Design and Implementation of 3D Auditory Scene Visualizer Towards Auditory Awareness With Face Tracking

Yuji Kubota, Masatoshi Yoshida, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno
Graduate School of Informatics, Kyoto University
{ykubota, yoshida, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp

Hark, hark, I hear! The Tempest, Willian Shakespeare

Abstract

*If machine audition can recognize an auditory scene containing simultaneous and moving talkers, what kinds of awareness will people gain from an auditory scene visualizer? This paper presents the design and implementation of 3D Auditory Scene Visualizer based on the visual information seeking mantra, i.e., "overview first, zoom and filter, then details on demand". The machine audition system called HARK captures 3D sounds with a microphone array, localizes and separates sounds, and recognizes separated sounds by automatic speech recognition (ASR). The 3D visualizer implemented in Java 3D displays each sound stream as a beam originating from the center of the microphones (*overview mode*), shows temporal snapshots with/without specifying focusing areas (*zoom and filter mode*), and shows detailed information about a particular sound stream (*details on demand*). In the details-on-demand mode, ASR results are displayed in a "karaoke" manner, i.e., character-by-character. This three-mode visualization will give the user auditory awareness enhanced by HARK. In addition, a face-tracking system automatically changes the focus of attention by tracking the user's face. The resulting system is portable and can be deployed in any place, so it is expected to give more vivid awareness than expensive high-fidelity auditory scene reproduction systems.*

1. Introduction

1.1. Machine audition should help people.

If machine audition can recognize an auditory scene that contains simultaneous and moving talkers, what kinds of improvements in awareness will people gain from an auditory scene visualizer? People often complain about audio recordings of meetings; they contain a lot of noise and utterances are blurred by interfering sounds and moving talker, and thus unintelligible. Although they could readily hear what each person said in the meeting, they cannot understand the recordings well. We ascribe this unintelligibility problem mainly to *a lack of auditory awareness*.

Auditory awareness is critical for improving the intelligibility of audio recordings. For example, stereo recording may improve the intelligibility by giving users spatial information such as the 2D localization of sound sources. Some studies have exploited binaural cues to improve the sound localization in listening over headphones. Recently, 5.1-channel and 7.1-channel stereophonic techniques have become popular. Video recording with a stereo microphone may improve such intelligibility by providing visual information about talkers and other sound sources such as a TV, music player, or air-conditioner.

High-fidelity auditory scene reproduction systems have been developed based on wave field synthesis (WFS) [1]. WFS is a holophonic reproduction process that synthesizes, by analogy with visual holograms, an acoustic scene while conserving the spatial characteristics of distance and direction. It usually needs a large number of loudspeakers, say 50–100 in a concert hall. The high-fidelity approach to auditory awareness is expensive. In addition, a psychophysical observation indicates that people may not be able to recognize more than two sound sources at once [2].

We exploit a novel approach for improving auditory awareness. It is based on machine audition, that is, sound source localization, separation, and recognition.

1.2. Machine audition based on computational auditory scene analysis

During the past decade, we have seen the emergence of new research on understanding arbitrary sounds such as environmental sounds [3] and music [4] as well as a mixture of sounds including voiced speech, non-speech sounds, and music. Understanding various kinds of sound sources is a challenging and little-studied area of multimedia, computational intelligence, and visualization. This interdisciplinary research area in machine audition is called *computational auditory scene analysis* (hereafter, CASA) [5].

The three main functions of CASA are as follows.

1. *Sound source localization* identifies and tracks where each sound originates from,

978-0-7695-3454-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ISM.2008.107

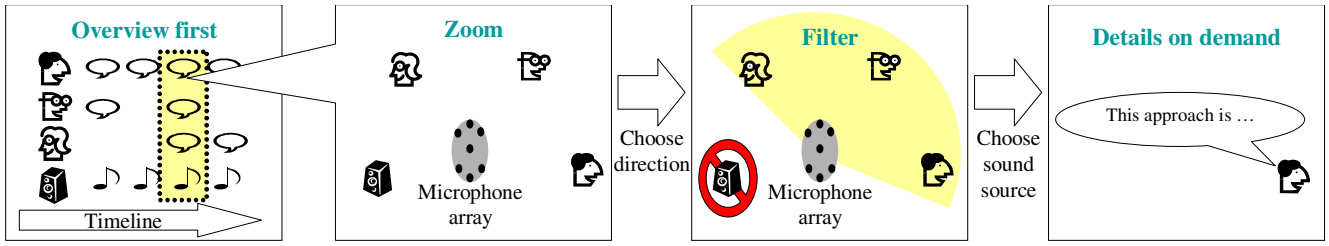

Figure 1. Visual-Information-Seeing Mantra: overview first, zoom and filter, then details on demand (O-ZF-D) functions for an auditory scene (meeting)

2. *Sound source separation* separates multiple sounds that originate from each sound source, and
3. *Separated sound recognition* recognizes a separated sound, e.g., by automatic speech recognition for speech by simultaneous talkers [6] and by beat tracking for music [7].

At a crowded party, one can pay attention to one conversation and then switch to another one. This phenomenon is known as the *cocktail party effect* [8]. It shows that humans have the ability to selectively attend to sound from a particular source, even when it is interfered with by other sounds. This capability is insufficient from the viewpoint of CASA or auditory scene understanding because it does not give an overview of the auditory scene but only a partial aspect.

As a step towards recognizing simultaneous talkers in contrast to a cocktail-party computer, the portable robot audition system called HARK has been developed and released as open source software [9]. They reported an interesting demonstration. When three actual talkers placed a meal order at the same time, a robot equipped with HARK localized, separated, and recognized each meal order and responded by rephrasing each meal order and announcing the total cost of the orders with a delay of 1.9 sec.

1.3. 3D auditory scene visualizer

The 3D visualizer with HARK, a CASA implementation, may provide an inexpensive auditory scene reproduction system with better auditory awareness. This is because CASA functions enable users to localize and recognize each sound source easily. Various functions for recognizing and understanding visual scene have been developed; for example, thumbnails or icons for indexing, zooming in and out for scrutiny and overview, and fast and slow plays for browsing and examining are commonly used in our daily lives. Its equivalent in audition, however, has not been well studied and HARK would provide some novel solutions.

This paper presents the design and implementation of 3D auditory scene visualizer based on the visual information seeking mantra "*overview first, zoom and filter, then details on demand*" (O-ZF-D) [10]. In combination with face tracking, it also provides autonomous focus changing so that users may feel auditory awareness. When the user moves his/her head toward the right, new sounds are heard if they exist. When he/she hears a sound and moves closer to the display, the sound is played louder, and vice versa.

The rest of this paper is organized as follows: Sections 2 and 3 describe the design and implementation of the 3D visualizer with Auditory Scene XML, respectively. Section 4 describes the graphical user interface. Section 5 discusses our observation and Section 6 concludes the paper.

2. Design of 3D Auditory Scene Visualization

The 3D auditory scene reproduction system for improving auditory awareness can be classified into two categories:

1. **Holistic approach**: its goal is mainly to provide a high-fidelity synthesis, and users are responsible for recognizing auditory events.
2. **Reductionistic approach**: its goal is to help users in recognizing auditory events by providing various kinds of auditory information.

Since people may have difficulty in discriminating auditory events as described above, we take the latter approach. For such hierarchical decomposition, auditory scene representation method is also required in addition to a visualizer with CASA functions.

2.1. Design based on O-ZF-D mantra

We designed the 3D auditory scene visualizer on the basis of the O-ZF-D mantra to provide a view of sound sources in a user-friendly manner. Figure 1 shows these three levels at a meeting of three participants with intermittent music.

1. **Overview first (O-level)** provides a temporal overview of an auditory scene by showing where each sound is arriving to the microphones.
2. **Zoom and filter (ZF-level)** provides the presence of sound sources at a specified time by zooming or filtering, and
3. **Details on demand (D-level)** provides information about a specific sound source by playing back an appropriate sound.

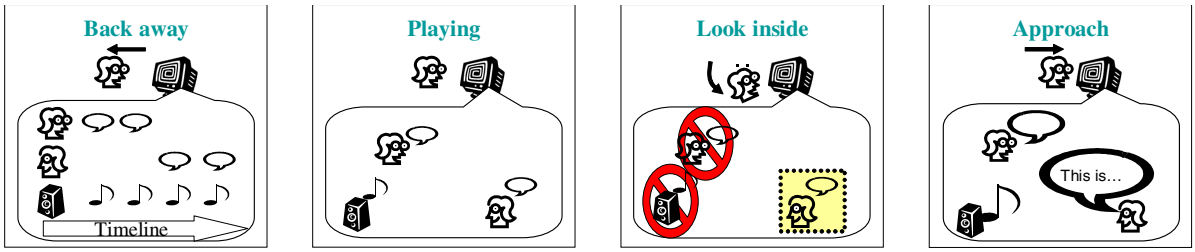

Figure 2. GUI control by three face movements: approach, back away, and look inside

The left box of Figure 1 shows the speech events of the participants and non-speech events of the music in the meeting along a timeline. The O-level function, thus, provides an overview of the sound events, such as who is talking, and how long. The user can get information useful for searching for sound events of interest by looking at the overview.

The middle two boxes show that our system displays the directions and ASR results of sound events at the particular time for which the user wants to get more detailed information than that provided by the overview. The ZF-level function, thus, provides a more detailed display of the presence and component of sounds than the overview and also provides playback of the separated sounds. It improves the intelligibility of separate sounds by helping the user identify the component of the sound.

The right box shows that the D-level function provides the playback and ASR results of the sound event specified by the user in a karaoke manner. Thus, he/she can focus on sounds of interest by looking at the information displayed by the O- and ZF-level functions.

2.2. Design of GUI control by face tracking

In order to improve auditory awareness, *implicit* and *unconscious* control of GUI is also introduced by face tracking. When the user controls GUI by a pointing device such as a mouse, such GUI is explicit and conscious.

Three face movements, **Approach**, **Back away**, and **Look inside**, are exploited for such implicit control. We first explains how such face movements will help visual recognition. When we look at an object such as a globe, we often move our face to change our view of the object. For example, we bring our face closer to read details of placenames printed on it. We move our face toward the right to know the eastern part of the place. We keep our face away from the globe if we want to see the outlines of continents.

On this visual analogy, we designed the 3D viewer interface with face tracking in order to enable the user to search for what he/she wants by changing the content of visualized sound information, and to enable the user to notice an unexpected sound originating from a different direction. These three face movements can control the switching between the auditory scene visualizer's three level functions.

Figure 2 shows a situation in which the user controls by three face movements the information about the meeting described in Figure 1. The second box of Figure 2 from the left shows our system playing back recorded sound.

The left box shows the user backing away from the monitor. The system provides an O-level function to display an overview and play back the sound at high speed. The third box of Figure 2 shows the user looking inside the monitor. When the user wants to discriminate a particular sound event from nearby sound sources, the user always move his/her face to the place where the sound exists. Thus, our system provides a ZF-level function for listening to the sound sources by choosing and filtering.

2.3. Design of Auditory Scene XML

Since the 3D auditory scene visualizer works on archived data as well as online, an auditory scene should be represented symbolically. We designed an auditory scene extensible markup language (XML) for annotating auditory scene by CASA functions. The diagram of auditory scene XML is depicted in Figure 3. The annotating auditory scene descriptions are summarized below:

- **RawInfo**: The configuration of a recorded sound data; microphones, sampling rate and file location (URI).
- **SoundSeparationInfo**: The configuration of signal processing; ShiftSize and FrameSize.
- **MediaTime**: The start and end time points of auditory scene information.
- **MicArray**: The setting of a microphone array.
- **Mic**: The location of each microphones.
- **SoundSource**: The identifier of separated sound source data, ID and file location (URI).
- **FrameVector**: Total frame number of the auditory scene information of separated sound source.
- **Direction**: The elevation and azimuth directions of separated sound source in 3D.
- **SoundType**: This description describes the type of separated sound source is human voice or not.
  - **Likelihood**: Likelihood of the sound type.
  - **SpeechRecognition**: This description describe speech recognition in a karaoke file format when the sound type is human voice.

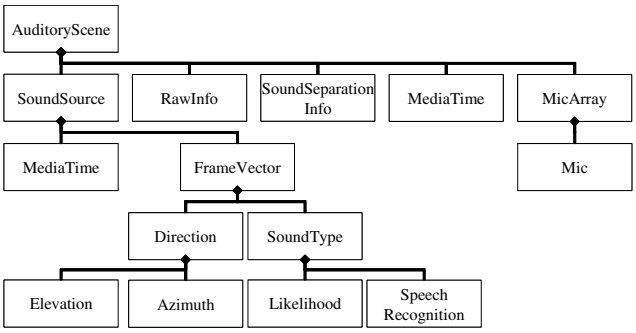

Figure 3. Hierarchy of Auditory Scene XML.

Usually an auditory scene representation in auditory scene XML is created by processing outputs obtained by HARK robot audition system. Figure 4 shows an excerpt from an auditory scene which recorded the situation that a person says "Hello" in Japanese. The excerpt shows that MicArray of 8 elements, SamplingRate of 48 KHz, FrameSize is 1,024 with 512 point shift. The 3D position of each microphone is specified with Mic entry. As explained above, RawInfo and SoundSeparationInfo describe the auditory analysis parameters of HARK. The source information of each separated sounds are described by the SoundSource element are described by the SoundSource element and subelement such as Direction and SoundType.

```
<AuditoryScene>
 <RawInfo SamplingRate="48000" BitSize="16"
          uri="C:/data/example.raw"/>
 <SoundSeperationInfo ShiftSize="512"
                      FrameSize="1024" />
 <MediaTime onset="0" offset="2904" />
 <MicArray size="8">
  <Mic id="1" PositionX="5.91"
       PositionY="5.82" PositionZ="0.00" />
  <!--  6 microphone information is here.  -->
  <Mic id="8" PositionX="-6.93"
       PositionY="0.00" PositionZ="0.00" />
 </MicArray>
 <SoundSource id="1"
              uri="C:/data/example_1.sep" >
   <MediaTime onset="563" offset="1142" />
   <FrameVector size="580">
     <Direction>
       <Elevation>8,8,11,11,10,9,9,9</Elevation>
       <Azimuth>84,85,85,87,87,69,69,69</Azimuth>
     </Direction>
     <SoundType type="speech">
      <Likelihood>85.36,79.61,77.90,62.71,
       53.1,43.92,44.22,44.38</Likelihood>
      <Recognition>[0],KO,[94],N,[119],NI,
       [142],TI,[143],WA</Recognition>
     </SoundType>
   </FrameVector>
 </SoundSource>
</AuditoryScene>
```

Figure 4. Auditory Scene XML (excerpt)

3. Implementation of the System

The 3D auditory scene visualization system consists of three main subsystems as is shown in Figure 5. It is based on a client-server architecture.

1. Computational auditory scene analysis (CASA) system, HARK open source software:
   (a) Audio signal recording module,
   (b) Sound source localization module,
   (c) Sound source separation module, and
   (d) Automatic speech recognition module.
2. Face tracking client system, and
3. 3D visualizer server system.

3.1 HARK robot audition system

The CASA system localize and separates sounds to create the auditory scene XML because our 3D auditory scene visualizer needs to detect the source directions and separate each source from the mixture of sounds. To achieve the O- and ZF-level functions, the source directions must be detected by the sound source localization module. To achieve the D-level function, each source must be separated and recognized from a mixture of sounds by the sound source separation and the automatic speech recognition modules. The modules share the auditory analysis parameters, ShiftSize and FrameSize, which are described at SoundSeparationInfo entry of the auditory scene XML.

Audio signal recording module This module produces an output of multi-channel audio signals. We used Holophone H2-PRO (7.1-channel surround sound microphones) for a microphone array.

The configuration of this microphone is described at Mic and MicArray entries of the auditory scene XML. In addition, the sampling rate of the output is described at SamplingRate attribute of RawInfo entry.

Sound source localization module In this paper, we use a steered beamformer [11] with multiple Kalman filters [12] as a sound source localization method because this method does not require any prior information. In addition, it can track sources robustly when the paths of moving talkers cross, then, the source is given the same ID as an ID of the previous moving talkers. The mean square error of this localization method is 3.6 deg^2 when localized a single

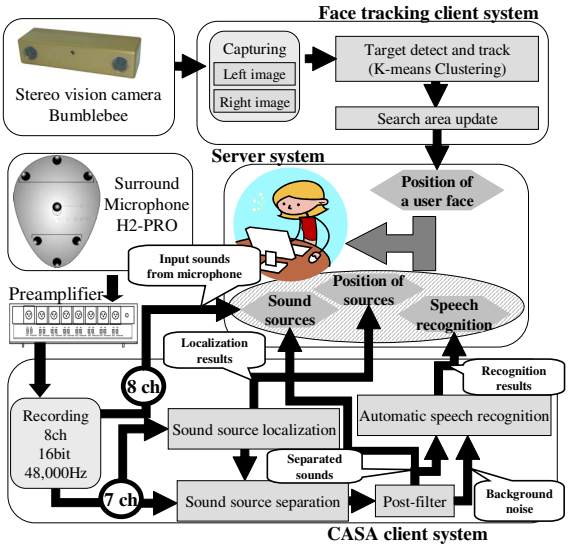

Figure 5. Overview of components in 3D auditory scene visualizer.

loudspeaker which moves around the stationary 8-ch microphone array within 3m[12]. Thus, this sound source localization module is sensible to achieve the O- and ZF-level functions.

The results of tracking sound sources are described at the auditory scene XML: the id of sound sources is described id attribute of SoundSource entry, the time of onset and offset are described at MediaTime entry. In addition, the localization result of sound sources is described at Elevation and Azimuth entries of corresponding SoundSource entry.

Sound source separation module In this paper, we use ManyEars (http://sourceforge.net/projects/manyears/) as a sound source separation module. ManyEars is composed of geometric source separation and multi-channel post-filter[11]. The geometric source separation requires sound source directions as prior information, thus, the sound source localization module sends the localization result. This method was evaluated for separation of three voices (two female, one male) with background noise. The conventional signal-to-noise ratio of the separated results are 12.1 dB (female 1), 9.5 dB (female 2) and 9.4 dB (male). Thus, we assume that the method has enough performance to achieve the D-level function.

The results of separation are sent and archived by the 3D visualizer server system, and the location of the files is described at uri attribute of the SoundSource entry which correspond the localization result.

Automatic speech recognition module In this paper, we use Multiband Julian [13], which is based on the Japanese real-time large vocabulary speech recognition engine Julian. In addition, Julian has the option "-walign" which provides viterbi alignment per word units from the recognition result. We use the option to generate recognition result in Karaoke format. This method is based on missing feature theory (MFT)[9]. MFT uses only reliable acoustic features in speech recognition and masks unreliable parts caused by errors in preprocessing consisting of above sound source localization module and separation module.

The average word correct rates (WCR) of isolated word recognition for three simultaneous speech signals was about 71.0 % [9]. It is hard to provide accurate information, thus, we designed that our system provides the speech recognition when the system works on archived data which includes revised speech recognition by humans. The revised recognition result is described at SpeechRecognition entry.

3.2 Face Tracking

To achieve GUI control by face tracking, our system must detect and track a user's face. The system shown at the top of Figure 5 consists of the following two modules:

1. Clustering and target area tracking module and
2. Search area update module.

In this system, the clustering and target area tracking module detects a user's face and the search area update module tracks it. We used Bumblebee and Triclops library (Point Gray Research) for the stereo vision camera systems to generate depth image and detect the 2D location of the user's face $X_u(k), Y_u(k)$ by clustering and tracking result in frame k. The three face movements are classified by the distance $D_u(k)$ and the azimuth angle $\theta_u(k)$ between user's face and the camera mounted on top of the system's monitor.

$$D_u(k) = \sqrt{X_u(k)^2 + Y_u(k)^2}, \theta_u(k) = \tan^{-1}\left(\frac{X_u(k)}{Y_u(k)}\right)$$

The details of the interface are described in Section 4.2.

Clustering and target area tracking module We use the pixel-wise K-means clustering algorithm with target and background samples [14] to detect and track the user's face. The use of negative information as well as positive information enables the pixel classification to be performed by checking whether a pixel is more similar to the target than to the background. Thus, this algorithm can track robustly in unconstrained environments with limited knowledge and input information.

Search area update module [14] After the clustering, we update the search area according to the target detection result in frame.To determine the shape of the search area, we

apply a Gaussian probability density function to represent the distribution of the detected target pixels.

3.3. 3D visualizer

We implement our 3D visualizer based on the Model-View-Controller (MVC) architecture using Java and Java 3D. Figure 6 shows the MVC architecture of the 3D visualizer. This visualizer has two modes: online and offline. Thus, the visualizer has different contents on each mode.

In the model component, the contents are auditory scene information: ID of sound sources, binary sound data, localization and speech recognition. In the offline mode, the contents are retrieved from archived auditory scene XML when a user selects the scene which he/she wants by controller component. In the online mode, the contents except speech recognition are retrieved from HARK in real-time.

The viewer component contains four contents: timeline viewer, 3D viewer of directions, sound playback component and closed-captioned component of speech recognition in a karaoke-like manner. The contents works in synchronization with the playing back sound. In the online mode, captured sounds are played and the sound source directions are displayed by timeline viewer and 3D viewer. In the offline mode, same information in the online mode and speech recognition results are provided.

The controller component changes the state of the model contents by pointing device and face tracking system. Pointing device is used to click and drag at the graphical user interface of the visualizer. In the online mode, a user can select the directions and ID to change playing sounds. In addition, in the offline mode, a user select the time point and the amount of sounds to skip scenes, then, the model component retrieve information of matching the criteria specified from archived auditory scene XML.

4. GUI: 3D Viewer

4.1 Graphical User Interface

Here, we explain the graphical user interface (GUI) of our system when it provides the auditory scene in which one moving man and one sitting man were talking in a room with some environmental sounds. This recording condition is shown in Figure 7.

The GUI of our system is shown in Figure 8. A control panel (①) has buttons for five ordinary audio functions: Play, Pause, Stop, Fast forward (FFW), and Record. The system can be controlled just like a typical audio system. If the Record button is clicked in the online mode, the captured auditory scene information can be saved as auditory scene XML file. In the offline mode, the FFW button allows high-speed playback. Moreover, the playback location

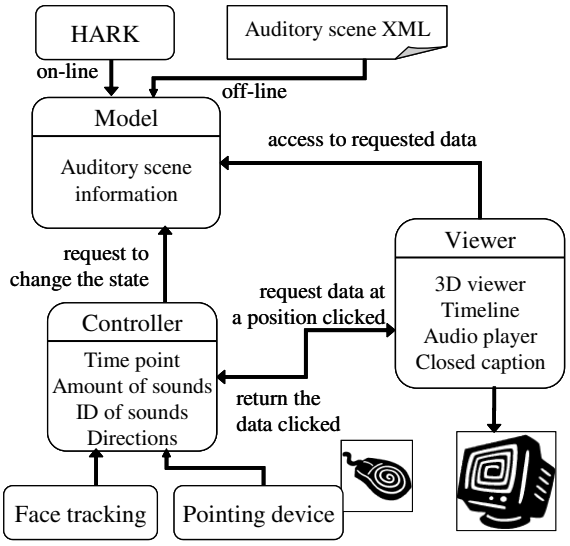

Figure 6. 3D visualizer system architecture based on the MVC model.

can be jumped to a desired time point by clicking a word in the speech recognition results (④) or a point in the timeline (⑤) of the GUI. Furthermore, automatic playback at high speed is possible when the number of sound sources is smaller than the number selected by a user at the right side of the control panel. For example, if the user selects "1", our system skips scenes in which no sound sources exist.

The GUI provides the following three functions.

Overview First Horizontal directions of sound sources are displayed in different colors along a timeline (⑤). However, the directions of sounds that are not human voices are displayed in the same gray color to distinguish human voices clearly in the overview. The horizontal axis is the temporal axis, and the vertical axis shows the azimuths of sound sources. This function achieves the O-level.

Zoom and Filter When the Play button is clicked, the system plays recorded raw sounds. The central display shows the directions of sound sources synchronously while the recorded raw sounds are playing. The directions of the sound sources are displayed by arrow beams (③) centered on the microphone array (②). Unique IDs of sound sources are attached to the beams, and each beam is displayed in the same color as in the graph on the timeline (⑤). In addition, when the system runs in offline mode, the right side of the central display indicates the sound sources that are human voices while synchronously playing sounds by changing the color of the words in a karaoke-like manner (④).

Details on demand To achieve the D-level function, the system must provide an interface that enables a user to choose the sound sources that he/she wants to listen to. Our system provides two interfaces: for choosing a sound source by the unique ID labeling the beam and for choosing by the

range of directions. When an ID on a beam is clicked with a mouse, the beam is highlighted and the sound source corresponding to the beam is played back. When the azimuth and elevation of sound sources are selected by clicking the Direction buttons and dragging, the sound sources in that direction are played back and the beams are highlighted. The selected range of directions is displayed in a different color as a sphere consisting of rectangles (⑥ in Figure 9).

4.2 User interface with face tracking

To click the Face toggle button, our system provide an interface that enables a user to control the content via two ordinary audio buttons, Play and Stop, and three face movements.

This interface enables a user to control the content by the following four operations.

Play and Stop buttons When the Play button is clicked, the system plays back the recorded sound, and the 2D location of the user's face is stored as the origin location, distance, and azimuth angle, which are described by $X_o, Y_o, D_o, and \theta_o$. The classification of the three face movements depends on this origin values. This function is shown in the second box from the left in Figure 2. On the other hand, when the Stop button is clicked, the origin location, distance, and azimuth angle are initialized and the operation of this interface is stopped.

Approach, Back away To achieve the functions in the right and left boxes of Figure 2, our system controls the volume of sounds and the playback speed as triggered by the user approaching or backing away while the system is playing back recorded sound. The volume and playback speed are based on the current distance $D_u(k)$ and the origin distance D_o. When the user approaches the monitor, the interface raises the volume. On the other hand, the interface lowers the volume when the user backs away from the monitor. When the user is a sufficient distance away to mute the volume, our system provides the function of the O-level to control the playback speed.

Look inside To provide the function of the ZF-level (third box in Figure 2), our system plays back sound sources when the user looks inside the monitor to select the range of directions in which he/she wants to listen. When the user moves his/her face around the monitor by more than a threshold angle θ_{thr} or clicks the Direction button on the control panel, our system shows a sphere for selecting the range (Figure 9).

$$\theta_{thr} \leq |\theta_u(k) - \theta_o|$$

We introduced the threshold to prevent false operation. In this experiment, the threshold angle was $\frac{\pi}{12}$. The selected range of directions $\theta_{dir}(k)$ is calculated when the user

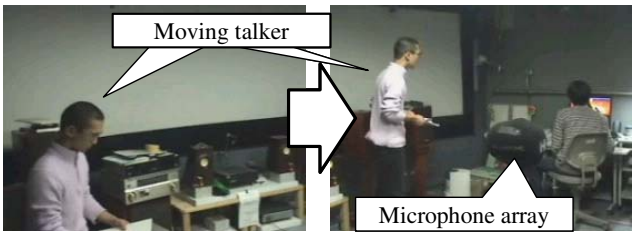

Figure 7. Snapshots of two people's meeting with H2PRO 7.1-ch surround microphone.

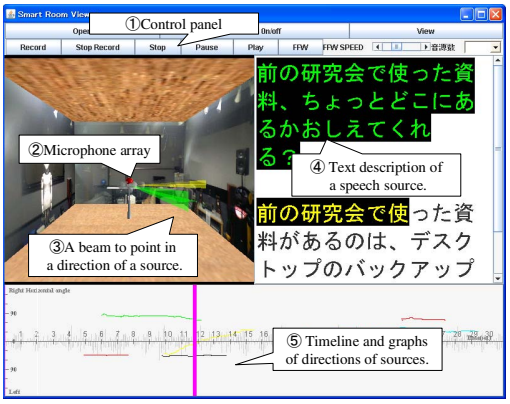

Figure 8. Overview level with sound beams and automatic speech recognition results in a karaoke manner. Time-line zone also provides an overview of sound source localization and separation.

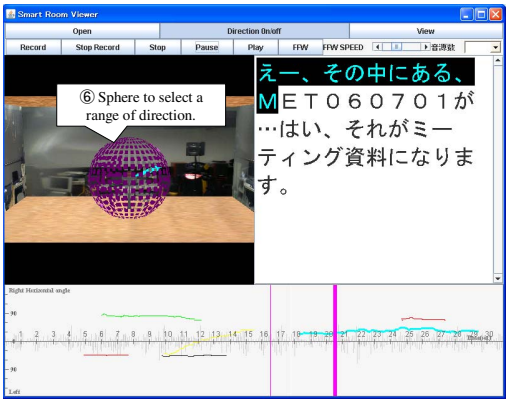

Figure 9. Zoom-and-Filter level with a selected area for playback of a particular sound source.

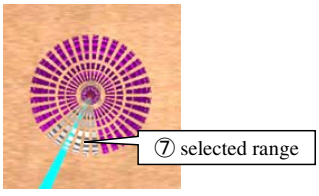

Figure 10. Topdown view of a selected area for sound source playback

moves his/her face around the monitor.

$$\theta_{dir}(k) = \theta_u + \theta_{sys} + \pi,$$

where θ_{sys} is the angle of view at the center of the 3D GUI. A user can change the angle of view in 90-degree increments by clicking the View button on the control panel. In the situation shown in Figure 9, which shows the GUI is top side of Figure 10, our system selects the far end of the right range of directions (⑦) when the user moves his/her face to the left. Thus, our system enables a user to select a range of directions by changing the direction of his/her gaze.

5. Discussion

Our 3D auditory-scene visualizer demonstrated auditory scene visualization based on the concept of "O-ZF-D" using an interface with face movement tracking. Our system enables users to overview and retrieves various sounds obtained by sound-source localization/separation.

Regarding auditory scene visualization, our system provides only two types of information about sounds, the directions of sound sources and the speech recognition results of separated sound. Further information is expected to be obtained from sounds, such as sound type and the loudness. Especially, the environmental sounds can be described by onomatopoeia; sound imitation words. Furthermore, the information of sounds can be visualized in various ways at the O-, ZF-, and D-levels. For example, loudness can be displayed by the width of the graph of sound source directions. The sound type also can be displayed by using labels added to the graph. The labels could be "text" for environmental sounds recognition result. By referring to these labels, users can understand the contents of the sounds without listening them. Thus, we suppose that potential applications of our system include supporting of the hearing-impaired. For example, Tokuda *et al.* reported the auditory scene visualizer using transmissive head mounted display to reduce the bound of the eyes' movement between the presentation apparatus and sound sources [15]. We suppose that the system will enables the hearing-impaired to share auditory information with the hearing people without watching the monitor continuously.

Regarding the interface with face movement tracking, our system uses only the position of the face, which is the direction and distance between the camera and user's face. Further information could be utilized, for example, the acceleration of the moving face and/or the gaze direction. Both of these types of information are changeable depending on the target and degree of user's interest. Thus, various types of additional information could lead to the design of a more efficient interface.

6. Conclusion

In this paper, we reported on our development of a 3D auditory scene visualizer with face tracking. The user interface of the visualizer is based on "overview first, zoom and filter, then details on demand" and introduce the analogy of face movements to determine the user's intention for gain more vivid auditory awareness. The visualizer enables users to find the position in time and the identification what a sound source exists from a mixture of sounds by pointing device and three face movements.

As a result, users can gain at least four auditory awareness, i.e., auditory detection, auditory discrimination, auditory identification and auditory comprehension. The O- and ZF- level functions provide auditory detection and discrimination to enable the user detects and discriminates what sound events exist at a desired time by showing the timeline and the 3D directions of sound sources. The D-level function provides auditory comprehension by showing the speech recognition results in a karaoke manner and playing the separated sound sources of selected directions. To provide auditory identification, our visualizer displays non-speech events in same gray color and displays the information of same human voice event in same color per sound source.

In addition, the users may control naturally the contents of visualized auditory information by face movements. We will approach the desire sound events when we want to listen attentively, thus, our system provides detail information when a user approaches the monitor which shows our system, and vice versa. And our system provides autonomous focus change when the user moves his/her head from side to side to hear new sounds if they exist. Thus, our system can help users in recognizing auditory events by providing above auditory awareness.

Future work includes improving the display of sound source identification results for recognizing the details of non-speech events and compensation. To present the sound source identification results to users, it is necessary to have a system that can identify a mixture of sounds. Moreover, to include an additional function for indicating non human voices in our system, our CASA system should be improved in terms of recognizing a mixture of large vocabulary continuous speech signals with noisy sounds. These functions will make our system convenient for users. The ultimate purpose of our system is to realize sharing the auditory information, i.e., "what you hear is what I see" for supporting the hearing-impaired and revitalize recording and storing auditory scene information.

Acknowledgments

We thank Prof. Toshikazu Wada of Wakayama University for his help in the face tracking module and Dr. Kazuhiro Nakadai of Honda Research Institute Japan Inc. (HRI-JP) for his discussions. We also thank HRI-JP for allowing us to use their Sound Viewer and for their cooperation in developing the HARK robot audition software. We also thank Mr. Satoshi Kaijiri, who graduated from Kyoto University, for his development of early version of Sound Viewer.

References

[1] E. Corteel. On the use of irregularly spaced loudspeaker arrays for wave field synthesis, potential impact on spatial aliasing frequency. *Proceeding of the 9th International Conference on Digital Audio Effects*, ppp.209–214, 2006.

[2] M. Kashino and T. Hirahara. One, two, many – judging the number of concurrent talkers. *Journal of Acoustic Society of America*, 99(4):2596, 1996.

[3] K. Ishihara, T. Nakatani, T. Ogata, and H. G. Okuno. Automatic sound-imitation word recognition from environmental sounds focusing on ambiguity problem in determining phonemes. *PRICAI 2004: Trends in Artificial Intelligence*, LNCS 3157, pp.909–918. Springer Verlag, 2004.

[4] D. P. Ellis. Extracting information from music audio. *Communication of the Association for Computing Machinery*, 49(8):32–37, 2006.

[5] D. Rosenthal and H. G. Okuno, Eds. *Computational Auditory Scene Analysis*. Lawrence Erlbaum Associates, 1988.

[6] J.-M. Valin, S. Yamamoto, J. Rouat, F. Michaud, K. Nakadai, and H. G. Okuno. Robust recognition of simultaneous speech by a mobile robot. *IEEE Transactions on Robotics*, 23(4):742–752, 2007.

[7] K. Yoshii, K. Nakadai, T. Torii, Y. Hasegawa, H. Tsujino, K. Komatani, T. Ogata, and H. G. Okuno. A biped robot that keeps steps in time with musical beats while listening to music with its own ears. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1743–1750. 2007.

[8] E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of Acoustic Society of America*, 25:975–979, 1953.

[9] S. Yamamoto, K. Nakadai, M. Nakano, H. Tsujino, J.-M. Valin, K. Komatani, T. Ogata, and H. G. Okuno. Design and implementation of a robot audition system for automatic speech recognition of simultaneous speech. *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*, pp.111–116. 2007.

[10] B. Shneiderman. *Designing the User Interface (3rd Ed)*. Addison-Wesley Pub., 2003.

[11] J.-M. Valin, J. Rouat, and F. Michaud. Enhanced robot audition based on microphone array source separation with post-filter. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2123–2128. 2004. http://manyears.sourceforge.net/.

[12] M. Murase, S. Yamamoto, J.-M. Valin, K. Nakadai, K. Yamada, K. Komatani, T. Ogata, and H. G. Okuno. Multiple moving speaker tracking by microphone array on mobile robot. *Proceedings of the Nineth European Conference on Speech Communication and Technology*, pp.249–252, 2005.

[13] Nishimura, Y. and Furui, S. Multiband Julius. http://www.furui.cs.titech.ac.jp/mband_julius/.

[14] C. Hua, H. Wu, Q. Chen, and T. Wada. A pixel-wise object tracking algorithm with target and background sample. *Proceedings of the 18th International Conference on Pattern Recognition*, pp.739–742, 2006.

[15] K. Tokuda, K. Komatani, T. Ogata, and H. G. Okuno. Hearing-Impaired-Supporting System in Understanding Auditory Scenes by Presenting Sound Source Localization and Speech Recognition Results in Integrated Manner on HMD, (written in Japanese). *Proceedings of the 70th Information Processing Society of Japan*, 5ZD-7, 2008.