}_n(t) / \|\vec{r}_m - \vec{r}_n\|^2}{\sum\limits_{ForallNodes} 1 / \|\vec{r}_m - \vec{r}_n\|^2} \qquad (1)$$

where $\vec{p}_n(t)$ is the flow parameters at the vertex of the cell at the time-step t and $\vec{r}_n$ the vertex locations.

Following, this interpolation process, a white band noise is then shaped and modified in amplitude and frequency to simulate a wind effect. White noise is filtered through a band-pass filter, with constant bandwidth, were the central frequency is linearly mapped to the field velocity modulus $\|\vec{v}(t)\|$ at the given point. The amplitude of the sound is calculated from both velocity value $\|\vec{v}(t)\|$ and the angle α between a virtual pointer and the field velocity vector at a given point by the simple relationship $F_1(\|\vec{v}(t)\|) \times F_2(\alpha)$ This simple mapping makes the sonification synthesis sensitive to the amplitude and orientation of the flow relative to the probe. Here the scaling function F_1 first transforms the modulus of the speed vector by a simple non-linear mapping $\|\vec{v}\|^{5/3}$ based on some psychophysical considerations [18, 19], then the new value of this mapping is normalized to a value in the interval [0,1]. Similarly, the function F_2 scales the angle α using the same non-linear relation and then is normalized to a value in the interval [0.5, 1] to ensure that we hear the simulated wind even if we are not facing it directly.

An interesting extension of this sonification algorithm is to expand the interpolation function to nodes inside an influence radius *R* where the value of the flow field is interpolated using Schaeffer's interpolating function. In this case, nodes of the specific subset area around the virtual microphone contribute to the synthesis of a sound, allowing for a global/local smooth sound. By adding an interactively changeable radius factor to expand or contract the space of interaction, the user could easily change between local and global sonification.

Another way for producing spatial sound is to view the grid nodes in a radius *R* around the virtual microphone position $\vec{r}_m$ as virtual sound generators located at a distance $d = \|\vec{r}_m - \vec{r}_n\|$ from the microphone. The contributions of each virtual speaker are then added using the familiar $1/d^2$ law for sound propagation. As in the interpolation schemes, the radius of influence could be modified to change the extent of the sensitivity of the virtual microphone. One could also use different attenuation laws that amplify certain preferential orientations in the flow field. For any of those sonification schemes, one will have to do a proper perceptual analysis to determine the true efficiency of each mapping function.

IV MULTI-MODAL INTERFACE

In this section, we discuss the implementation details of the proposed multi-modal interface. Figure 2 shows a block diagram of the proposed interface.

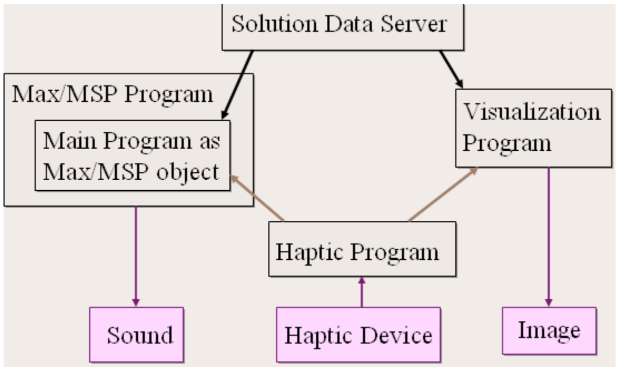


Figure 2: Block diagram of the multi-modal interface.

The system is implemented on a dual Xeon PC under Windows XP operating system using Visual Studio .NET programming environment and Max/MSP graphical programming environment. There are two main and one helper threads that run simultaneously (Figure 2). Both main threads (sonification and visualization) are completely independent of each other and depend on the haptic helper thread to receive position and orientation of the virtual pointer, plus the radius of the interaction space. Since the main purpose of this interface is to sonify CFD data, the sonification thread is only dependent on receiving a pointer position provided by the haptic thread, to manipulate the output sound. The visualization thread is also dependent on the helper thread for visual interaction with the CFD data through the means of a virtual pointer. However, if other threads are not running, it can be run by itself without interaction, for the purpose of viewing the data alone. In this case, the CFD data can be viewed from different viewpoints without interactive sonification.

The data is by the simulation server as a series of time-steps, where the data structure stays the same, but the attribute values at each nodes change. By data structure we mean the coordinates and connections between the data nodes and vertices of a CFD mesh. At a given time-step, visualization of the CFD data is done using OpenGL Performer from geometry produced by the VTK library. The advantage of OpenGL Performer is that, for a given visualization task, the display modality can be easily changed from a computer screen to a stereo cave display for a more immersive effect.

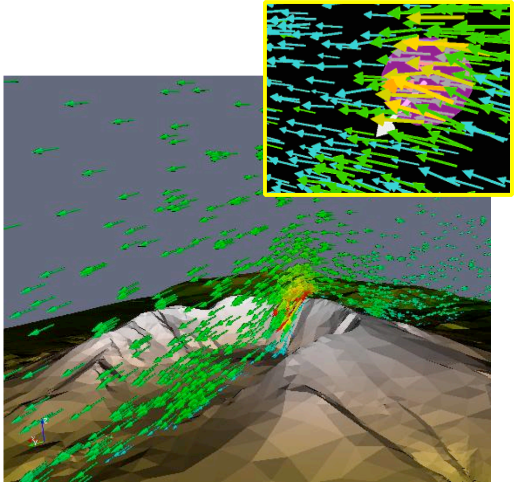
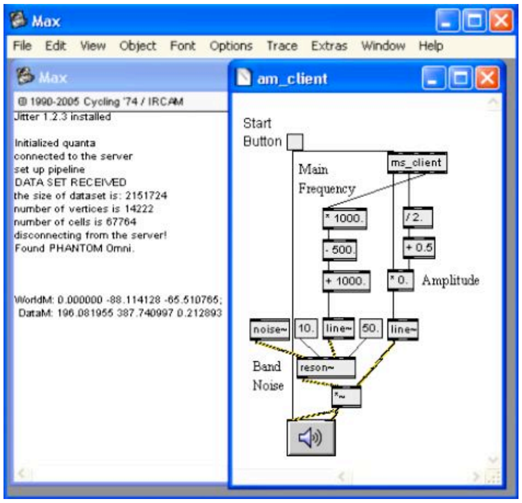


Figure 3: CFD visualization of the airflow over Mount-Saint Helens with a zoomed region showing the 3D pointer.

Because sonification is the main emphasis of this project, only a basic visualization interface was developed for inclusion with the sonification thread. Visualization consists of displaying arrows at each node of the fluid field and a representation of the virtual pointer and the (influence) interaction space in the field (see Figure 3). The direction, size and color of each arrow correspond to the velocity vector values at that node. A virtual pointer is displayed as a white arrow whose position and direction corresponds to the relative position and direction of the haptic device (see Figure 5). The influence space is represented as a sphere located at the end of the virtual pointer, with a diameter specified by the haptic device. Besides a virtual pointer, the visual fluid field can be rotated to look at it from different viewpoints using the mouse.



```
@ 1990-2005 Cycling '74 / IRCAM
Jitter 1.2.3 installed

Initialized quanta
connected to the server
set up pipeline
DATA SET RECEIVED
the size of dataset is: 2151724
number of vertices is 14222
number of cells is 67764
disconnecting from the server!
Found PHANTOM Omni.


WorldM: 0.000000 -88.114128 -65.510765;
 DataM: 196.081955 387.740997 0.212893
```

Figure 4: Simple program in Max/MSP environment.

The haptic device is used to provide feedback on the position of the virtual pointer within the fluid field data allowing users to navigate in 3D inside the field only. In addition to navigation, and, depending on the field values at the virtual pointer, the haptic device could also be used as a force-feedback interface, producing a force that is proportional to the flow density and its direction. In this configuration, the multi-modal interface could provide haptic clues on the properties of the field, in addition to visual and sound clues. This functionality will not be described in this paper.

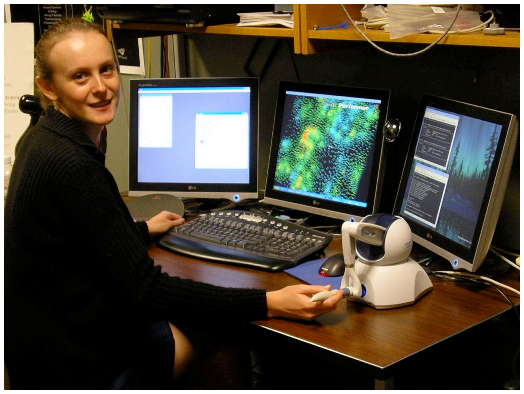


Figure 5: Multi-modal interface for exploring real-time CFD data

Both, main and secondary threads independently connect to the solution server to receive the dataset. Connection is done using Quanta libraries [21], an advanced communication package used for data exchange with the simulation server. The simulation server is in charge of distributing, in real-time, the CFD time-steps computed on a remote high performance computer. After receiving a predefined number of time-steps, both programs disconnect from the simulation server and start the rendering process.

Max/MSP libraries are used to produce a sound for the given dataset. Basically, Max/MSP provides a nice graphical programming environment for various sound manipulations (see Figure 4). A main thread is written as an object for the Max/MSP interface, which can then connect to other Max/MSP objects to produce a sophisticated sonification program synthesizing a meaningful sound.

Because each main thread has a copy of the dataset and directly receives virtual pointer position form the haptic thread, they are completely independent from each other in the data processing or producing pipeline. At a given moment, the sonification thread reads the pointer 3D position and interaction space diameter from the haptic tread and depending on those values, calculates which mesh nodes are the closest, and then interpolates the value of the flow properties that need to be rendered at the pointer position using the algorithm described in Section III. The 3D pointer orientation is also received from the haptic thread and used to calculate the mapping function described in Section III. The current mapping produces a bandpass-noise with a mean frequency ranging between 500Hz and 1500Hz, dependent on velocity value at the pointer, and amplitude dependent on both velocity value and the angle between the velocity vector and the pointer.

V CONCLUSION

In this paper, we present results on a sonification interface for a virtual wind tunnel. So far, we have implemented two algorithms that create very promising results for CFD sonification. The next step in our research will be to do a proper usability study that will help us choose which scheme is best for a virtual wind tunnel interface. The current implementation allows us to explore multiple sonification algorithms using MAX/MSP as a fast prototyping environment for exploring various mapping functions. This is key because in a virtual wind tunnel applications different flow field may require very different sonification functions or parameters.

We are also planning to explore the use of multiple speakers around the user head to provide information on the sound directions. As demonstrated by other researchers, this will give the user a better feeling of immersion into the simulation, giving him/her a more natural representation of the fluid field direction.

We plan to explore other fluid-field sound renderings with this versatile architecture, including sonification along path lines, streak lines, streamlines, and stream tubes.

REFERENCES

[1] P. Boulanger, M. Garcia, C. Badke, and J. Ryan. An Advanced Collaborative Infrastructure for the Real-Time Computational Steering of Large CFD Simulations. European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006 TU Delft, The Netherlands, September 2006.
[2] http://www.paraview.org
[3] http://www.avs.com
[4] http://www.amiravis.com
[5] R. B. Welch and D. H. Warren, Intersensory interactions. In K. R. Boff, L. Kaufman and J. P. Thomas (Eds.) Handbook of Perception and Human Performance: Vol 1. Sensory Processes and Perception, New Cork: Wiley, 1986, pp. 25-1 - 25-36
[6] C. Spence and J. Driver (Eds.) Understanding intersensory integration. Oxford: Oxford University Press, 2004, pp. 377-378
[7] N. Walker, J.T. Cothran, Sonification Sandbox: a Graphical Toolkit For Auditory Graphs, Proceedings of the 9th International Conference on Auditory Display, Boston, MA, July 2003, pp.161-163
[8] H.G. Kaper, S. Tipei, E. Wiebel, Data Sonification and Sound Visualization, Computing in Science and Engineering, Vol. 1 No. 4, July-August 1999, pp. 48-58
[9] M. Noirhomme-Fraiture, O. Schöller, C. Demoulin, S. Simoff, Sonification of time dependent data, Proceedings of International Workshop on Visual Data Mining, Helsinki, Finland, August 2002, pp. 113-125
[10] K. Metze, R.L. Adam, N.J. Leite, Cell Music: The Sonification of Digitalized Fast-Fourier Transformed Microscopic Image, Proceedings of XXIII Congress of the Brazilian Society of Computation, Campinas - SP, Brazil, 2003, pp. 119-124
[11] M. Ballora, B. Pennycook, P.C. Ivanov, L.Glass, A.L. Goldberger, Heart Rate Sonification: A New Approach to Medical Diagnosis, LEONARDO, Vol. 37, No. 1, 2004, pp. 41–46
[12] T. Bovermann, T. Hermann, H. Ritter, The Local Heat Exploration Model for Interactive Sonification, Proceedings of International Conference on Auditory Display, Limerick, Ireland, July 2005, pp. 85-91
[13] Y. Shin, C. Bajaj, Auralization I: Vortex Sound Synthesis, Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization, 2004,
[14] E. Childs, The Sonification of Numerical Fluid Flow Simulations, Proceedings of the 2001 International Conference on Auditory Display, Espoo, Finland, July 29-August 1 2001, pp 44-49
[15] E. Klein, O. G. Staadt, Sonification of Three-Dimensional Vector Fields, Proceedings of the SCS High Performance Computing Symposium, 2004, pp 8 [16] G. Kramer, B. Walker, T. Bonebright, P. Cook, J. Flowers, N. Miner, J. Neuhoff, R. Bargar, S. Barrass, J. Berger, G. Evreinov, W.T. Fitch, M. Gröhn, S. Handel, H. Kaper, H. Levkowitz, S. Lodha, B. Shinn-Cunningham, M. Simoni, S. Tipei, Sonification Report: Status of the Field and Research Agenda, http://www.icad.org/websiteV2.0/References/nsf.html
[17] C. Wassgren, C. M. Krousgrill, P. Carmody, Development of Java applets for interactive demonstration of fundamental concepts in mechanical engineering courses, http://widget.ecn.purdue.edu/~meapplet/java/flowvis/Index.html
[18] W.A. Yost, Fundamentals of Hearing: An Introduction, Forth Edition, 2000, New York: Academic Press.
[19] S.A. Gelfand, Hearing: An Introduction to Psychological and Physiological Acoustics, Forth Edition, Revised and Expanded, 2004, New York: Marcel Dekker.
[20] B. Gold, N. Morgan, Speech and Audio Signal Processing: Processing and Perception of Speech and Music, 2000, New York: Wiley.
[21] http://www.evl.uic.edu/cavern/quanta/index.html