

# Sonification of Three-Dimensional Vector Fields

Eric Klein and Oliver G. Staadt

Center for Image Processing and Integrated Computing (CIPIC)

Department of Computer Science

University of California, Davis

eklein@ucdavis.edu, staadt@cs.ucdavis.edu

**Keywords:** Sonification, Vector Field, Spatial Audio

## Abstract

We describe and analyze a new technique for sonification of three-dimensional vector fields. This technique allows the user to use commodity hardware and widely available 3D sound interfaces to map vectors in a listener's local neighborhood into smooth wind-like sound (aerodynamic sound). The four types of information provided by this technique are flow direction, flow velocity, flow vorticity and local flow patterns. The result is a system that helps the user achieve an intuitive understanding of the data set by using their auditory sense.

## INTRODUCTION

The concept of using sound in computer simulations to enhance the user's understanding of their environment is not new. In fact, it has been used, for example, in computer games since at least the 1980s. It is however not often used in scientific visualization applications. As the size of scientific data sets is increasing rapidly, it is becoming more and more difficult to explore and analyze these data due to their size and complexity. Often the data sets are so massive that when using purely visual techniques, the results can overload the user's visual system. Employing large-scale high-resolution display environments can help to visualize larger amounts of data, but require significant hardware resources that are typically not available at the scientist's desk. Combining visualization techniques with sonification has the potential of offloading some of the data interpretation to an often underused sense, the auditory sense. An ideal application area to incorporate sound is in 3D vector fields. These vector fields are often very dense volumes of data (often millions of voxels), sometimes with many time steps, and using current techniques for vector field visualization such as particles, streamlines, streamers, streamtubes [1], LIC (Line Integral Convolution) [2] and other texture based methods [3], and feature extraction techniques [4, 5], it is often difficult to understand the flow pattern in such a system. Another factor that makes vector fields prime candidates for sonification is that they typically represent some kind of flow, whether it is a fluid, a gas, or a plasma. The idea of a flow combines very naturally with the concept of wind, and most people have an intuitive sense of what wind should sound like for a flow (such as air flow in a wind tunnel).

The system we describe uses 3D spatial audio techniques to position sound sources at distinct positions in relation to the listener's head. The position of these sound sources, combined with the motion of these sounds over time is used

to convey flow direction, velocity, and vorticity. We achieve these results using commodity hardware and software, and our system is therefore usable on standard desktop computers. To our knowledge, this is the first system to allow the user to understand flow direction, velocity, turbulence and local flow patterns in a vector field by using sonification in an intuitive fashion tightly integrated with visualization. The new contributions of this work are the addition of directional information, and using the change in wind direction over time to convey vorticity information.

## PRIOR WORK

There has been a large amount of work in computer-synthesized sound, and there is a specialized research community devoted to computer-synthesized sound [6].

### Spatial Sound

Our approach makes extensive use of 3D sound spatialization [7, 8]. Sound spatialization techniques allow a sound to be played back in such a way that a listener perceives the sound to be coming from a specific location in space. The listener can therefore determine if a sound is coming from the left or right, up or down, and from ahead or behind. In many cases, spatial audio techniques allow the user to determine a sound's location with considerable accuracy. This can be accomplished by using either multiple speakers (e.g., a 5.1 surround sound system) or a head related transfer function (HRTF) [9] and a set of headphones. HRTFs are an active area of research, and many groups are working on developing and improving HRTFs [10].

HRTFs have developed to such a point that it is often possible using headphones to accurately identify the location of a virtual sound source. When head tracking is added and the users are able to move their head to help locate the position of the sound source, these systems become very powerful.

Spatial audio is useful in any situation where the user's innate auditory capabilities can be used to process data. It has been used in a wide range of applications, from game technology, to supplying immersive sound in virtual environments [11]. For our technique, spatial audio is a basic building block, and provides the critical link to the listener's auditory system.

### Sonification for Scientific Applications

There has been very interesting research into the use of sound for exploring scientific datasets. Some of this work has focused on applying sound to specific applications such as understanding data from computational fluid dynamics

(CFD) simulations [12] and exploring data from ocean buoy sensor data [13]. Other research [14, 15, 16, 17] has focused on general techniques and ideas that are applicable to exploring scientific datasets in general. The techniques referenced here are generally concerned with how to map various points in a multi-dimensional dataset into sound. It should also be noted that in general these techniques usually tend to focus on a single sound channel, rather than using stereo sound or full spatial audio. We will only explore work directly related to our technique, but we refer the reader to the International Community for Auditory Display [6] for further information.

## Vector Field Sonification

There are two distinct approaches that have been successfully employed for sonification of vector field data. Eckel [18] has used the *CyberStage* for exploring vector fields. The technique that was employed used streamlines to visualize the flow of air from a car’s air conditioning system. Additionally they used sound to represent the velocity of particles as they moved along the streamlines. The faster a particle moved, the louder the sound would be, and also the higher the sound frequency would be. Additionally, they also used a position tracked probe within the immersive environment of the *CyberStage* to allow the user to sample the velocity, and its corresponding sound, at specific locations in space.

A second approach has been introduced by Volpe and Glinert [19]. Here, a vector field is visualized using streamtubes. A key problem of using streamtubes is that it is often difficult to see some of the twisting of the tubes due to self-occlusion or other visual elements in a complex dataset. This twisting of a streamtube represents vorticity in the region of the twist, and [19] developed a scheme to represent this changing vorticity with sound. The user of their system could move a slider along a streamtube, and as the tube twisted, the sound being played would be smoothly and continuously changed. More specifically, they developed a new technique to create a sound that can have an apparent infinite increase, or decrease, in pitch over time. This technique was then used to create the changing sound as the slider moved along the streamtube. As the streamtube twisted in one direction, an increasing pitch would be heard, and as the streamtube twisted in the opposite direction, a decreasing pitch would be heard. Using this technique it is possible, even in a visually complex environment, to understand the changing vorticity along the length of a streamtube.

## SONIFICATION OF VECTOR FIELDS

In our technique, points in the local region surrounding the listener’s virtual position in the 3D environment are sampled and converted to *wind sounds*. A wind sound generated for a single point is based on the interpolated vector for that point, and it conveys two pieces of information: (1) how fast the data flow is at that point, and (2) in which direction it is flowing. This corresponds to the data provided by a single vector: flow direction, and flow velocity. In addition to the data provided by a single point, our approach also uses data from a region of points to add vorticity data to the sound played. This is accomplished by randomly sampling different points within the listener’s local region several times a second, and smoothly interpolating between the two associated sounds. This approach works well in practice and conveys to the listener vorticity data as well as flow direction and velocity.

Our technique maps the vectors that are sampled to positions surrounding the listener’s head. We then use spatial audio techniques to play sounds from those positions. The resulting effect is of wind flowing to the user from this point in space. As the position of the sound changes in space, the wind also appears to shift direction and intensity.

## Base Sound Samples

We have chosen to use a static sound sample, and the only changes made to this sound sample are the changes managed by the spatial sound algorithm, such as changing timing to different speakers, volume and pitch. We experimented with several different base sound samples, including plain white noise, different variations of colored noise, pre-sampled wind noises, and cabin noise from a Boeing 747. Although some of the pre-sampled wind noises created interestingly haunting sound effects, the sound that introduced the least distortion of the data, and was easiest to interpret was the plain white noise.

## Sampling Vectors

The technique presented in this paper relies upon vector samples from the input data set. The data sets that were used for our experiments comprised rectilinear grids of vectors. Since most samples that need to be taken will not be located at the original grid points, it is necessary to interpolate the actual sample vectors. In our implementation we are using trilinear interpolation for the vectors. We might achieve better accuracy using higher-order interpolation schemes, such as tricubic interpolation, but this is computationally more expensive. Furthermore, we can observe that although spatial sound algorithms are powerful today, they still suffer from limited precision with respect to exact spatial localization. In fact, the error introduced by the use of trilinear instead of tricubic interpolation will be small compared to the error inherent in the human auditory system working with spatial audio algorithms. It is therefore appropriate to use simpler and faster trilinear interpolation. It is worth noting, however, that in order to achieve more accurate interpolation values for vectors, using spherical linear interpolation, rather than simple linear interpolation, is an important requirement.

## Mapping a Vector to Sound

The first step in our technique is to map a single data point to the location of a virtual sound source. To better understand this concept, it helps to think about what a vector represents. The vector information at any given point in space defines in which direction a particle at that position will move. It also tells us how far that particle will move in a single time step. This first implies that the inverse of the vector will define from which direction a particle has come from, and therefore, where the sound associated with that particle will appear to come from. Second, there is a relationship between the magnitude of the vector and the apparent velocity of the wind noise. There are two properties of the sound that we use to convey information about speed: volume and pitch. The higher the volume, and the higher the pitch, the faster the wind appears to be blowing. In other words, the greater the magnitude of the sampled vector, the higher the volume and the pitch should be. With spatial audio systems, the volume of a sound can be controlled by the distance of the sound from the listener. The

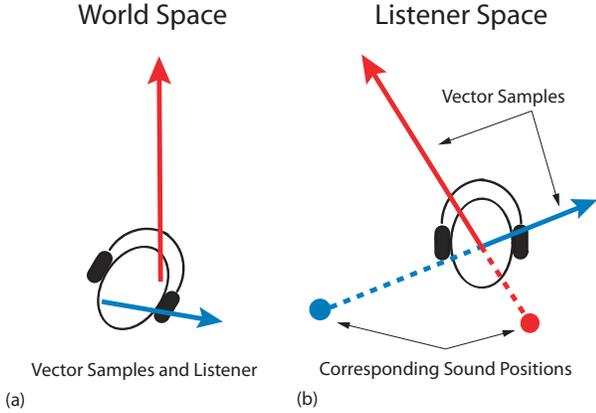


Figure 1: Vectors must first be translated from the world coordinate space (a) to the listener’s coordinate space (b). The sound position is located in the opposite direction from the listener with respect to the direction of the sampled vector. The distance from the sound location to the listener is inversely proportional to the magnitude of the vector, thus causing longer vectors to sound louder. This is illustrated in part (b).

farther away the sound source, the quieter it will be perceived by the listener. The two attributes of direction and distance are illustrated in Figure 1b. Controlling pitch is slightly more complicated, but it can be done by applying Doppler shift effects to the sound source. With currently available spatial sound APIs, it is possible to set a velocity as well as a position for a sound, and this velocity is then used to determine the Doppler shift for the sound. The end result is that by setting a velocity for the sound, we can use the Doppler effect to alter the pitch of the base sound, thus simulating different wind speeds. It should be noted that this is in contrast to Volpe and Glinert [19] who use pitch to represent vorticity changes along a streamtube.

It is important to note that the listener’s head can conceivably be oriented in any direction. Thus, we have to map from the global coordinate frame to the local coordinate frame of the listener when determining the location of a sound source in relation to the listener’s head. This is illustrated in Figure 1b.

With these goals in mind, the following equation for the position  $\mathbf{p}$  of the sound in relation to the listener’s head can be derived as

$$\mathbf{p} = \mathbf{T} \left( -\frac{\mathbf{v}}{|\mathbf{v}|} \frac{1}{\sqrt{|\mathbf{v}|}} \right).$$

In this equation,  $\mathbf{p}$  is the point in relation to the listener’s head,  $\mathbf{v}$  is a vector from the vector field, and  $\mathbf{T}$  is the transformation from the world coordinate frame to the listener’s coordinate frame. The  $\frac{-\mathbf{v}}{|\mathbf{v}|}$  term denotes the normalized vector from the listener in the direction of the sound source (in global coordinates). This follows directly from our requirement that the direction from the listener to the sound source be the inverse of the vector direction. The term involving the inverse square root of the vector magnitude gives us the scaling factor for the distance, and thus the volume. This is valid when we consider that sound often follows a  $\frac{1}{\text{distance}^2}$  attenuation pattern. This again corresponds to the require-

ment that the longer the vector is, the louder the sound should be, and thus the closer it should be to the listener. Lastly, we need to set the velocity of the sound, which is straightforward. The goal here is to have a velocity that points towards the listener from the sound, and to have the magnitude of that velocity increase with the magnitude of the vector. Since the direction from the sound source to the listener is the same as the direction of the actual vector sample, we only have to alter the magnitude of the velocity if necessary. The equation representing this is

$$\mathbf{d} = \mathbf{T}_s(\mathbf{v}),$$

where  $\mathbf{d}$  denotes the Doppler velocity vector and  $\mathbf{T}$  is again the transformation from world coordinates to the local coordinates of the listener. Finally, the function  $s(\mathbf{v})$  preserves the direction of  $\mathbf{v}$ , but also scales  $\mathbf{v}$  based on its magnitude. In our current implementation we use  $s(\mathbf{v}) = \mathbf{v}$ , but it would be straightforward to use others such as  $s(\mathbf{v}) = \frac{\mathbf{v}}{|\mathbf{v}|} |\mathbf{v}|^2$ , which would cause the pitch to increase faster as magnitude increases. Further user studies will help to identify functions that best convey a proper sense of wind velocity.

### Mapping a Region of Vectors to Sound Vorticity

We have described a scheme to map a single vector to a virtual sound source in the previous section. This gives us the ability to convert vectors from the input data set into a sound. In this section we describe how to convey vorticity information. Since vorticity is the measure of change over a region (it can also be thought of as turbulence), this makes it necessary to consider more than just a single point in order to represent vorticity.

Our goal here is to simulate the smooth flowing sounds that wind makes in areas of low vorticity, and to create rapidly shifting sounds for wind in areas of high turbulence. This can be achieved by using not just a single vector sample at a location corresponding to the center of the listener’s head, but rather by transitioning between different samples taken over a volume roughly equal in volume to the user’s head. If all of the samples in this area are roughly the same magnitude and direction, this corresponds to low vorticity. In this case the transitions between sounds will be minor, and thus, the sound will be perceived as rather constant and smooth. Conversely, if the vectors vary widely, then the transitions will be more pronounced, and thus the sound will appear to shift more, giving the impression of higher turbulence. This is also supported by making the speed of the transition from one vector position to the next dependent upon the magnitude of the vectors involved. This will lead to the perception of faster flow shifting more rapidly, as we would expect in reality.

There are three problems that need to be solved to make this scheme work. The first is how to sample the vectors. The second is how often to sample the vectors. The final, and most difficult problem is how to smoothly transition from one sample sound to the next.

One possible approach to sampling the vectors is to define a sphere centered on the listener’s head, as in Figure 2, and take a uniform random sample at each sample time from within that sphere. Our experience shows that this approach is very effective and that more sophisticated techniques will not increase the quality significantly.

The sample rate can be determined by the magnitude of the vectors being sampled. Again, the most straightforward approach is to use each sample vector’s magnitude to determine how long the transition to the next sample will take.

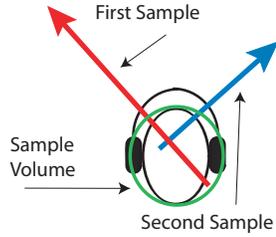


Figure 2: Two consecutive vector samples taken at random locations within the listener's head volume.

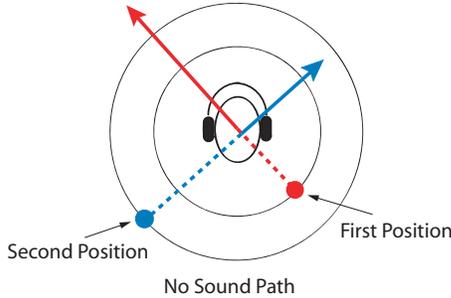


Figure 3: Discontinuous sound path.

This has the advantage of appealing to our intuitive understanding of how wind flows. If, for a given point in space and at a given point in time, a wind vector is relatively strong, we intuitively feel that the wind sound should shift relatively fast, and vice versa. The challenge is to determine exactly how fast this transition should be. The approach that we have chosen is to define minimum and maximum transition speeds, which are 1 Hz and 10 Hz, respectively. We then determine the smallest and largest vector magnitudes for the data set, and define the smallest magnitude vector to have a transition time of one second, and the largest magnitude vector to have a transition time of one tenth of a second. The transition times in-between are calculated by linear interpolation.

There are several possible approaches that can be followed to handle the transition from one vector sample to the next, but many of them do not behave in an intuitive fashion. A straightforward approach is to move the sound from one position to the next abruptly. However, this has the highly undesirable effect of producing sharp jumps in the resulting sound, which can disorient the listener. In other words, this solution is only  $C^{-1}$ -continuous (see Figure 3). Another approach is to cross-fade between the two sounds. However, this is merely a variation on the first technique, and although the resulting sound is perceived to be smoother, the sound source locations still appear to "teleport," and the transition is still discontinuous (see Figure 3).

This suggests that the proper approach is to have the sound follow a path between two consecutive sample points. The most obvious path is a straight line, but this will also be suboptimal. It is clear that if the two sound positions are on opposite sides of the listener's head, then the straight line solution will create a path that passes through the center of the listener's head, thus causing the sound volume to

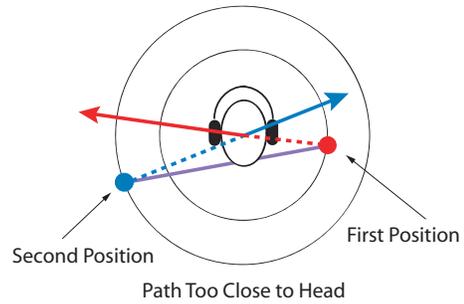


Figure 4: Using a straight line path results in undesirable sound level changes.

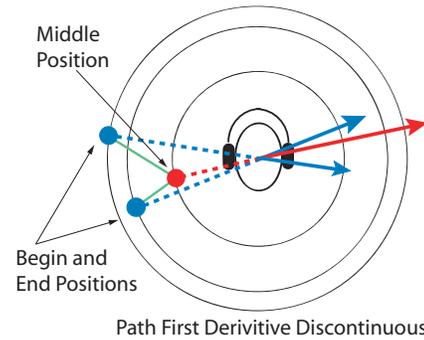


Figure 5: Using continuous angle and magnitude interpolation can result in discontinuities in the first derivative for the volume.

increase and decrease while moving along this path (see Figure 4). This effect is not desirable. A better approach is to employ spherical linear interpolation between the two different vector positions, and vary the position distance linearly along the path. Although the resulting transition appears to be smoother, it still has flaws. Consider the situation with three collinear vector samples. The first and last samples are low velocity vectors. The middle sample, however, is a high velocity vector. Furthermore, for reasons of simplicity, let us assume that all three sample vectors have the same direction. That means that the sound path will head directly towards the listener's head between samples one and two, and then immediately reverse directions and head directly away between samples two and three as depicted in Figure 5. The resulting path is only  $C^0$ -continuous. In order to achieve a smooth sound level transition between sample positions, it is necessary to use a curved path.

In our approach we use a Hermite curve to achieve  $C^1$  continuity. The Hermite curve follows a simple path (see Figure 6). The endpoints of the curve are located at the two consecutive sample positions. Each sample position defines a *distance shell* centered around the center of the listener's head. All points on this shell are the same distance from the center point as the sample position. The tangent plane of the distance shell passes through the sample position, and all valid tangent vectors of the Hermite curve lie on this plane. The tangent vectors used for the curve are very important: We want the curve at each sample point to be tangential

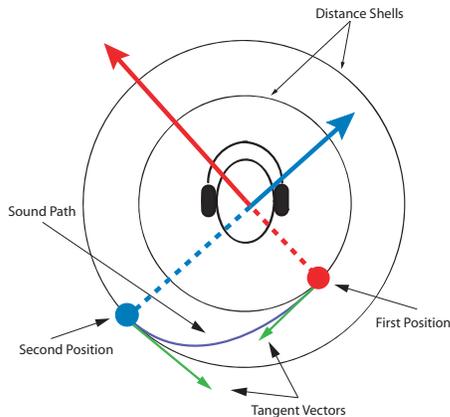


Figure 6: Basic geometry of Hermite curve defined between two subsequent sample positions.

to the corresponding distance shell. In other words, the curve at the start and end points should be orthogonal to the vector passing through the center of the listener’s head and the start or end point. This ensures that the path will arrive and depart from a sample position tangent to its distance shell, which ensures  $C^1$  continuity. Furthermore, the path should also have the property that if both sample points are the same distance from the listener’s head, all points on the path should lie at this distance. This can be achieved by adjusting the length of the tangent vectors of the Hermite curve.

We have obtained good results with the following heuristic approach (see Figure 7): First, trisect the angle defined by the vectors from the origin to the two sample points. This will define two unit vectors, one third and two thirds of the way, conceptually, along the sound path. Second, find the intersection of the vector that is one third of the way along the path with the tangent plane of the start position. The vector from the start position to that intersection point is one of the tangent vectors that will be used for plotting the Hermite curve. The other tangent vector can be determined by symmetric calculation of the end position and the unit vector, two thirds of the way along the path. Using this scheme, the path remains almost equidistant from the listener if the start and end points are equidistant from the listener.

It should be noted that this technique ensures that although the volume remains  $C^1$ -continuous, the transition path is still simply  $C^0$ -continuous. This is due to the fact that the path can take sharp turns at the sample positions. Resolving this problem requires that the two tangent vectors for the paths entering and leaving a sample point lie along the same line. Unfortunately, this also requires the ability to know more than one sample point ahead in time, which can create difficulties in data sets with multiple time steps, or in situations where the user position can vary dynamically. Consequently, this leads to the need to predict both the temporal and spatial paths of the user. Fortunately, we have found that  $C^1$ -continuity of the volume is more important than  $C^1$ -continuity of the path. It appears to be that human perception of abrupt direction changes in the path is not as pronounced as perception of abrupt sound level variations.

By using Hermite curves to approximate the path of vir-

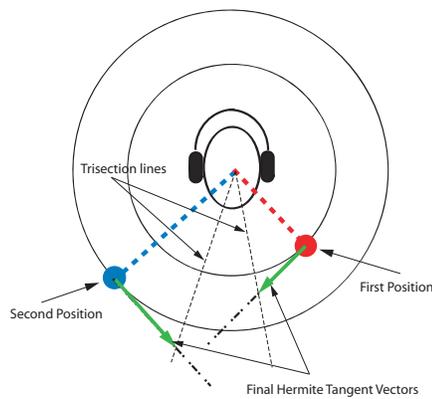


Figure 7: The intersection of the trisection lines with the tangent lines defines the endpoint of the tangent vectors used for the Hermite curve.

| Data Feature           | Sound Property   |
|------------------------|--|
| vector direction       | sound location   |
| vector magnitude       | sound level and pitch  |
| vorticity (turbulence) | also controls sample rate<br><i>gustiness</i> of wind<br>how swiftly and at what<br>velocity the wind sound shifts<br>around |

Table 1: Vector field data conveyed using sound.

tual sound sources, and by moving the sound smoothly along this path at times between sample events, we have created the illusion that the sound is smoothly and continuously shifting between sample positions. This in turn simulates the shifting wind noises associated with turbulent airflow. Similarly if two sample vectors are close together in orientation and velocity, their path will be very short and the two sounds will sound essentially the same, with no noticeable transition between them, thereby simulating smooth and steady wind flow.

## RESULTS

The primary new contributions of this technique are the addition of auditory directional information and of vorticity information through the shifting over time of the sound direction.

Our technique has been implemented and tested on both Microsoft Windows and Linux operating systems, using *Direct Sound* and *OpenAL*, respectively. In both cases, all computers used were commodity desktop computers. Our system has also been tested using headphones as well as with surround sound speakers. We have found that surround sound speakers offer some advantage over headphones, unless the headphones are used alongside a head tracking system.

Using our technique the listener gets a very good sense of the data flow in the local area surrounding his or her virtual position. By navigating within the vector field, the listener is able to gain a good intuitive understanding, using only sound, of the general flow properties of the vector field. Table 1 illustrates the various information conveyed to the listener from these aerodynamic sounds.

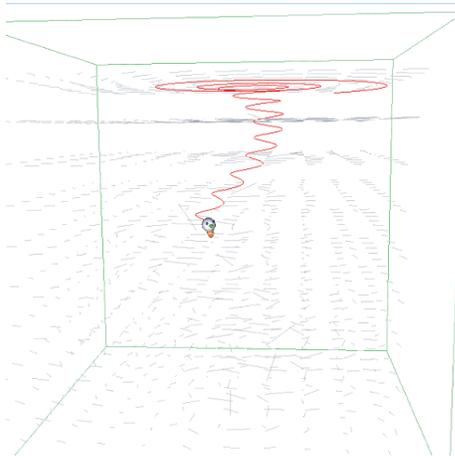


Figure 8: Screen capture from a data exploration session on a laptop computer.

We have used this technique on human scale datasets representing air-turbulence, such as the well known simulated tornado dataset, and a simplified simulated tornado with time-varying data. We have also used it on a dataset representing a smaller scale evolution of an argon bubble moving laterally. Figure 8 shows an exploration session using our system. The user controls a proxy head object with a mouse, and that proxy head trails a traditional streamline to help give a visual cue about the behavior of the vector field.

As mentioned previously, this technique works best with surround sound speakers, unless headphones are used together with a tracking system. This is a very important point to consider when setting up this system, because without a proper audio setup, the process of localizing the sound source is made much more difficult. The end result of poor localization of sound in this system is that it is much more difficult to determine the direction of wind flow.

The superiority of speaker arrays over general headphones is due to a number of factors. First and foremost, when using an array of speakers, the sounds truly are coming from different directions, and it is natural and easy for the user to distinguish where the sounds appear to be coming from. In contrast, when using the headphones, even though HRTFs are quite sophisticated today, it can still be difficult to distinguish front from back and up from down. This issue is compounded by the fact that using headphones with standard sound APIs, we are using a default HRTF. If instead we were using a custom HRTF, tailored to the user's body, it is possible that sound localization with the headphones would be easier. Lastly, since the base sound sample is white noise, a somewhat unfamiliar sound to most people, the transformations applied by the HRTF to the sound may not be as helpful in localization as if those same transformations were applied to a very familiar sound, such as human speech.

Headphones do however become significantly more useful when used with a head tracking system. This is due to the simple fact that now the user can move their head around within the environment and have the sound change as they do this. This can be very useful for not just improved front to back localization, which surround speakers are good at as well, but also up and down localization, which standard speaker setups are not generally as good at addressing. Additionally, as the user moves around within the environment,

the user's position in relation to the different speakers is no longer a concern. One additional benefit is that it is not a problem in theory to add more user's and have each of them hearing different sounds, which would be difficult, to say the least, with a large fixed speaker setup.

Another significant issue we found was the importance of the ratio of the size of the sample volume, essentially the user's head volume, to the density of the vectors within the field. If the sample volume is too small, then most or all of the samples will fall very close to the same vector, and there will be extremely low vorticity heard by the user, even if the neighboring vectors are wildly different. On the other hand, if the sample volume is too large, and encompasses too many vectors, the sounds heard by the user will probably always sound very turbulent, because the user is no longer considering only relatively nearby vectors. The key to having the vorticity sound make sense is to have a reasonable sample neighborhood for the data being examined. Currently, we are doing this experimentally, but we hope to develop more automated techniques for this in the future.

We have found that this technique can be used to augment traditional visualization techniques. Using only sonification, our technique would require a considerable amount of work to fully understand a vector field. However, when used in conjunction with other techniques it helps the listener to achieve a better understanding of the data. Not only can the user see what is happening nearby, but they can also hear where the flow is currently coming from, relatively how fast that flow is, and how turbulent that flow is. The effect can be subtle, but all of the cues, particularly the vorticity data, can help the user quickly understand what is going on within a dataset.

We found that the greatest use for this technique is the information that it provides on vorticity and flow patterns. It is possible to provide visually the direction and magnitude information about a given location, but it is much harder to depict vorticity and local flow behavior. With continually shifting aerodynamic sounds, and their shifting behaviors, it is very easy and natural to understand the vorticity and flow patterns within a data set.

A good example scenario where this would be the case is in evaluating the flow of air around a wing. Using stream ribbons and particle systems, it could be difficult to identify areas of turbulence near the wing. Since the wing is a three-dimensional structure, using LIC is possible, but visually quite busy. On the other hand, with our system, the user could simply walk along the wing, listening as they went for areas of increased vorticity, and in so doing, identify these areas quickly and naturally.

This project is especially useful for situations where additional information (e.g., the interior or exterior views of a vehicle) that the user must explore with their eyes is available, and yet flow (air flow or water flow around the vehicle for example) information must still be conveyed, such as in the wing-air-flow example mentioned above. It can also be useful as an aid to accessibility, in the case where visually impaired users may need to explore the data. Furthermore, this technique requires only inexpensive hardware. Although it does benefit from the availability of head tracking hardware, it is functional with a basic commodity sound card and a set of headphones. In the situation where this technique is used outside of a head-tracked immersive environment, the listener can be represented as an avatar within the dataset. This setup has the ability to convey useful information about local flow patterns in the neighborhood of the avatar, and the avatar can be rotated like a human head to get cues

similar to those obtained by a human operator tilting their head.

## FUTURE WORK

There are several areas where this technique can be further improved. An interesting possibility is to use spatial sound to help navigate to critical points within the data. Calculating the direction of curvature for a streamline at a given point and using that as the sound direction could do this. Then using the local rate of curvature we could derive a pitch and/or volume. This information would combine to give the listener a guess at a direction to navigate in, and approximately how close they are to the critical point. The neighborhood sampling scheme discussed earlier could further be employed (but with averaging as well as interpolation) to smooth out any outliers.

Another possibility is to use dynamically synthesized sounds instead of the static samples currently in use. These dynamic sounds could be based on attributes from the data using a technique similar to that employed by Dobashi et al.[20]. This could allow for a wider range of data conveyed to the user exploring the data. One example might be in a dataset representing mixing fluids. Each fluid might have a different effect on the sound so that the listener could distinguish between which fluid, or mix of fluids, they are currently located in, based on the sound generated.

One key area of interest is to conduct formal user studies to find out in which situations the system is of the greatest everyday use, and to identify facets of human auditory perception that can be further leveraged by our system. Although we have had a number of people use the system, and respond positively to the experience we would very much like to quantify how much it improves a data exploration session. We mean by this: How much does this system speed up the exploration process? How much does it contribute to finding important features in the data? And are there datasets for which it is more or less appropriate to use?

## ACKNOWLEDGEMENTS

We would like to thank Bernd Hamann and the members of CIPIC for the support of this research. The Argon Bubble data set is courtesy of the Center for Computational Sciences and Engineering (CCSE) at the Lawrence Berkeley National Laboratory and the Tornado data set is courtesy of R. Crawfis at Ohio State University and N. Max at University of California, Davis.

## References

- [1] D. Darmofal and R. Haimes, "Visualization of 3-D vector fields: Variations on a stream." <http://citeseer.nj.nec.com/2303.html>.
- [2] B. Cabral and L. Leedom, "Imaging vector fields using line integral convolution," in *Computer Graphics Proceedings, Annual Conference Series*, pp. 263–270, 1993.
- [3] F. Taponecco and M. Alexa, "Vector field visualization using markov random field texture synthesis," in *Proceedings of the symposium on Data visualisation 2003*, pp. 195–202, Eurographics Association, 2003.
- [4] M. Roth and R. Peikert, "Flow visualization for turbomachinery design," in *Proceedings of IEEE Visualization*, pp. 381–384, IEEE CS Press, 1996.

- [5] H. Theisel and H.-P. Seidel, "Feature flow fields," in *Proceedings of the symposium on Data visualisation 2003*, pp. 141–148, Eurographics Association, 2003.
- [6] International Community for Auditory Display. <http://www.icad.org/>, 2003.
- [7] Microsoft Corporation, "Direct sound API for 3D sound." [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9\\_m/directx/sound/directsound.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9_m/directx/sound/directsound.asp).
- [8] Creative Labs Corporation, "OpenAL library for 3D sound." <http://www.openal.org/>.
- [9] V. R. Algazi, R. O. Duda, R. Duraiswami, N. A. Gumerov, and Z. Tang, "Approximating the head-related transfer function using simple geometric models of the head and torso," *J. Acoust. Soc. Am.*, vol. 112, pp. 2053–2064, Nov. 2002.
- [10] UC Davis CIPIC Interface Laboratory. <http://interface.cipic.ucdavis.edu/>.
- [11] M. Naef, O. Staadt, and M. Gross, "Spatialized audio rendering for immersive virtual environments," in *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 65–72, ACM Press, 2002.
- [12] R. K. McCabe and A. A. Rangwalla, "Auditory display of computational fluid dynamics data," in *Auditory Display: Sonification, Audification, and Auditory Interfaces*, edited by G. Kramer. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII, pp. 321–340, Addison Wesley, 1994.
- [13] B. L. Sturm, "Surf music: Sonification of ocean buoy spectral data," *Proceedings of the International Conference on Auditory Display (ICAD), Kyoto, Japan*, 2002.
- [14] P. Astheimer, "Sonification tools to supplement dataflow visualization," in *In Scientific Visualization: Advanced Software Techniques*, edited by P. Palamidese, pp. 15–36, Ellis Horwood, 1993.
- [15] D. A. Rabenhorst, E. J. Farrell, D. H. Jameson, T. D. Linton, and J. A. Mandelman, "Complementary visualization and sonification of multidimensional data," *Extracting Meaning from Complex Data: Processing, Display, Interaction*, edited by E. J. Farrel, vol. 1259, pp. 147–153, 1990.
- [16] S. K. Lodha, C. M. Wilson, and R. E. Sheehan, "Listen: sounding uncertainty visualization," in *Proceedings of the 7th conference on Visualization '96*, pp. 189–ff., IEEE Computer Society Press, 1996.
- [17] S. K. Lodha, T. Heppe, J. Beahan, A. Joseph, and B. Zane-Ulman, "Muse: A musical data sonification toolkit," *Proceedings of the International Conference on Auditory Display (ICAD), Palo Alto, California*, pp. 36–40, 1997.
- [18] G. Eckel, "A spatial auditory display for the cyberstage," *Proc. 5th International Conference on Auditory Display*, 1998.
- [19] C. Volpe and E. Glinert, "Auralization of streamline vorticity in computational fluid dynamics data," *IEEE Visualization*, pp. 51–57, 1997.
- [20] Y. Dobashi, T. Yamamoto, and T. Nishita, "Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 732–740, 2003.