

## 3D Auditory Scene Visualizer With Face Tracking: Design and Implementation For Auditory Awareness Compensation

Yuji Kubota, Shun Shiramatsu, Masatoshi Yoshida,  
Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University  
{ykubota, siramatsu, yoshida, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp

### Abstract

*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”. The machine audition system called HARK captures 3D sounds with a microphone array. The natural language processing called SaliencyGraph visualizes topic transition by using discourse saliency. The 3D visualizer implemented in Java 3D displays topic transition and each sound stream as a beam originating from 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 mode). This three-mode visualization will give the user auditory awareness enhanced by HARK and SaliencyGraph. In addition, a face-tracking system automatically determines the user’s intention by tracking the user’s face. The resulting system will enable users to manage and browse auditory scene files effectively, so it should accelerate and support the information explosion to compensate the lack of auditory awareness.*

### 1. Introduction

#### 1.1. Information explosion about audio recording

The production and storing of information has increased to the extent that we are now seen to be in the midst of an information explosion. Lyman and Varian have reported the estimation that audio and digital video files were stored roughly 12 thousand gigabytes per year on the Internet [1]. However, most of the stored audio files are commercial production and does not include auditory scene information, e.g., audio recordings of meetings, lifelogs and lectures. We estimate that processing and storing auditory scene information contains two issues; difficulty in discriminating a particular sound source from an audio recording, and difficulty in information retrieval and browsing.

Regarding difficulty in discriminating, people often complain about audio recordings of meetings; they con-

tain a lot of noise and utterances are blurred by interfering sounds 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 in machine audio for improving the intelligibility of audio recordings. For example, stereo recording may improve the intelligibility by giving users the spatial information such as the 2D localization of sounds.

Regarding difficulty in information retrieval and browsing, lack of browsability compels us to listen throughout the recorded sound with large data to search for a desired sound. Even if literal information of the recordings (e.g., conference minutes of audio recordings of meetings) are prepared, readers of the transcription of long audio recording may feel difficulty in overviewing the contextual flow and finding the desired scene information. Such overviewing work may be alleviated, if system based on natural language processing is available for automatic identification of participants’ concerns of the meetings, and for tracking various concerns sentence by sentence.

We exploit a novel approach for compensation auditory awareness to revitalize recording and storing auditory scene information. It is based on machine audition and natural language processing.

#### 1.2. Machine audition based on computational auditory scene analysis

Understanding various kinds of arbitrary 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) [2].

The three main functions of CASA are as follows.

1. *Sound source localization* identifies and tracks where each sound originates from,
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 [3].

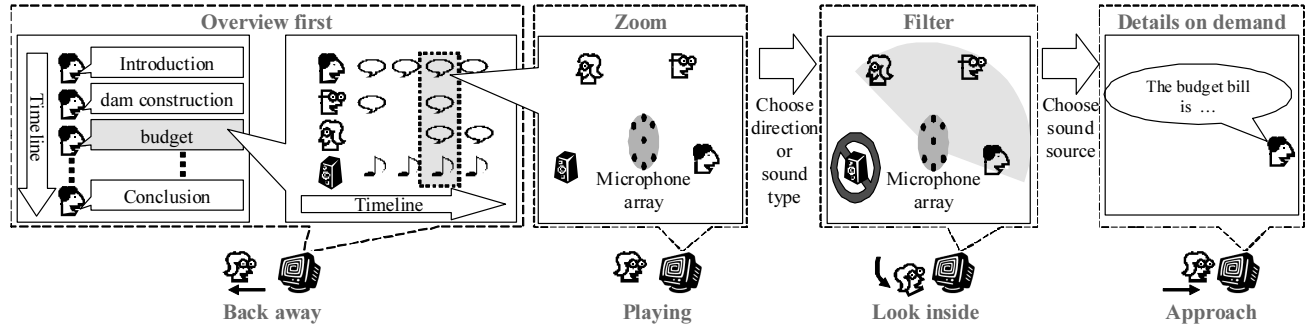


Figure 1. Visual-Information-Seeing Mantra is controlled by three face movements: overview first, zoom and filter, then details on demand (O-ZF-D) functions are switched by approach, back away, and look inside for an auditory scene (meeting)

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* [4]. 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 [5].

### 1.3. Visualizing topic transition based on natural language processing

We reported *SaliencyGraph* which is computed result-ing graph as the temporal change of joint attention to the major latent topics with additional user supplied terms by using discourse saliency [6]. The three main functions of discourse saliency are as follows.

1. *Visualize dynamic transition of topics*: It helps a user to understand an overview of topic dynamics on the basis of *reference probability* which is quantified as saliency of a term for each sentence.
2. *Extract latent topics*: Since a transcription of an auditory scene file contains too many terms to visualize the topic dynamics, several topics are extracted from the transcription by using *Probabilistic Latent Semantic Analysis* (PLSA).
3. *Extract important terms related to topics*: To help a user to grasp the meaning of PLSA latent topics, terms representing the topics are extracted on the basis of *Pointwise Mutual Information* (PMI).

*SaliencyGraph* can be applied to a user interface for browsing long discourse, as illustrated in Figure 2.

- (1) A user can automatically overview the contextual flow

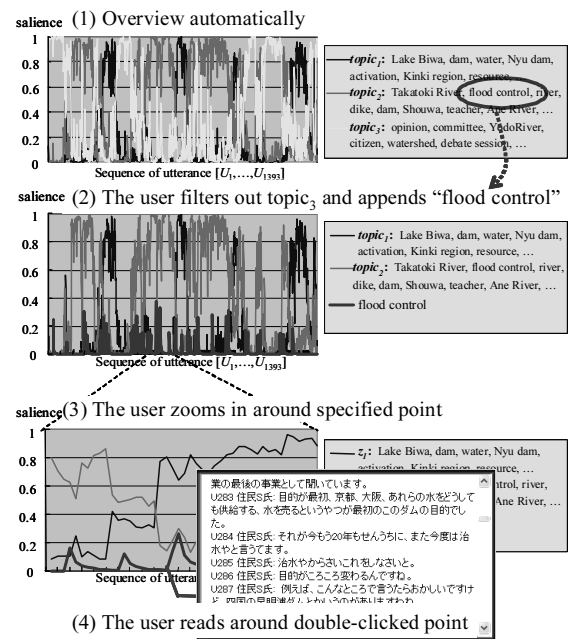


Figure 2. Example of browsing conference minutes using *SaliencyGraph*

- (1) The user can automatically overview the contextual flow of discourse on the basis of the latent topics and the terms representing each topic.
- (2) The user filters out *topic<sub>3</sub>* and appends the term "flood control" according to her interest.
- (3) She zooms in around specified point, a likely mixture point among *topic<sub>1</sub>*, *topic<sub>2</sub>*, and "flood".
- (4) She reads details by double-clicking on it.

This user interface should be effective in browsing long discourse and analyzing the discussion.

### 1.4. 3D auditory scene visualizer

The 3D visualizer with *HARK* and *SaliencyGraph* may provide an auditory scene browsing and management system with better auditory awareness. This is because CASA

functions of HARK enable users to localize and recognize each sound source easily. Various functions for understanding and searching 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 and SaliencyGraph 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) [7]. 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

### 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 and topic transition in a user-friendly manner. The top of Figure 1 shows these three levels at a meeting of three participants with intermittent music.

1. **Overview first (O-level)** provides two temporal overviews: agendas by showing topic transition of meeting and 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, and
3. **Details on demand (D-level)** provides information about a specific sound source by playing back an appropriate sound.

The left box of Figure 1 shows the flow of participants’ topic of the whole meeting. The second box shows speech events of the participants and non-speech events of the music at the specified term in the meeting along a timeline. The O-level function, thus, provides two overviews, topic transition and sound events, such as what participants are talking about and who is talking. The user can get information useful for searching for sound events of interest by looking at the two overviews.

The third and fourth boxes show the 3D directions and automatic speech recognition (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 more detailed information than the O-level function by showing directions in 3D space and playback of the separated sounds. It improves the intelligibility by helping the user discriminate desired sound source from a mixture of sounds.

The right box shows that the D-level function provides the playback and ASR results of the chosen sound source 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 explain 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 place-names 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 GUI with face tracking in order to determine the user’s intention and notice an unexpected sound originating from a different direction for giving auditory awareness. These three face movements can control to switch between the O-ZF-D level functions.

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

The left part 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 part of Figure 1 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 based on CASA functions works on archived data as well as online, an au-

itory scene should be represented symbolically. We designed an auditory scene extensible markup language (hereafter, ASXML) for annotating auditory scene by CASA functions. The annotating auditory scene descriptions are summarized below:

- **RawInfo:** The configuration of a recorded sound data; sampling rate and file location (URI).
- **SoundSeparationInfo:** The configuration of signal processing; ShiftSize and FrameSize.
- **MediaTime:** The start and end time points of recording sound and separated sound sources.
- **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 (speech) or not.
  - **Likelihood:** Likelihood of the sound type.
  - **SpeechRecognition:** Speech recognition in a karaoke file format when SoundType is speech.

Usually an auditory scene representation in auditory scene XML is created by processing outputs obtained by HARK robot audition system. The details of the processing are described in Section 3.1.

### 3. Implementation of the System

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

1. 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,
3. SaliencyGraph client system, and
4. 3D visualizer server system.

#### 3.1. HARK robot audition system

The CASA system localizes and separates sounds to create the ASXML for visualization by the O-ZF-D level functions. To achieve the O- and ZF-level functions, the source

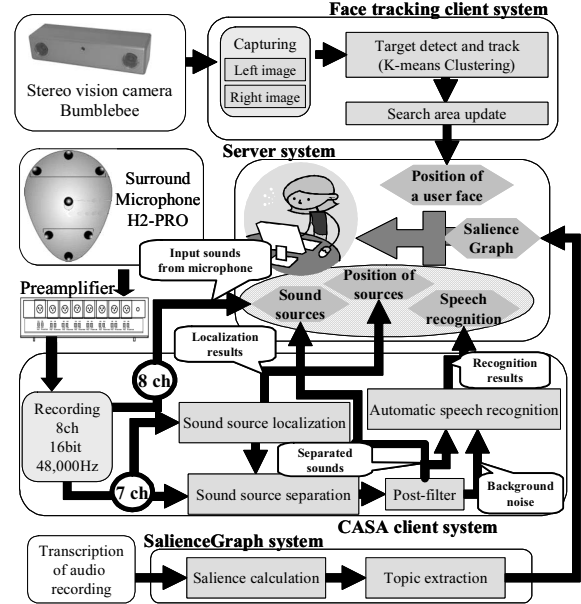


Figure 3. Overview of components in 3D auditory scene visualizer.

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 ASXML.

**Audio signal recording module** This module produces an output of multi-channel audio signals. We used Holo-phone 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 ASXML. 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 [8] with multiple Kalman filters [9], because this method does not require any prior information. In addition, it can track sources robustly when the paths of moving talkers cross. The mean square error is  $3.6 \text{ deg}^2$  when localized a single loudspeaker which moves around the stationary 8-ch microphone array within 3m[9]. Thus, this sound source localization module is sensible to achieve the O- and ZF-level functions.

The result 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 which is composed of geometric source separation and multi-channel post-filter[8]. 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. As a result, the conventional signal-to-noise ratios 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** We use Multi-band Julian [10], 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)[5]. 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 of isolated word recognition for three simultaneous speech signals was about 71.0 % [5]. 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 3 consists of two modules: clustering and target area tracking module, and search area update module.

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 [11] 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 [11]** 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. SaliencyGraph

We designed SaliencyGraph provides the overview when the system works offline mode to achieve O-level function of whole audio recording when the transcription is prepared. The system shown at the bottom of Figure 3 consists of two modules: saliency calculation module and topic extraction module.

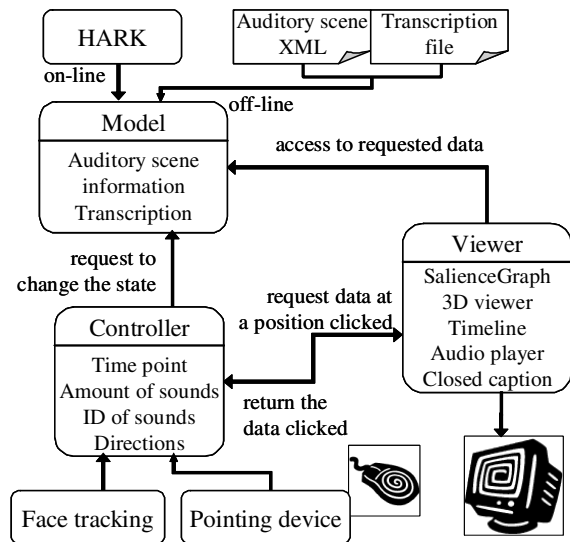
**Saliency calculation module** We used *reference probability* as a metric for discourse saliency on the basis of centering theory [6]. Saliency of each term is calculated at each sentence because it dynamically changes sentence by sentence. Since this requires to extract linguistic features, the transcription should be analyzed by CaboCha [12], a Japanese dependency parser. The analysis result is annotated with Global Document Annotation (GDA) [13], a XML tagset for linguistic information. Saliency values are calculated by using a logistic regression model acquired from a GDA corpus (e.g., a corpus of conference minute).

**Topic extraction module** After the saliency calculation, we extract latent topics by using *probabilistic latent semantic analysis* (PLSA) [14]. PLSA compresses saliency values of a lot of terms into those of several latent topics. Topic transition can be visualized as saliency dynamics of the topics. Furthermore, to represent meaning of the latent topics, they should be associated to related terms. We use the following weighted *pointwise mutual information* (PMI) to extract the terms  $w$  representing each topics  $z$ .

$$\Pr(w|z)\text{PMI}(w, z) = \Pr(w|z) \log \frac{\Pr(w, z)}{\Pr(w)\Pr(z)}$$

### 3.4. 3D visualizer

We implement our 3D visualizer based on the Model-View-Controller (MVC) architecture using Java and Java



**Figure 4. 3D visualizer system architecture based on the MVC model.**

3D. Figure 4 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 transcription and information: binary sound data, ID, localization and speech recognition of sound sources. In the offline mode, the contents are retrieved from archived transcription file and ASXML when a user selects the scene which he/she wants by controller component. In the online mode, the contents except speech recognition and transcription are retrieved from HARK in real-time.

The viewer component contains five contents: SaliencyGraph, timeline viewer, 3D viewer of directions, sound playback component and closed-captioned component of speech recognition in a karaoke-like manner. The contents work 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, SaliencyGraph and speech recognition results are also 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 selects the time point and the amount of sounds to skip scenes, then, the model component retrieve information of matching the criteria specified from archived ASXML.

## 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 5.

The GUI of our system is shown in Figure 6. 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 ASXML file. In the offline mode, the FFW button allows high-speed playback. Moreover, the playback location can be jumped to a desired time point by clicking a word in the speech recognition results (④) or a point in the timelines (⑤, ⑥). 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 and salience of the chosen topics (⑦) are displayed in different colors along a timeline (⑤, ⑥). However, the directions of non-speech sounds are displayed in the same gray color to distinguish human voices clearly. The each horizontal axis is the temporal axis, the top vertical axis shows the azimuths and the other vertical axis shows the salience.

**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 (⑧ in Figure 7).

#### 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 lower part from the left of Figure 1. 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 lower parts of Figure 1, 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 lower part in Figure 1), 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  $\theta_{thr}$  or clicks the Direction button on the control panel, our system shows a sphere for selecting the range (Figure 7).

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

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 moves his/her face around the monitor.

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

where  $\theta_{sys}$  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

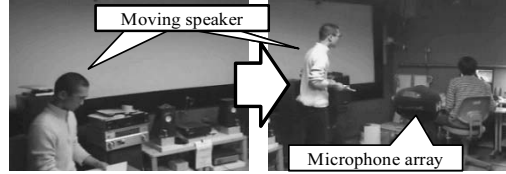


Figure 5. Snapshots of two people’s meeting with H2PRO 7.1-ch surround microphone.

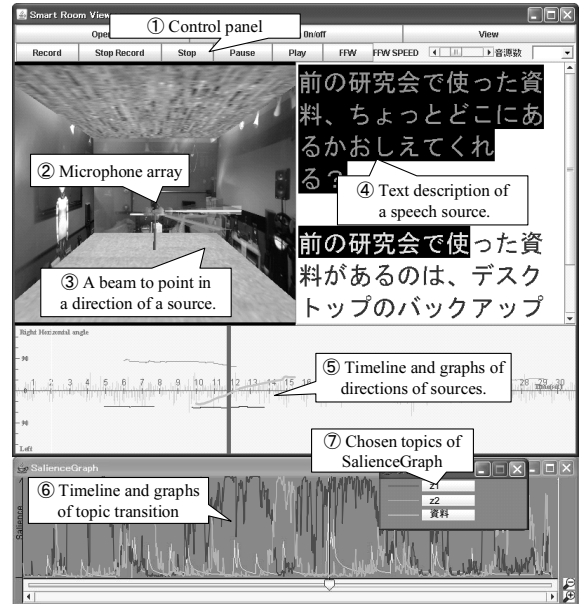


Figure 6. The GUI of 3D auditory scene visualizer. ZF-level with sound beams and speech recognition results in a karaoke manner. O-level with two timelines provides topic transition and sound source localization.

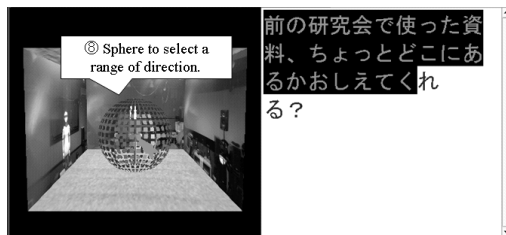


Figure 7. Zoom-and-Filter level with a selected area for playback of a particular sound source.

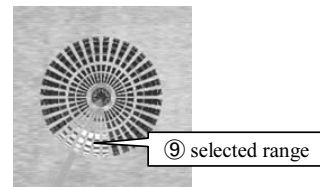


Figure 8. Topdown view of a selected area for sound source playback

the situation shown in Figure 7, which shows the GUI is top side of Figure 8, our system selects the far end of the right range of directions (9) 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 HARK and SaliencyGraph.

Regarding the CASA functions of HARK, our system provides only two types of information about sounds, the directions and the speech recognition results of sound sources. 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.

Regarding SaliencyGraph, our system provides only topic transition for overview when it works on prepared transcription of the recording. If the transcription are retrieved from the ASR results or real-time transcription such as shorthand, SaliencyGraph will work on online.

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 understand the transition of topics and sound events, and the identification what a sound source exists from a scene.

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 transition 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 different events of speech in different light colors and displays non-speech events in gray color.

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 retrieval and browsing auditory events by providing above auditory awareness. We plan that our system will make the auditory scene information more accessible to revitalize recording and storing.

## References

- [1] P. Lyman and H. R. Varian. "How Much Information", 2003 <http://www.sims.berkeley.edu/how-much-info-2003>
- [2] D. Rosenthal and H. G. Okuno, Eds. *Computational Auditory Scene Analysis*. Lawrence Erlbaum Associates, 1988.
- [3] J.-M. Valin, *et al.* Robust recognition of simultaneous speech by a mobile robot. *IEEE Trans. Robotics*, 23(4):742–752, 2007.
- [4] E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *JASA*, 25:975–979, 1953.
- [5] S. Yamamoto, *et al.* Design and implementation of a robot audition system for automatic speech recognition of simultaneous speech. *IEEE ASRU-2007*, pp.111–116. 2007.
- [6] S. Shiramatsu, *et al.* SaliencyGraph: Visualizing Saliency Dynamics of Written Discourse by Using Reference Probability and PLSA. *PRICAI '08*, (in printing).
- [7] B. Shneiderman. *Designing the User Interface (3rd Ed)*. Addison-Wesley Pub., 2003.
- [8] J.-M. Valin, *et al.* Enhanced robot audition based on microphone array source separation with post-filter. *IEEE/RSJ IROS'04*, pp.2123–2128. <http://manyyears.sourceforge.net/>.
- [9] M. Murase, *et al.* Multiple moving speaker tracking by microphone array on mobile robot. *EUROSPEECH'05*, pp.249–252.
- [10] S. Furui, *et al.* [http://www.furui.cs.titech.ac.jp/mband\\_julius/](http://www.furui.cs.titech.ac.jp/mband_julius/).
- [11] C. Hua, *et al.* A pixel-wise object tracking algorithm with target and background sample. *ICPR '06*, pp.739–742, 2006.
- [12] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. *CoNLL-02*, pp.1–7, 2002.
- [13] K. Hasida. <http://i-content.org/GDA/>.
- [14] T. Hofmann. Probabilistic Latent Semantic Indexing. *SIGIR '99*, ACM Press, pp.50–57, 1999.