Soundr: Head Position and Orientation Prediction Using a Microphone Array

Jackie (Junrui) Yang Gaurab Banerjee Vishesh Gupta Monica S. Lam James A. Landay
Stanford University, Stanford, CA, USA
{jackiey, gbanerje, vgupta22, lam, landay}@stanford.edu

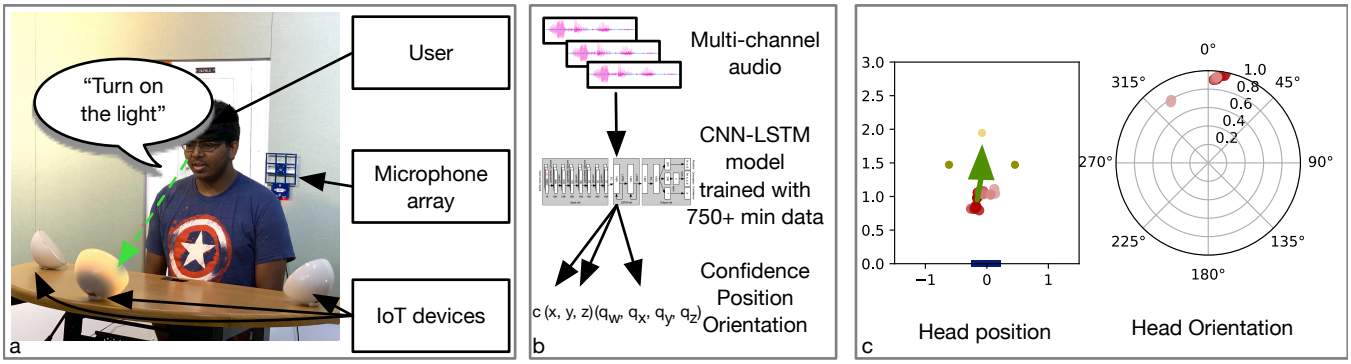


Figure 1: Soundr allows smart speakers to know the user's location and in which direction they are talking. (a) User looks at a connected lamp and tells it to turn on. (b) Our CNN-LSTM model, trained with 700+ min of data, processes the multi-channel audio from the microphone array. (c) Soundr outputs the predicted head position and orientation and turns on the intended light.

ABSTRACT

Although state-of-the-art smart speakers can hear a user's speech, unlike a human assistant these devices cannot figure out users' verbal references based on their head location and orientation. Soundr presents a novel interaction technique that leverages the built-in microphone array found in most smart speakers to infer the user's spatial location and head orientation using only their voice. With that extra information, Soundr can figure out users references to objects, people, and locations based on the speakers' gaze, and also provide relative directions.

To provide training data for our neural network, we collected 751 minutes of data (50x that of the best prior work) from human speakers leveraging a virtual reality headset to accurately provide head tracking ground truth. Our results achieve an average positional error of 0.31m and an orientation angle accuracy of 34.3° for each voice command. A user study to evaluate user preferences for controlling IoT appliances by talking at them found this new approach to be fast and easy to use.

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376427

Author Keywords

Smart speakers; Internet of Things; Machine learning; Acoustic source localization

CCS Concepts

•**Human-centered computing** → **Natural language interfaces; Sound-based input / output; Interaction techniques; •Computing methodologies** → *Neural networks;*

INTRODUCTION

Voice-based virtual assistants have recently seen wide adoption. According to a report from Voicebot.ai and Voicify [3], 26.2 percent of adults in the U.S. own a smart speaker. However, current-generation smart speakers only recognize the content of the user's speech, while a human can also identify where the user is and who or what they are addressing. Future smart speakers should learn this as well to let users express their intentions more easily; from the user's location and gaze, the assistant can infer the user's references to objects, people, and locations, and also provide relative directions.

Notably, humans can distinguish the position and orientation of other speakers with only an auditory signal [18]. This inspired us to use the multi-channel microphone that is already present in most smart speakers to obtain this information. The multi-channel microphone is traditionally used to increase the sensitivity of the microphone in any specified direction, thus recognizing speech from afar. We can leverage the extra information embedded in the raw audio streams collected from the microphone, namely the phase differences between

the sound waves received by different microphones and sound reflection patterns, to know where the user is talking from and in which direction the user is talking towards.

Prior work has also investigated using multi-channel audio to predict the user's position and orientation. Traditional research [5, 7, 8, 19, 25] in this area uses signal processing approaches that are prone to noise, preventing them from mass adoption. Recent advances in this area involve using machine learning to directly process the audio input [26, 27, 31], but the results are disappointing since they suffer from a lack of training data.

To build a user position and head orientation prediction system that works in real life, we leverage virtual reality (VR) technology to collect data. Since commercial VR technology can track a user's head position and orientation accurately while blocking little to none of the user's voice, Soundr uses a VR headset to collect ground-truth tracking data. We collected more than 750 minutes of training data, which is 50 times larger than that obtained by the best prior work [13]. We proceeded to design a machine learning model that uses a convolutional neural network (CNN) with long-short-term-memory (LSTM) [9] architecture to produce accurate position and orientation results at low latency. Based on our evaluation, our model can reach 0.31m average error on position and 34.3°average error on orientation if given training data from the same room and the same user.

With that extra information, Soundr can figure out a user's references based on their gaze. One important application for this is to control Internet of Things (IoT) appliances within a room. With Soundr, users can simply look at and talk to the appliances that they want to control without having to name each of them individually, e.g., *"Turn on the light"* to turn on the light in front of the user. They can also express their intentions directly and the system will be able to fulfill their request based on where they are, e.g., *"Make this area brighter"* to turn up the lights for the local area. We evaluated users' preference for such a system by comparing it with two baseline conditions: controlling a device by its name and using a phone-based augmented reality (AR) app. The results show that users can complete the same control tasks faster with Soundr than with both other conditions.

The contributions of this project[1] include:

1. A novel interaction technique that uses a microphone array[2] for head tracking so smart speakers can understand references to objects, people, and locations based on the speakers' gaze, and also provide relative directions.

2. The first algorithm that detects head orientation of a human speaker with a single microphone array, achieving an average orientation error of 40°, and a positional error of 0.33m for different users in the same room.

3. An evaluation that shows users using our Soundr system can control IoT devices faster than using AR or conventional speech methods.

[1]Source code and dataset: https://jya.ng/soundr

[2]We used a miniDSP UMA-16 microphone array for both data collection and the user study.

RELATED WORK

There are three categories of related work for Soundr: multi-modal interactions involving voice, acoustic source localization, and head orientation tracking from audio.

Multi-modal interaction with head tracking

Soundr allows users to use their voice and head orientation to control IoT devices. Prior work has also described other kinds of interactions that involve head orientation, along with other modalities.

Malkewitz [14] presents one of the earliest works in this area, which demonstrates the possibility of using head orientation and speech input to control a graphical user interface. Ronzhin and Karpov expand on this technology for accessibility purposes [21]. Ito applies the same concept to control multiple home appliances [10]. This research requires the user to wear a specially made headset to achieve head tracking and audio capture. Jeet et al. also propose a similar system for accessibility purposes [11]. To solve the problem of head tracking without requiring the user to wear a headset, Segura et al. [24] propose using a network of microphones and multiple video cameras and fusing their result to achieve this task.

In general, although prior work proposed a similar application, due to the limitations of their technology, these systems cannot be implemented in a smart speaker. Soundr, on the other hand, only relies on audio data collected from a small-sized microphone array, and can thus implement the aforementioned applications entirely on a smart speaker.

Acoustic source localization

Traditionally, predicting the position of a sound source is a signal processing problem [31]. There are three approaches [5, 8]: time-delay-based, beamforming-based, and high-resolution-spectral-estimation-based methods. Time-delay-based methods compute the delay between received signals and compute the position of the sound source based on the position of the microphones and time delay [5,25]. Beamforming-based methods add the streams of audio signals received from multiple microphones with a certain delay (steering) to form an audio signal that amplifies the sound signal transmitted from a specific position in space. By computing the power of the signal when the microphone is steered at different places (Steered Response Power), we can find the actual position of the sound source [8, 19]. Spectral-estimation-based methods first estimate the wavelength of the signal by doing spectral analysis. Then they filter the raw audio with that wavelength to produce a series of narrowband signals. Finally, they estimate the position of the sound source by finding the time delay on that narrowband [22]. These traditional methods usually need the microphone array to have a size comparable to the distances of potential sound sources, e.g., to estimate the speaker's position in a 3.4m x 5m room, they need a 2.1m microphone array [7]. A microphone of this size is hard to fit in a smart speaker, which typically measures less than 20cm.

Recently, there is also related work on using machine learning methods on the audio data to leverage the sophisticated capabilities of deep neural networks to improve single speaker

position prediction [26, 27, 31]. However, machine learning requires a large dataset, which is not easy to acquire for speaker positioning. Prior work relies on human annotation of recorded video data, which limits the size of the dataset and the accuracy of the groundtruth. This limits the accuracy of the related work, especially when tested with human subjects [31].

The largest dataset used with real human speaker data is the IDIAP AV16.3 dataset [13], which has only 15 minutes of data. These types of datasets tend to be composed of multi-channel audio and video recordings that use human annotators to add labels on the position of the user. This method is prone to human error and costly to generate a large amount of data. Other previous work uses a loudspeaker to simulate human voices and positions the loudspeakers using either human labor or robots at different positions/orientations of the room. While less labor-intensive, the number of positions and orientations is usually limited and the sound characteristics of a loudspeaker are different than the actual human voice.

In general, traditional methods for position and orientation from audio data are less successful in terms of accuracy while more recent work using neural networks suffer greatly from a lack of data. In comparison, Soundr uses its unique data collection system to collect a much larger and more accurate dataset for training. This enables us to build complex models with better performance than prior work.

Head orientation from multi-channel audio

Soundr not only predicts the user's position from the captured audio, but also the user's head orientation. Head orientation prediction is a hard problem [17]. Most related work in this area uses several microphone arrays distributed around the room to predict the direction of the speaker [4, 6, 16, 17]. Ryoichi et al. present an algorithm based on Hidden Markov Models to predict speaker orientation using single-channel audio [28], but the result is not competitive with those that use multiple microphone arrays. Some prior work leverages existing algorithms for position prediction, such as Cross-power Spectrum Phase, and extracts many coefficients derived from the algorithm as a feature vector for machine learning [29]. However, these attempts suffer greatly from noise (achieving only 47.5% accuracy when the signal to noise ratio (SNR) is 20 dB) and their training data is collected from a loudspeaker instead of a human speaker. "Are you talking to me" uses a smaller microphone array, but only distinguishes whether the user is talking in the direction of the microphone, which makes it less usable in our example applications [15]. In comparison, Soundr has a large dataset with real human users recorded from multiple different noisy rooms. With the help of machine learning, Soundr works well, even with noise.

SYSTEM DESIGN

We present the system design of Soundr in this section with an overview of how the Soundr system works, our design rationale, the details of our data collection system, and the machine learning system.

Architectural overview

Soundr provides the voice-based virtual assistant with a new interaction modality: head position and gaze direction without any additional hardware. This new modality plays an important role in human conversations, but it is not supported in current electronic appliances.

There are three possible ways to infer a user's head position and direction: 1) collect this information from a device that is on the user's body, 2) infer it from image data from a camera, and 3) calculate it using sound collected from a microphone array. The first two approaches both have their practical and social limitations and the third one appears ideal since it collects no additional personal private information, makes the least assumptions of the user, and requires no modification to the smart speaker hardware.

However, the current state-of-the-art techniques are inaccurate and not robust. As discussed above, traditional signal processing methods are prone to noise; neural-network-based approaches have great potential, but still cannot reach the level of performance necessary for real applications.

We identify three areas of improvement for neural network-based approaches and present the corresponding solutions.

(1) Need for more data. We need a wider and deeper model to produce accurate prediction results, and a wider and deeper model needs more data to train on. We developed a data collection system to automatically collect audio recordings with position and direction labels by leveraging virtual reality (VR) tracking systems. This method allows us to substantially scale up the collection of training data. This system provided us with 50x the amount of data of prior work.

(2) Ability to leverage information in long audio clips while maintaining a low latency. Prior work shows models with longer audio clips as input can produce better results, but those models also create longer latency from input to output, which is not ideal for interactive applications. Also, models that can process longer audio are more complex, and therefore harder to train. To process longer audio clips while maintaining low latency and model simplicity we couple a convolution neural network with a LSTM network with one hidden layer.

(3) Processing based on voice commands. Prior work predicts the head orientation and position using a short audio segment (~0.1 seconds). However, for our applications, we only need one result for each voice command (2 to 3 seconds). Just averaging the predicted head positions and orientations over all audio segments is not ideal since not every segment in a voice command results in the same accuracy. Some audio segments may have more distinctive sounds and thus give better results and other segments may not contain the user's voice. To better process audio associated with a voice command, Soundr uses a machine learning model that produces the position and orientation of the user and a confidence value that is used to compute a weighted average over positions and orientations produced from all the audio segments in a voice command. This allows the model to give a higher confidence value when an audio segment is more distinctive for predicting position and orientation. By computing this weighted average, Soundr can get higher accuracy than computing a simple unweighted average of the result of all audio segments in a voice command.

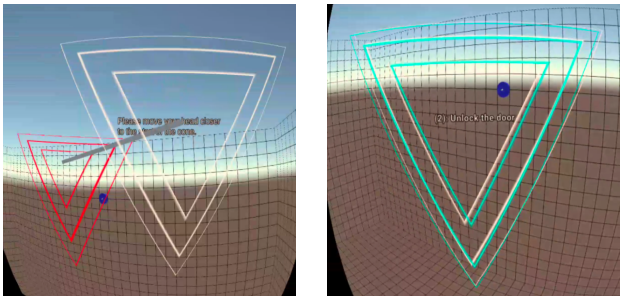


Figure 2: Data collection VR user interface
Left: User cone (white) is not aligned with target cone (red). User will move towards the target cone. Right: User cone (white) is aligned with target cone (cyan). User will be prompted to read the command "Unlock the door".

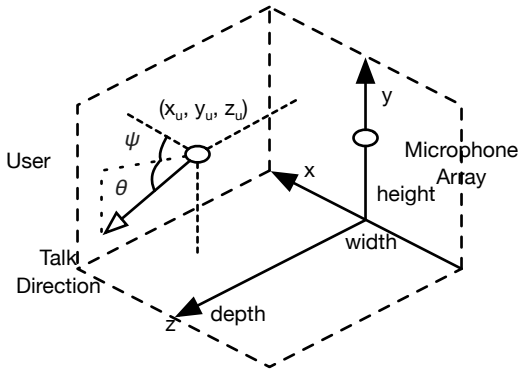


Figure 3: Coordinate system used in data collector:
The ground is the x-z plane. The microphone is placed on a wall, designated as the $x-y$ plane. The microphone is always positioned as $(0, 1.5, 0)$, and therefore 1.5 meters off ground. The user's head position can be defined as (x_u, y_u, z_u). The user's orientation can be defined as pitch θ and yaw ψ.

To sum up, Soundr collects training data that contains audio data from a multi-channel microphone array, user positioning data (position and orientation) from the VR headset tracking data, and voice activity detection (VAD) data by running state-of-the-art VAD algorithms [1] using the audio collected on the VR headset. We use these data sources to train a machine learning model that gets multi-channel audio data as input and outputs position, orientation, and confidence level.

When the end-user is using the smart speaker for our test applications, we collect the multi-channel audio data from the smart speaker, and then run it through our algorithm to get the position, orientation, and confidence. Similar to most smart speakers, we also pass the audio data to a voice recognition algorithm to get the content of the user's speech. We then map the position, orientation, and confidence data from our model to the time range of our voice recognition result and compute the weighted average according to the confidence as the position and orientation of the command. Finally, the content of the user's speech and the average position and orientation from Soundr's machine learning model can be passed to the applications to satisfy the user's needs.

Data collection system

Soundr uses a unique data collection system that collects real human voice data from users wearing a VR headset to provide position and orientation ground truth and guide the user to talk in different positions and at different directions. In this way, we can collect human speaking data along with accurate annotations without intensive and error prone labor. This collection system collects three streams of data: multi-channel audio data from the microphone array, tracking data, and single-channel audio from the user's headset microphone.

The goal of the data collection system is to collect as much audio data as possible while making sure that the dataset has as little bias as possible. Simply asking the user to randomly walk while talking is not ideal since it is likely to create an unevenly distributed dataset and the model trained with it will be biased towards this walking pattern and not being able to generalize across different scenarios.

To avoid those biases, Soundr's data collection workflow has three steps: 1) We generate an evenly distributed set of combinations of positions and orientations before the data collection, 2) We ask the user to wear a VR headset and follow the instruction shown in the VR headset (see Figure 2). The instructions guide the user towards those generated combinations. 3) We collect data from both the headset and a microphone array, align the data streams together, and process them to produce the three streams of data for training.

We generate the position and orientation using systemic sampling. We first divide the space we use for data collection into 36 blocks (6 columns in width and 6 rows in depth) and yaw of head rotation into 4 brackets (0-90°, 90-180°, 180-270°, and 270-360°). For each space block, we randomly generate four points within that block as the target position of the user. Each point is then randomly matched with one yaw rotation bracket and we randomly choose a yaw in that bracket and a pitch between $-60°$ to $60°$. Note that we always ask the user to keep a neutral position in roll since we don't think the model would be able to distinguish different rolls just from the sound.

To minimize the user's time in walking and maximize the user's time in producing usable speech data, we need to connect those generated positions and orientations into a path. We do not want the path to always start from a specific position and end at another position, since the user will get fatigued during the process and we do not want the model to learn that as a factor for position prediction. Therefore, we divide those combinations into four laps around the room. In each lap, we always start from one corner and move down a column with incremental depth, then we go through all the other columns similarly.

Our VR application guides the user to move towards the generated positions and orientations in the four laps. The interface of our VR program is shown in Figure 2. The application shows three triangles in front of the user's viewport in the

Dataset name	*Number of participants (number of coauthors included)*	*Duration (minutes)*	*Environment Description*
Office	9(3)	397	Office space (with constant server noise and other people talking)
Living room	6(0)	228	Shared kitchen / living room space (with regular kitchen noise and occasional people talking in the background)
Conference room 1	3(0)	64	Conference room (with noise of door opening and closing)
Bedroom	2(1)	41	Bedroom (with regular noise of floor creaking while walking)
Conference room 2	1(1)	24	Another conference room

Table 1: Data collected for developing the machine learning model.

shape of a cone (user cone). It will also show a similar cone (target cone) at the target. The cone starts out red, indicating that the user's position and rotation are not yet aligned (Figure 2 left). We instruct the user to align their user cone with the target cone. As the user gets to the target cone, the target cone will turn cyan, indicating that the user has reached the destination. Then a command shows up inside the target cone (Figure 2 right). Each time, one command is selected from our pool of 136 typical voice assistant commands[3]. We instruct the user to read the command at a natural speed and in a natural voice. They then press a button on the VR controller to display the next command. Four commands are displayed at each position (so that the user spends more time speaking instead of moving) and then the system shows the target cone at the next position and orientation.

We collect tracking data from the VR headset in the coordinate system shown in Figure 3. Note that for rotation, we collect rotation quaternions since they are more accurate for data processing. We refer to rotations using Euler angles in this paper for ease of understanding. We process the data collected to generate the three data streams we need for training. For the single-channel audio data collected from the VR headset, we divide it into 0.1 sec segments and pass it into a state-of-the-art model (WebRTC VAD [1]) for voice activity detection and produce a single classification tag of whether there is voice activity during that time. For head tracking data, we remove the roll collected by the headset and set it to 0°. We keep the multi-channel audio data unchanged. We then align these three data streams together to compensate for network latency and droped frames from the VR headset and use this to generate three data streams for training.

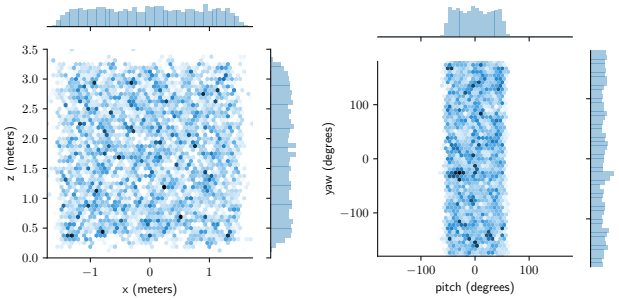
Data collection procedure

To collect training data for Soundr we need to ensure that we have a diverse set of people with different heights and genders so that the system will not be biased towards a specific group of people. As such, we recruited external participants to help us collect training data.

Tasks

At the beginning of each collection session, we asked participants to fill in a demographic questionnaire (gender, age), and we measured their height. Then we asked them to wear a VR headset and follow the instructions in the headset. As described earlier, our data collection program guides the user towards different positions and head orientations through four "laps" composed of 144 total different positions and 576 total spoken commands. We asked our participants to finish two laps first, followed by an optional rest period. After resting, they returned and finished the remaining two laps of the study. The study took about 30 minutes to complete.

[3]A list of our example commands can be found in the Auxiliary Materials.



(a) Hex bin plot showing head position distribution while talking

(b) Hex bin plot showing head orientation distribution while talking

Figure 4: Collected tracking data is evenly distributed.

We performed data collection in five different rooms. In each room, we set up a space of 3.5m (depth) x 3.5m (width), including 0.5m clearance on both depth and width for safety. The data collection system generated evenly distributed targets to guide the user to walk in a 3m x 3m space.

Participants

We recruited 16 participants (9 female), aged 19 to 29 (median 22). The participants' height ranged from 150cm to 189cm with a median of 170cm. Each participant was compensated with a $15 gift card for participation.

Results

The data we collected are shown in Table 1. The five rooms we collected data in included an office, a shared kitchen/living room, a bedroom, and two different conference rooms. Note that for the living room and office dataset, there were other people talking in the space at the same time. The living room dataset also had kitchen appliance noises. Apart from the data we collected from our participants, our co-authors also contributed part of the data (as listed in the table). Each of our participants contributed data only in a single room.

The distribution of the tracking data is shown in Figure 4. As expected from our data collection system, the position

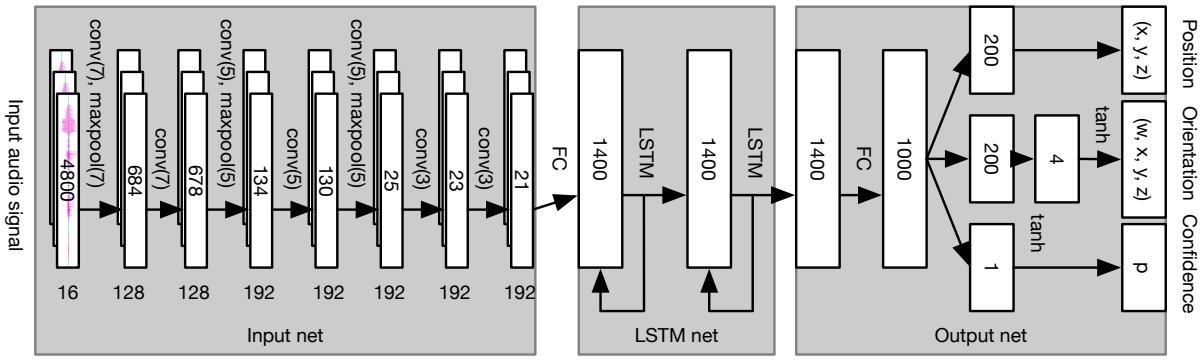


Figure 5: Machine learning architecture of Soundr. LSTM: long short-term memory layer; FC: fully-connected linear layer; conv(n): 1D convolution layer with kernel size of *n*; maxpool(n): 1D max pooling layer with kernel size of *n*; numbers on the white squares: feature size; numbers under the white squares: number of channels. All unmarked arrows are fully-connected linear layers. We use ReLu for activation across all layers. Batch normalizations are added between convolutional layers.

data we collected is evenly distributed across the x and z axis, representing the width and depth of the room. (Data across y axis are not evenly distributed since they correspond to the participants' height.) The orientation data is also evenly distributed across different yaw, and for pitch it is evenly distributed within $(-60^\circ, 60^\circ)$.

Machine learning

The machine learning system is one of the most crucial components of Soundr. It predicts the user's head position and orientation while speaking any voice command. It consumes the data collected from the microphone array and produces three groups of values: position of the speaker (*position*), head orientation of the speaker (*orientation*), and how confident the system is about the position and the orientation (*confidence*).

We first describe the architecture of the model and then list the techniques that we used to train the model.

Model

The machine learning model (shown in Figure 5) is composed of roughly three parts: 1) *Input net*: a convolutional neural network to process raw multi-channel audio and generate higher-level features, 2) *LSTM net*: a long-short term memory (LSTM) network 3) *Output net*: one linear layer followed by three stacks of linear layers dedicated to each one of the three groups of outputs.

The *Input net* is similar to that used in prior work [31]. It accepts raw multi-channel audio samples as input and processes it through a stack of convolutional layers with decreasing kernel sizes and sequence lengths and increasing channel sizes. Compared to prior work, our network is deeper and wider to accommodate our additional output values.

The *LSTM net* addresses one of the limitations of prior research. As mentioned in prior work [31], a longer audio input will result in better prediction results. However, if we simply increase the input length, our model will be harder to train due to its larger size and the model can only output results after a longer time interval, which is not ideal for an interactive application. An LSTM layer is a recurrent neural network that remembers inputs over arbitrary intervals. It produces one output and one hidden output. The hidden output is passed to the same layer of the network as it processes the next audio clip. Therefore, our machine learning model can make the decision not only based on the current audio clip but also audio clips that came before it. So with added LSTM layers, we can keep a reasonable network size and a fast response time, while the network leverages previous audio clips to produce a more accurate and stable result.

Finally, in addition to *position*, our model has more outputs compared with prior work: *orientation* and *confidence*. We added a *confidence* output to make our model produce more robust results from human voice commands. As stated in *Architectural overview*, we computed a weighted average of the *position* and *orientation* results across a voice command with the predicted *confidence* results. This allows us to leverage segments with more distinctive sound and produce better overall result for each voice command.

Training

The hard part of training this network is that the initial confidence reported from the network is inaccurate, so the model may be biased towards showing more accurate results for those audio clips assigned higher confidence initially. Therefore, we train this network with the prediction error of each clip individually and then with the weighted average error of an entire sequence in turn.

We provide the model with a sequence of 20 consecutive 0.1 second audio clips in a batch size of 15 sequences in each training iteration. As stated above, in every 300 iterations, we train to minimize the loss of the position and orientation of individual audio clips compared with their ground truth for 200 iterations and then train to minimize the loss of the weighted average position and orientation of the entire sequence comparing with the average position and orientation of their ground truth. We use the Adam optimizer [12] with a learning rate of 0.00003.

Performance evaluation

Since machine learning models perform best for the environment/user that it was trained on, we evaluate the machine learning model used in Soundr with different configurations

of training and testing data, shown in Table 2. For each configuration of training and testing data, we train the model for 160,000 iterations. For some configurations, we perform a fine-tuning procedure that trains the model for another 20,000 iterations on a selective set of training data. After that, we run the model on the test set and compute the average error.

In the first configuration, we test the best-case scenario for our machine learning model, where we have the resources to create a personalized model and we train for a specific user and the environment. To do this, we randomly selected one male and one female participant from the *office* dataset, one male participant from the *living room* dataset and one female participant from the *conference room* dataset, and used their data from the last lap as testing data. We trained the model with all the other data that we have. Then we fine-tuned each model using the data collected from the same user. The results show that our model can predict position with an average error of 0.31m and orientation with an average error of 34.3°.

In the second configuration, we test the result where we only have data from other users for the room the subject is in. We used the same participants in the first configuration, but we removed all the data from the subject in the training set and used them as test data. We also fine-tuned the model using data collected from the same room. The results show our model can predict position with an average error of 0.33m and orientation with an average error of 40°.

In the third configuration, we test the performance of the model if we have neither data from the same room or from the same user. We removed all the data from the conference room dataset, trained the model using all the other datasets, and test the model on the conference room dataset. We did not fine-tune in this configuration. The results show our model can predict the position with an average error of 0.57m and the orientation with an average error of 57°.

The best prior work on speaker position prediction with a similarly sized microphone array [30] has an average positional error at 0.5m (showing in Table 2) when the same room is used for training and evaluation. Soundr produces a better result when the testing data is collected from the same room (configurations one and two) and can produce comparable results even if tested on a new room without training data. On the orientation side, we could not find any prior work on head orientation prediction with real human data with a single microphone, so we compare with the best prior work [23], which predicts a similar metric with *multiple microphone arrays*. Note that as shown in prior work [8, 28] on position and orientation prediction, a larger microphone array can produce better results. Our results in the first two configurations are slightly worse than the 29.07° average error achieved in that paper, but still better than the baseline condition [4] mentioned in that paper (44.48°) even with only a single microphone array. This shows that our model is better than prior work so that it can process less ideal signals and produce similarly accurate results. We also tried to use only the *Input net* without the *LSTM net*, confidence output, and weighted average (similar to Vera-Diaz et al. [31]) for an ablation study with the *same user, same room* configuration. The ablation study shows that the

Configuration	*Average position error (m)*	*Average orientation error*
Same user, same room	0.31	34.3°
Different user, same room	0.33	40.0°
Different user, different room	0.57	57.0°
Ablation Study	0.75	77.6°
Baseline-1 [30]	0.50	-
*Baseline-2** [23]	-	29.1°
*Baseline-3** [4]	-	44.5°

Table 2: Machine learning results for Soundr
*: requires multiple microphone arrays

Soundr model with our modifications performed much better than a simple CNN on our dataset.

EXAMPLE APPLICATION DESIGN

Current generation smart speakers only support controlling smart home devices using the device's name. With Soundr, a virtual assistant would be able to know more information from the user's request, including the user's head position and orientation. We describe two common requests that can be fulfilled well with Soundr. The first is to directly control appliances with Soundr informing the system which device the user is intending to control (*"Turn on this light"*). The second is to ask the system to fulfill their need using Soundr informing the system which devices are relevant to the user's environment (*"Make the brightness higher in this area."*). We will discuss how we implement those examples with Soundr, and how we evaluate our solution in a user study.

Direct control of smart home appliance

For this type of request, the system uses the user's head position and orientation to figure out the user's references. We think this is useful due to prior work on human-machine dialogues [20], which stated that users are more likely to verbalize the subject, verb, and the object, but would describe locative information in another modality. Soundr is able to allow users to describe the action and type of the object verbally and provide the position of the appliance using head orientation.

To achieve this, we designed an evaluation function to rank each device according to the user's position. The intuition is that the user is more likely to select the device that is closer to the user's position and closer to the user's current head orientation. We define the logical distance of device x, d_x, as

$$d_x = \frac{1-e^2}{r_x(1-e\cos(\theta_x))} \tag{1}$$

where r_x is the Euclidean distance between device x and the user's position, and θ_x is the angular difference between the user's head orientation (as predicted by Soundr) and the direction of device x relative to the user. A lower distance means that this device is close to what the user intends to select.

Figure 6 shows the contour of the positions that will have the same logical distance. The equation forms elliptical contours

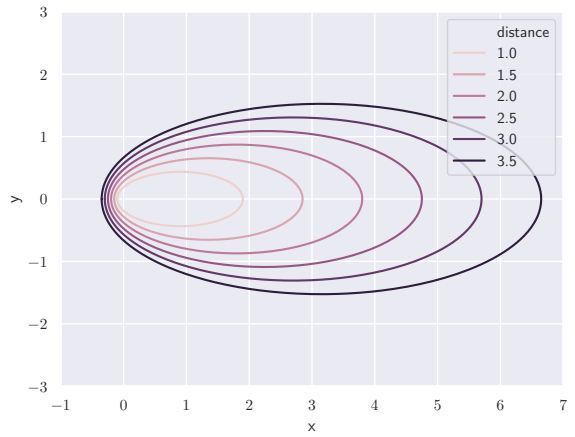


Figure 6: Device distance for direct control. The user is at $(0,0)$ and is facing the x-axis.

that are in front of the user's talking direction. Positions with lower distance are closer to the user's current position and positions with higher distance are farther away from the user's position but still roughly in front of the user. Positions that are far behind the user's talking direction get assigned a very large distance since they are unlikely to be the user's intended target.

We set a hard threshold so devices with a logical distance greater than 4.0 will never get selected. For those devices with distances lower than the threshold, we sort them according to their logical distance in an ascending order $\{x_1, x_2, ..., x_n\}$. Instead of just taking the device with the lowest distance, x_1, we want to ensure we do not select the wrong device when two devices are close in their logical distances. We consider all devices whose logical distance is no greater than $d_{x_1} + 1.0$ to be candidate devices. We then send an "identify"[4] action to all of these devices and ask the user to clarify. The user can specify the device by saying its relative position, such as "the right one", or they can move to a slightly different position and say "this device". With the extra information, Soundr can further confirm the user's intended device to control.

Intention-based control of smart home appliances

Instead of directly operating a single device, in many cases, the user is trying to achieve some goal. For example, the user may want to make an area brighter. He or she can either turn on a few lights that are responsible for lighting up the area (direct control), or just tell the system that he or she wants to *"make the area brighter"* (intention-based control).

To support intention-based control, Soundr not only acquires the 3D positions of a device during configuration, but it also acquires the region that the device may have an effect on, which we call the "effective range". For example, the effective range of a ceiling light may be the region right under the light, and the effective range of a fan may be an area in front of the fan. With this effective range, we can determine if the user's position when they issue the command is within the range of any devices that can help achieve the user's intention. If any devices are found, we can then command the devices to satisfy the user's intention. For example, the user can say *"make the area brighter here"* and Soundr will automatically increase the brightness of the ceiling lights that affect the area in question.

[4]The identify action is commonly supported by IoT devices for notifying the user about a specific device. For example, the identify action on a connected lamp is usually a quick on and off cycle.

Configuration of smart home appliances

To configure a device to work with Soundr, the user needs to let the system know where the device is in 3D space and optionally what is the effective range of the device. To tell the system where the devices are, the user first notifies the system which device he or she intends to configure by either using an app or pressing a physical "configuration" button. Then the user speaks *"this is the device"* from multiple different angles to the device. In this way Soundr will be able to acquire a few positions and directions of the user's configuration command. Let the ith configuration command be represented by a ray with a position vector $\vec{u}_i$ and a unit direction vector $\vec{v}_i$, $1 \leq i \leq n$, where n is the number of commands issued. Let $\vec{p}_x$ be the position of device x. The distance between device x and the ith ray, d_{ix}, is defined as:

$$d_{ix} = \begin{cases} ||\vec{p}_x - (\vec{u}_i + t_i * \vec{v}_i)|| & t_i \geq 0 \\ ||\vec{p}_x - \vec{u}_i|| & t_i < 0 \end{cases} \quad (2)$$

where

$$t_i = \vec{v}_i \cdot (\vec{p}_x - \vec{u}_i) \quad (3)$$

That is, d_{ix} is the distance between device x and the ith ray if the device is in front of the user; otherwise, it is simply the Euclidean distance between the device and the user.

The total distance between the device and all the rays is thus:

$$D(p_x) = \sum_{i=1}^{n} d_{ix} \quad (4)$$

To solve for the unknown device location p_x, we use a Gauss-Newton solver to find p_x that minimizes the total distance with all the rays. In practice, the user only needs to identify a device four times to accurately pinpoint the position of the device.

For defining the effective range of a device, the user also first notifies the system which device he or she wishes to configure and then speaks *"I'm in range"*. A circular region with a diameter of one meter will be created and added to the effective range of the device. The user can do this multiple times to expand the effective range of a device.

Evaluating user preference with Soundr

To evaluate users' subjective preference for Soundr, we conducted a user study and asked users to compare Soundr with two existing baseline methods of controlling IoT appliances. The first baseline method (*basic light*) is to use voice to control devices by name. The second baseline method (*AR light*) is to use a commercial phone-based AR app [2] to control devices by pointing a phone at them. We used the same pipeline for voice recognition in the *basic light* condition and the *Soundr*

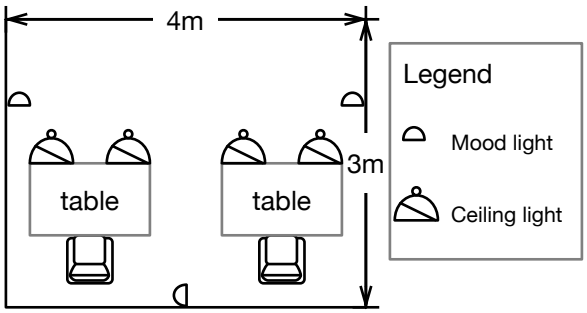


Figure 7: Room layout for user study.

condition to keep the comparison fair. We asked the user to use all three conditions to configure the lights and control the lights according to a few scenarios we gave them.

Setting

To give the user the full experience of configuring and using these methods to control smart home appliances, we used the same office space as was used for data collection. We deployed seven internet-connected lights in the space, as shown in Figure 7. Four of them are ceiling lights and three of them are mood lights. The ceiling lights were hung from the ceiling and the mood lights were placed on the ground. Two tables and two chairs were placed in the room to simulate a shared office space. The participant was asked to sit at the table on the right, and the table on the left was used to simulate a table for another coworker. Two mood lights (right and bottom) are for the user and one mood light (left) is for the coworker. Each pair of ceiling lamps, on the left and on the right, is controlled by the same switch and is thus treated as one unit. This made a total number of five devices (two ceiling light pairs and three mood lights). For *Soundr* all five devices support direct control and the ceiling lights support intention-based control.

Tasks

Our study follows a within-subjects design and consists of three counter-balanced conditions. We first asked each participant to configure the lights using three different methods. To reduce time, we preconfigured 3 of the 5 devices, so the participant only needs to configure one ceiling light and one mood light. For the *basic light* condition, the participant used the Apple Home app to name each light. For the *AR light* condition, the user used the AR app to add the devices into the 3D space. For *Soundr*, the user configured the position of both lights by saying *"this is the light"* four times towards each device for direct control. For the ceiling light, since it supports intention-based control, the user also needs to be in the effective range of the device and say *"I'm in range"*. After each condition, we asked them to fill out a questionnaire about their experience[5].

After configuration, we teach the participant how to control the appliances in all three conditions. For the *basic light* condition, the user can control a device using its name. For example, the user can control a device named "table lamp" by saying *"turn on the table lamp"*. For the *AR light* condition, the participant uses the phone app to point towards the device. The app overlays a slider-like object over every configured smart light, and the user can turn on and off the light by tapping on the slider. Note that the AR app required recalibration every time the app launched. For *Soundr*, the participant can either control each individual light using direct control by saying *"turn on the light"* or they can control the ceiling lights using intention-based control by saying *"make this area brighter"*.

After getting familiarized with the controls, we asked the user to perform a series of tasks according to four different scenarios we designed for all conditions. The first scenario is when the user just arrived at their IoT-equipped office. We asked them to make their work area brighter by turning on the ceiling light. The second scenario is when the user wants to watch a movie. We asked them to make their work area darker by turning off the ceiling light and turning on both of the mood lights in their area. After the movie, they are to turn the ceiling light back on. The third scenario is when the user leaves the office. We asked them to turn off their ceiling lights and their mood lights. The fourth scenario is when the user tries to help their coworkers turn the lights off. We asked them to turn off the ceiling light and the mood light for their coworker. Note that for the *AR light* condition, since the tasks are usually hours apart in the real world ("turn on the light when coming to work" and "turn off the light when leaving work"), the user is likely going to close the app and lock their phone between scenarios. So we asked them to manually close the app between scenarios to simulate the inconvenience of having to relaunch the app every time they wanted to control their IoT devices in the *AR light* condition. After each condition, we asked them to fill out a questionnaire about their experience and a NASA-TLX questionnaire to measure the cognitive load of the task.

Participants

We recruited 12 participants (5 female), and none was in the data collection study, aged 21 to 34 (median 26). Each participant was compensated with a $15 gift card.

Results

As shown in Figure 8, in the Soundr condition, users completed their control task faster than in the *basic light* (paired t-test $p = 6.48 * 10^{-5}$) and in the *AR light* condition (paired t-test $p = 0.03$). The configuration task takes longer in the Soundr condition than in the *AR light* condition ($p = 0.0001$).

From subjective feedback, we find that users think the Soundr condition is better than the *basic light* condition on ease of control ($p = 0.049$). Four users complained that the voice recognition system was inaccurate, which leads to a poorer reception of our technique. One of them also mentioned that the latency of the voice feedback in both the *basic light* condition and the Soundr condition caused frustration for him.

We also gathered other subjective feedback from our participants: Three users mentioned that remembering the name of each device is hard. Three users suggested that we integrate some functionality of Soundr and the *basic light* condition together. For example, user 5 suggested we should allow con-

[5]The questionnaire can be found in the Auxiliary Materials.

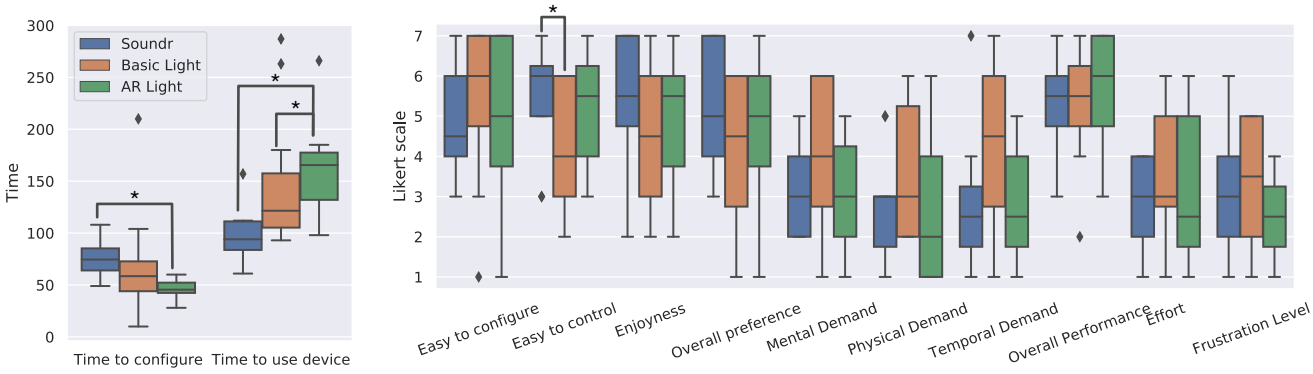


Figure 8: User study result Left: Time comparison between three conditions Right: Questionnaire result (last five questions are from NASA-TLX) *: statistically significant ($p<0.05$)

trolling a device by its name and by talking towards it so that the user can make the choice depending on the situation.

Overall, we observed advantages of Soundr in task completion time and user's perception of ease of control and a slight disadvantage on time for configuration. We also learned that our current implementation of Soundr has some problems with voice recognition, but that is out of the scope of this paper.

ADDITIONAL APPLICATIONS SCENARIOS

In our *example application design* section, we used Soundr to configure and control IoT devices. There are many other possible applications, as discussed below.

IoT device query, control, and configuration

Soundr can be used for many other scenarios similar to an AR-based IoT control system [32]. Soundr can be used to query the status of IoT devices. For example, *"What is the IP address of this printer?"* in a printing room with many printers, or *"Call customer support for this washing machine."* in a laundry room. Soundr can also be used to configure relationships between IoT devices. Such as, *"Control this light using this smart button"*; *"When the air quality is bad, turn this light red."* Soundr can also be used to provide context for device control (e.g., ''*Clean up this area, Roomba.*").

Meeting Assistant

Prior work on speaker diarization studied using a microphone array to predict speaker position and segment the transcription of multiple speakers [33]. Since Soundr provides better position prediction than prior work, it can be used to further enhance the accuracy of speaker diarization. Furthermore, Soundr can also provide speaker orientation, which is previously infeasible with a single microphone array. This information can even help figure out the spatial references made during the meeting, such as *"I'll send you the files afterward"*. The speaker position and orientation can be used to figure out who "you" is referring to in the last example, and the assistant can automatically send the files to the correct person.

Indoor navigation

Soundr can also be used in hands-free indoor navigation. Usually, indoor navigation requires the user to carry a specific kind of electronics device or requires video recording throughout the building, which may not be acceptable to the occupants for privacy reasons. With Soundr and a set of smart speakers deployed within a building, we can allow any person in the building to ask for directions verbally. Soundr can be used to figure out the exact location and orientation of that user and provide specific navigation instructions, such as *"Turn left. Walk 10 steps and the room will be on your left."*

LIMITATIONS AND FUTURE WORK

Although we have tested the system in more environments than prior work [26, 27, 31], more testing is needed to generalize to different environments. Also, Soundr did not perform as well if it has not been trained for a given environment or user. This means that if we want to deploy Soundr today, we need to calibrate a room with a VR headset when Soundr is installed. Note that a VR headset is unnecessary to use Soundr. By collecting sufficient data across environments and users, it may be possible in the future to provide a pre-trained model that works across environments so no calibration is required in deployment.

Currently, Soundr only uses the user's position and orientation to figure out the references. Future systems can leverage the referential terms that the user uses to further improve the result. For example, when the user says *"Turn on this ceiling light"*, we can ignore any light that is not a ceiling light.

CONCLUSION

Soundr shows a promising future where smart speakers can better serve users' requests by knowing their head position and orientation. Soundr can achieve this vision using off-the-shelf smart speakers; no additional hardware is needed. By making this technology more accurate and more versatile, future buildings can be less static and more responsive to users' needs.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Tianshi Li for her help on the early prototypes and writing. Also, the authors would like to thank Prof. Cheng Zhang for feedback on the early project idea. Finally, the authors would also like to thank the reviewers for their constructive feedback.

This work is supported in part by SK Telecom and the National Science Foundation under Grant No. 1900638.

REFERENCES

[1] 2018. WebRTC Home | WebRTC. https://webrtc.org/. (2018). (Accessed on 09/19/2019).

[2] 2019. Smart AR Home on the App Store. https://apps.apple.com/us/app/smart-ar-home/id1344696207. (2019). (Accessed on 09/19/2019).

[3] 2019. U.S. Smart Speaker Ownership Rises 40% in 2018 to 66.4 Million and Amazon Echo Maintains Market Share Lead Says New Report from Voicebot - Voicebot. https://voicebot.ai/2019/03/07/u-s-smart-speaker-ownership-rises-40-in-2018-to-66-4-million-and-amazon-echo-maintains-market-share-lead-says-new-report-from-voicebot/. (2019). (Accessed on 09/18/2019).

[4] Alberto Abad, Carlos Segura, Climent Nadeu, and Javier Hernando. 2007. Audio-based approaches to head orientation estimation in a smart-room. In *Eighth Annual Conference of the International Speech Communication Association*. 590–593.

[5] Michael S. Brandstein and Harvey F. Silverman. 1997. A practical methodology for speech source localization with microphone arrays. *Computer Speech & Language* 11, 2 (apr 1997), 91–126. DOI: http://dx.doi.org/10.1006/csla.1996.0024

[6] Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. 2005. Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays. In *Ninth European Conference on Speech Communication and Technology*. 2337–2340.

[7] Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. 2008. Comparison Between Different Sound Source Localization Techniques Based on a Real Data Collection. In *2008 Hands-Free Speech Communication and Microphone Arrays*. IEEE. DOI: http://dx.doi.org/10.1109/hscma.2008.4538690

[8] Joseph H. DiBiase, Harvey F. Silverman, and Michael S. Brandstein. 2001. Robust Localization in Reverberant Rooms. In *Digital Signal Processing*. Springer Berlin Heidelberg, 157–180. DOI: http://dx.doi.org/10.1007/978-3-662-04619-7_8

[9] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (Apr 2017), 677–691. DOI: http://dx.doi.org/10.1109/tpami.2016.2599174

[10] Eiichi Ito. 2001. Multi-modal Interface with Voice and Head Tracking for Multiple Home Appliances. In *INTERACT*. 727–728.

[11] Vikram Jeet, Hardeep Singh Dhillon, and Sandeep Bhatia. 2015. Radio Frequency Home Appliance Control Based on Head Tracking and Voice Control for Disabled Person. In *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE. DOI:http://dx.doi.org/10.1109/csnt.2015.189

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

[13] Guillaume Lathoud, Jean-Marc Odobez, and Daniel Gatica-Perez. 2005. AV16.3: An Audio-Visual Corpus for Speaker Localization and Tracking. In *Machine Learning for Multimodal Interaction*. Springer Berlin Heidelberg, 182–195. DOI: http://dx.doi.org/10.1007/978-3-540-30568-2_16

[14] Rainer Malkewitz. 1998. Head pointing and speech control as a hands-free interface to desktop computing. In *Proceedings of the third international ACM conference on Assistive technologies - Assets '98*. ACM Press. DOI:http://dx.doi.org/10.1145/274497.274531

[15] Menno Müller, Steven van de Par, and Joerg Bitzer. 2016. Head-Orientation-Based Device Selection: Are You Talking to Me?. In *Speech Communication; 12. ITG Symposium*. VDE, 1–5.

[16] Hirofumi Nakajima, Keiko Kikuchi, Toru Daigo, Yutaka Kaneda, Kazuhiro Nakadai, and Yuji Hasegawa. 2009. Real-time sound source orientation estimation using a 96 channel microphone array. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. DOI: http://dx.doi.org/10.1109/iros.2009.5354285

[17] Alberto Yoshihiro Nakano, Seiichi Nakagawa, and Kazumasa Yamamoto. 2009. Automatic estimation of position and orientation of an acoustic source by a microphone array network. *The Journal of the Acoustical Society of America* 126, 6 (dec 2009), 3084–3094. DOI:http://dx.doi.org/10.1121/1.3257548

[18] Alberto Yoshihiro Nakano, Seiichi Nakagawa, and Kazumasa Yamamoto. 2010. Auditory perception versus automatic estimation of location and orientation of an acoustic source in a real environment. *Acoustical Science and Technology* 31, 5 (2010), 309–319. DOI: http://dx.doi.org/10.1250/ast.31.309

[19] Leonardo O. Nunes, Wallace A. Martins, Markus V. S. Lima, Luiz W. P. Biscainho, Mauriacio V. M. Costa, Felipe M. Goncalves, Amir Said, and Bowon Lee. 2014. A Steered-Response Power Algorithm Employing Hierarchical Search for Acoustic Source Localization Using Microphone Arrays. *IEEE Transactions on Signal Processing* 62, 19 (oct 2014), 5171–5183. DOI: http://dx.doi.org/10.1109/tsp.2014.2336636

[20] Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97* (1997). DOI: http://dx.doi.org/10.1145/258549.258821

[21] Andrey Ronzhin and Alexey Karpov. 2005. Assistive multimodal system based on speech recognition and head tracking. In *2005 13th European Signal Processing Conference*. IEEE, 1–4.

[22] R. Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (mar 1986), 276–280. DOI: http://dx.doi.org/10.1109/tap.1986.1143830

[23] Carlos Segura, Alberto Abad, Javier Hernando, and Climent Nadeu. 2008. Speaker orientation estimation based on hybridation of GCC-PHAT and HLBR. In *Ninth Annual Conference of the International Speech Communication Association*.

[24] C. Segura, C. Canton-Ferrer, A. Abad, J.R. Casas, and J. Hernando. 2007. Multimodal Head Orientation Towards Attention Tracking in Smartrooms. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. IEEE. DOI: http://dx.doi.org/10.1109/icassp.2007.366327

[25] Petre Stoica and Jian Li. 2006. Lecture Notes - Source Localization from Range-Difference Measurements. *IEEE Signal Processing Magazine* 23, 6 (nov 2006), 63–66. DOI: http://dx.doi.org/10.1109/sp-m.2006.248717

[26] Yingxiang Sun, Jiajia Chen, Chau Yuen, and Susanto Rahardja. 2017. Indoor sound source localization with probabilistic neural network. *IEEE Transactions on Industrial Electronics* 65, 8 (2017), 6403–6413.

[27] Dmitry Suvorov, Ge Dong, and Roman Zhukov. 2018. Deep residual network for sound source localization in the time domain. *arXiv preprint arXiv:1808.06429* (2018).

[28] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. 2011. Single-channel head orientation estimation based on discrimination of acoustic transfer function. In *Twelfth Annual Conference of the International Speech Communication Association*.

[29] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. 2012. Estimation of talker's head orientation based on discrimination of the shape of cross-power spectrum phase coefficients. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[30] Jose Velasco, Daniel Pizarro, and Javier Macias-Guarasa. 2012. Source Localization with Acoustic Sensor Arrays Using Generative Model Based Fitting with Sparse Constraints. *Sensors* 12, 10 (Oct 2012), 13781–13812. DOI:http://dx.doi.org/10.3390/s121013781

[31] Juan Vera-Diaz, Daniel Pizarro, and Javier Macias-Guarasa. 2018. Towards End-to-End Acoustic Localization Using Deep Learning: From Audio Signals to Source Position Coordinates. *Sensors* 18, 10 (Oct 2018), 3418. DOI:http://dx.doi.org/10.3390/s18103418

[32] Jackie (Junrui) Yang and James A. Landay. 2019. InfoLED: Augmenting LED Indicator Lights for Device Positioning and Communication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology - UIST '19*. ACM Press. DOI: http://dx.doi.org/10.1145/3332165.3347954

[33] Takuya Yoshioka, Zhuo Chen, Dimitrios Dimitriadis, William Hinthorn, Xuedong Huang, Andreas Stolcke, and Michael Zeng. 2019. Meeting Transcription Using Virtual Microphone Arrays. *arXiv preprint arXiv:1905.02545* (2019).