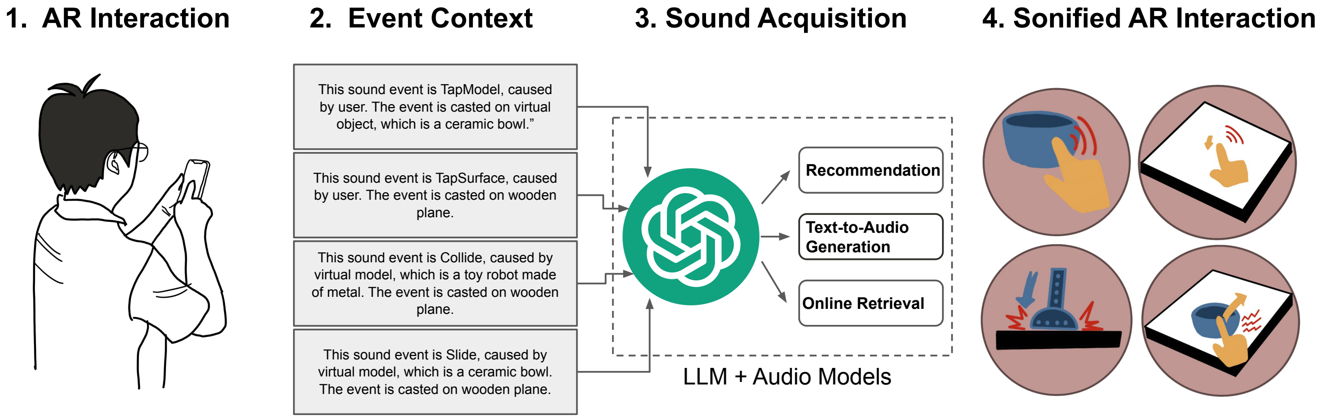
SonifyAR: Context-Aware Sound Effect Generation in Augmented Reality

Xia Su
University of Washington
Seattle, Washington, USA
xiasu@cs.washington.edu

Eunyee Koh
Adobe Research
San Jose, California, USA
eunyee@adobe.com

Chang Xiao
Adobe Research
San Jose, California, USA
cxiao@adobe.com



This sound event is TapModel, caused by user. The event is casted on virtual object, which is a ceramic bowl.”

This sound event is TapSurface, caused by user. The event is casted on wooden plane.

This sound event is Collide, caused by virtual model, which is a toy robot made of metal. The event is casted on wooden plane.

This sound event is Slide, caused by virtual model, which is a ceramic bowl. The event is casted on wooden plane.

Figure 1: SonifyAR is a Programming by Demonstration AR sound authoring pipeline. With SonifyAR, the user can first (1) conduct AR interactions, while SonifyAR (2) collects context information as text. (3) The context text is then processed by an LLM, which guides the sound acquisition processes with multiple models and methods. (4) The acquired sounds are then used to sonify AR events.

ABSTRACT

Sound plays crucial roles in enhancing user experience and immersiveness in Augmented Reality (AR). However, current AR authoring platforms lack support for creating sound effects that harmonize with both the virtual and the real-world contexts. In this work, we present SonifyAR, a novel system for generating context-aware sound effects in AR experiences. SonifyAR implements a Programming by Demonstration (PbD) AR authoring pipeline. We utilize computer vision models and a large language model (LLM) to generate text descriptions that incorporate context information of user, virtual object and real world environment. This context information is then used to acquire sound effects with recommendation, generation, and retrieval methods. The acquired sound effects can be tested and assigned to AR events. Our user interface also provides the flexibility to allow users to iteratively explore and fine-tune the sound effects. We conducted a preliminary user study to demonstrate the effectiveness and usability of our system.

CCS CONCEPTS

• **Human-centered computing** → **Interaction design**; **Interactive systems and tools**.

KEYWORDS

Mixed Reality, Sound, Augmented Reality, Authoring Tool

ACM Reference Format:
Xia Su, Eunyee Koh, and Chang Xiao. 2024. SonifyAR: Context-Aware Sound Effect Generation in Augmented Reality. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3613905.3650927

1 INTRODUCTION

The role of sound in Augmented Reality (AR) has grown profoundly, becoming a pivotal element that adds depth and immersion to user experiences. Past research has shown positive effects of AR sound in many aspects, such as depth perception and task completion [31], searching, and navigation [19] and even assistance for visually impaired people [18].

Despite the evident impact of sound in AR, its integration—particularly in the scope of sound authoring—remains inadequately addressed. Current AR authoring platforms like Reality Composer [8], Adobe Aero [5], and Unity [22] provide only rudimentary sound authoring functionalities. Central to these platforms is the Programming by Specification (PbS) methodology [16], where users define

CHI EA '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0331-7/24/05
https://doi.org/10.1145/3613905.3650927

AR sound contents and play conditions during the design phase, which is often referred to as the "trigger-action mechanism" [16]. This allows creators to define, with precision, the conditions for triggers and the consequent sound effects. Unfortunately, these current authoring workflows present three critical shortcomings in terms of AR sound authoring:

(1) **Limited sound interaction space.** Since triggers of action in current authoring pipelines are typically bound to virtual objects, sound effects for interactions between AR objects and the real-world environment (*e.g.*, a virtual ceramic cup clinking against a wooden table) or between the user and the real-world via the AR interface (e.g., a user tapping on a wall through AR interface) are neglected.
(2) **Lacking real-world context.** As authoring happens before AR experiences, key context information can be missing for creating matching sound effects. For instance, authoring the sound of a virtual ball dropping on real world surfaces is challenging since there is no contextual information regarding the surface material.
(3) **Limited sound source.** Given the limited availability of sound assets in the default library that comes with authoring platform and the scarcity of suitable sound resources online, AR authors struggle to find fitting sounds for distinct AR events. Examples include reproducing the the wing flutter sound of a virtual dragonfly or simulating footsteps of a virtual robot traversing diverse surfaces like wood, carpet, or glass.

To address these challenges faced by current AR sound authoring pipelines, we present *SonifyAR*: a context-aware AR sound authoring system using generative models. SonifyAR harnesses the advancements of Large Language Models (LLMs) and sound acquisition models to recommend, retrieve, or generate context-aware sound effects in response to AR events. Inspired by prior works like Rapido [14], PoseVec [28], and GestureAR [24], our approach adopts a Programming by Demonstration (PbD) [13] pipeline that enables users to discover and author context-aware AR sound during AR interaction. Our authoring system identifies potential sound-producing AR events and utilize LLM and sound acquisition models to generate context-aware sound effects in real time. To our knowledge, we are the first system to offer automatic, in-context sound authoring for AR.

The contributions of this paper are summarized as follows: First, we explored the landscape of sound authoring in AR, identifying a noticeable gap between the existing authoring tools and the broader potential opportunities in the design space. Second, to address this gap, we introduce a PbD AR authoring mechanism that can detect potential sound-producing user interactions in the AR environment and process the context information including the environmental information, virtual object information and user action information. Lastly, we implemented a generative-model-based sound authoring system that can automatically acquire appropriate sound effects with the compiled contextual information.

2 RELATED WORK

Prior investigations demonstrate that sound is crucial in AR experiences by directing the user's attention, enhancing the sense of presence, and creating interactive time-varying experiences [20]. Nonetheless, the current landscape of AR authoring tools and platforms reveals a deficiency in adequately supporting AR sound authorship. Tools with lower fidelity, like Halo AR [3] and ARVid [1], empower users to overlay virtual imagery, videos, and 3D assets onto the real world. In these tools, sound elements are typically integrated directly into virtual assets (e.g. videos with sound or virtual animals that roar). Mid-fidelity tools like Adobe Aero [5] and Apple's Reality Composer [8] provide trigger-action interactions, where sound elements can be selected and edited through user-friendly interfaces and linked to triggers like user taps or proximity events. High-fidelity tools, such as Unity [22] and Unreal [10], offer both visual editors and coding capabilities. They provide substantial flexibility in designing AR sound interactions, including the ability to set parameters like sound decay and code-specific conditions for activating sound assets. While most AR authoring tools support sound editing, two significant gaps remain: creators are often tasked with sourcing suitable sound effects, and the range of events that can be associated with sounds is limited. Our proposed system aims to address these issues.

Many methods can be used to retrieve, synthesize or generate sound for AR experiences. For example, Oncescu et al. [17] trained cross-modal embedding models to support text-to-audio retrieval. Multi-modality models like VAST [9] also supports text-to-audio retrieval that could potentially be used to provide sound assets for AR experiences. Similarly, FreeSound [2] implemented an online API that retrieves sound effects from dataset of professional assets with text. Besides retrieval methods, text-to-audio generation models [12, 15, 26] also emerge with promising performance and could be used to acquire sound for niche cases of text description. In general, retrieval methods can provide more realistic sound assets but do not ensure coverage for all text descriptions. While generative models might not achieve same sound quality, they can produce results for any text input. In SonifyAR, we adopt both the retrieval method and the text-to-audio generation model to utilize their complementary strength. Specifically, we use the FreeSound API for its extensive library of professional sound asset and AudioLDM [15] for its state-of-the-art performance on sound generation.

3 THE AUDITORY DESIGN SPACE OF AR

Inspired by Jain et al. [11] on VR sound taxonomy, we conduct a comprehensive analysis of the broader design space for AR sound interaction, focusing on the three key components of AR: user, virtuality and reality.

A typical AR interaction involves **the user**, representing the individual experiencing or interacting with the AR; **virtuality**, referring to the virtual object(s) placed in the AR experience; and **reality**, denoting the real world physical environment. Between these three components, we list potential sound-producing interactions that can occur in an AR session.

Virtuality: Sounds can generate from the internal activities of a virtual object. This could be bound to an object appearing or disappearing (*e.g.* a notifying sound when a virtual object show up), or from an object's animated behavior (*e.g.* the mechanical noise made by a moving robot or a virtual dinosaur roars).

User-Virtuality: When a user interacts with a virtual object, for instance tapping on a virtual dog, a corresponding sound, such as a bark, could be generated as feedback.

User-Reality: This type of event involves users interacting with the physical environment through their AR devices, even in the absence of AR objects. For instance, a user tapping on a real-world surface, such as a wooden table, as displayed on their phone screen, can generate sound feedback, thereby enhancing the interaction.

Virtuality-Reality: This aspect underscores a significant gap in many existing AR authoring systems, namely, the interaction between virtual objects and the real environment. For instance, if a virtual metallic robot walks on a real glass table, the resulting interaction should produce a distinct stomping sound, characterizing the collision between metal and glass.

User-Virtuality-Reality: When a user-initiated action involves both virtual object and the real world environment, sound can be provided as feedback to validate this action and improve immersiveness. For example, a user dragging and sliding a virtual cup on a real table surface could produce a sound mimicking the interaction.

Among the listed sound space, existing AR authoring tools, such as Reality Composer [8] and Adobe Aero [5], offer robust support for sound feedback in the domain of virtuality and user-virtuality interactions. However, the dimensions of virtuality-reality and user-reality are just as vital, if not more so, since they extend immersiveness beyond a purely digital domain. Yet, efficient methods for authoring sound feedback in these dimensions are conspicuously absent. In the subsequent section, we will discuss how SonifyAR is designed to tackle these dimensions, encapsulate all listed sound interactions within one single framework, and streamline AR sound authoring with LLM and generative models.

4 SONIFYAR: A CONTEXT-AWARE SOUND AUTHORING SYSTEM FOR AR

Drawing insights from several prior AR authoring research [14, 28], SonifyAR adopts a Programming by Demonstration (PbD) sound authoring framework. During the authoring process, users freely interact with the AR environment. Simultaneously, SonifyAR collects potential sound-producing interaction events, along with their context information such as detected planes and materials (Figure 2 left). The events and their context information are textualized and input into the LLM, which serves as a controller for our three sound acquisition methods: recommendation, retrieval, and generation. The LLM's response is interpreted and executed by models and pipelines (Figure 2 middle). Users can review, select, and iterate the sound acquisition results using a call-out panel, and all user-selected sounds will be instantly applied in AR (Figure 2 right).

4.1 Event Textualization

We use text as a unified format to incorporate all context information that SonifyAR collects for AR events, including event type, (*e.g.* tapping an object or dragging a virtual item), action source (*e.g.* user or virtual object), and action target (*e.g.* virtual object or real-world plane). We also enrich the context with more detailed descriptions of the involved components in AR events. All information forms a single text snippet following a predefined text template. One example of such context description is shown in the left part of Figure 2.

Event Types As an initial work, we identify six types of sound-related AR events that could potentially trigger sound feedback.

1. Tap Real Word Structure: A user taps on a real-world structure through the AR interface.
2. Slide: A user holds the virtual object and slide it on a real-world surface.
3. Collide: A virtual object collides with a real-world surface.
4. Show Up: A virtual object shows up in AR experience.
5. Tap Virtual Objects: A user taps on virtual object through the AR interface.
6. Play Animation: A virtual object plays an animation.

Scene Context Understanding When the AR experience starts, SonifyAR utilizes ARKit [6]'s plane detection functionality [7] to understand the surrounding environment. We also employ the Deep Material Segmentation model [23] to identify the material of planes (Figure 2 left). These material labels can help produce appropriate sound effects when either the AR object or the user interacts with real-world planes.

Virtual Object Understanding To ensure that sound assets are aligned with the material and state of virtual objects, we collect text descriptions for all virtual objects (*e.g., 'This model is a toy robot made of metal.'*) and their animations (*e.g., 'A toy robot walks.'*). These descriptions are typically provided by the asset creator, but if additional details are needed, we prompt users to contribute the relevant information.

Text Template By consolidating all context information into a single text snippet, we employ a unified template: *"This event is [Event Type], caused by [Source]. This event casts on [Target Object]. [Additional Information on Involved Entities]."* Note that not all events have a target object or additional information.

4.2 Sound Authoring

SonifyAR utilizes the LLM to acquire context-matching sound assets of an AR event. Inspired by HuggingGPT[21] and Visual ChatGPT [25], we utilize LLM as a controller of multiple sound acquisition methods. The LLM takes the textualized context information as input, and reply with commands for sound authoring pipelines (Figure 3). In our current stage, we support three major sound authoring methods: recommendation, retrieval, and generation.

4.2.1 *Sound Recommendation.*

LLM can recommend sound assets based on context information text. Similar to other AR authoring tools, we collect a set of sound effects from Adobe Audition sound effects [4], each labeled with a descriptive file name, like "Crash Aluminum Tray Bang" or "Liquid Mud Suction". The entire list of sound file names is provided to LLM as system context. When given a text description of an event, LLM selects and send back the top five most suitable sound effects from this list. SonifyAR would then parse the filename and add the top corresponding sound as one of the sound options for event. An example reply from LLM regarding the recommendation process is shown in Figure 3, under the "Recommend" section.

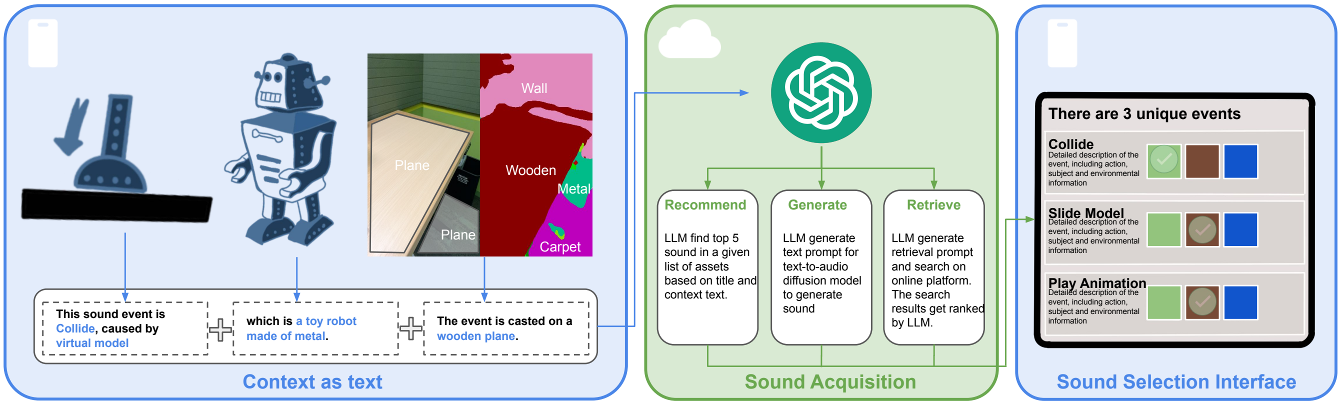

Plane
Wall
Wooden
Metal
Plane
Carpet
This sound event is Collide, caused by virtual model
which is a toy robot made of metal.
The event is casted on a wooden plane.
Context as text
Recommend
LLM find top 5 sound in a given list of assets based on title and context text.
Generate
LLM generate text prompt for text-to-audio diffusion model to generate sound
Retrieve
LLM generate retrieval prompt and search on online platform. The search results get ranked by LLM.
Sound Acquisition
There are 3 unique events
Collide
Detailed description of the event, including action, subject and environmental information
Slide Model
Detailed description of the event, including action, subject and environmental information
Play Animation
Detailed description of the event, including action, subject and environmental information
Sound Selection Interface

Figure 2: Pipeline of SonifyAR

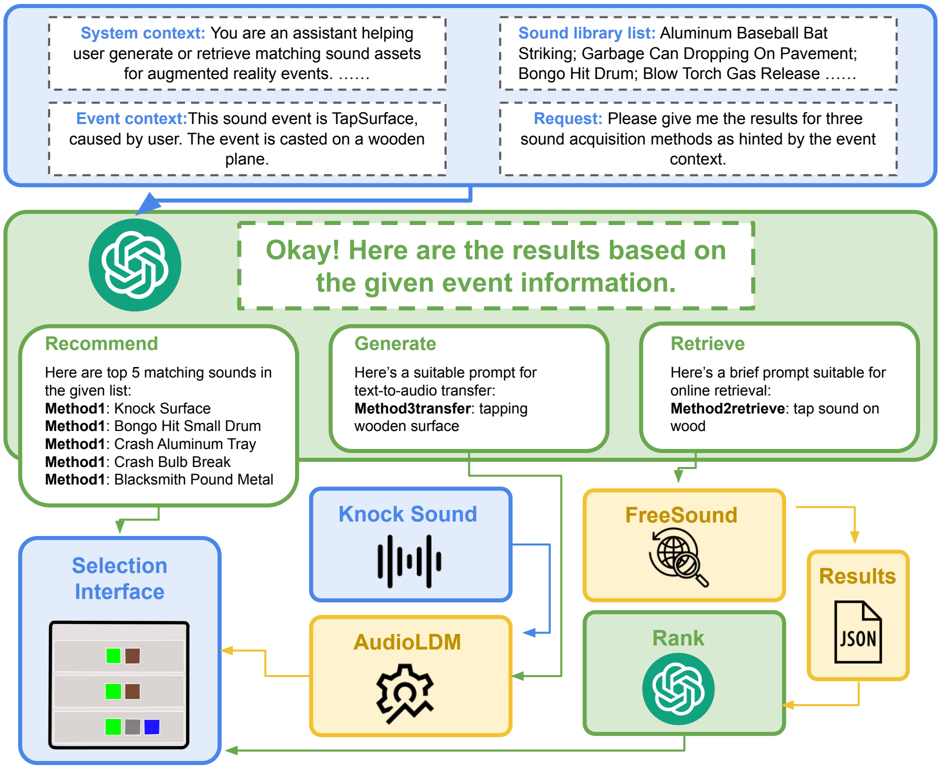

System context: You are an assistant helping user generate or retrieve matching sound assets for augmented reality events. ……
Sound library list: Aluminum Baseball Bat Striking; Garbage Can Dropping On Pavement; Bongo Hit Drum; Blow Torch Gas Release ……
Event context: This sound event is TapSurface, caused by user. The event is casted on a wooden plane.
Request: Please give me the results for three sound acquisition methods as hinted by the event context.
Okay! Here are the results based on the given event information.
Recommend
Here are top 5 matching sounds in the given list:
Method1: Knock Surface
Method1: Bongo Hit Small Drum
Method1: Crash Aluminum Tray
Method1: Crash Bulb Break
Method1: Blacksmith Pound Metal
Generate
Here's a suitable prompt for text-to-audio transfer:
Method3transfer: tapping wooden surface
Retrieve
Here's a brief prompt suitable for online retrieval:
Method2retrieve: tap sound on wood
Knock Sound
FreeSound
Results
JSON
Selection Interface
AudioLDM
Rank

Figure 3: SonifyAR's sound acquisition pipeline. "Method1" is short for "Method1recommend".

4.2.2 Sound Retrieval. Since a finite collection of sound assets may not always contain the desired sound effect, we expand our sound authoring capability with online retrieval. We select the FreeSound [2] as the searching database. We leverage the LLM to compress the entire event description into a shortened searching prompt, and search with such prompt using the FreeSound API. The returned results, provided as JSON strings, contain information on all search hits. The LLM then selects the top five results. The highest-ranking result is downloaded and presented to the user. Others in the top five alternatives are also stored in our system for users who wish to explore more options. This retrieval process is shown in the bottom right of Figure 3.

4.2.3 Sound Generation. We adopt text-to-audio generation model (AudioLDM [15] in our implementation) to complement the previous approaches. We ask the LLM to compress the event text description into a shortened generation prompt. Upon receiving such command from LLM, SonifyAR will send the prompt to the AudioLDM model, requesting text-to-sound generation. This newly generated sound are then stored in our system for user exploration at a later time. The audio diffusion model can also conduct text-based style transfer which we utilize to tweak sound assets. Specifically, when generating sound effects for tapping, sliding or colliding events, we will pass a default sound effect and request a style transfer operation with text prompt (see "Generate" in Figure 3). This pipeline could let the output sound has similar length and rhythm of the input sound, ensuring a good time match between sound and AR events. Also, this transfer ability can be used to modify sound assets with text input or generate similar sounds, enabling user to iterate on the sound assets.

All the sound authoring processes, including recommendation, generation, and retrieval, run in parallel on the backend. This prevents interference on the user's ongoing AR experience.

4.3 User Interface

As a PbD authoring framework, the SonifyAR application initiate directly with an interactive AR experience (Figure 4a left). Users can freely interact with the scene, performing actions like moving virtual objects or interacting with the real-world surfaces. When an AR event is detected, a text label appears on the top-left screen to inform the user (Figure 4a.A), and the sound acquisition component is activated to generate candidate sound effects for the detected events.

By clicking a button (Figure 4a.D), users can access an editing interface. The interface includes all unique AR events detected (Figure 4a.H) and the acquired sound effects (Figure 4a.I). Users can click on a sound effect to preview it and double click to select and activate it for the AR session. User can also long press a sound asset to call out a menu with a suite of exploratory options, like "style transfer" and a "generate similar sound". After operations, users can minimize the authoring panel (Figure 4a.G) to resume the AR experiences and test the selected sound. Users can always access the editing interface to modify their choices if necessary.

4.4 Example Usage Scenarios

E-commerce AR is widely adopted as presentation medium in e-commerce sites like Amazon and IKEA. We envision SonifyAR further enhancing the e-commerce AR experiences by adding material-aware sounds. For example, SonifyAR could make a virtual ceramic cup chime when touched or moved, enhancing media novelty and immersion, which, as demonstrated in prior experiments [27], can boost purchase intentions.

Accessibility Existing research [18, 29, 30] explored assisting blind or low vision (BLV) people with AR in two modalities: visual highlights and audio hints. SonifyAR is in alignment with the latter approach by simplifying the sound authoring process. Additionally, SonifyAR enables new AR sound interaction-the user-reality sound interaction (see section 3)-that can help BLV people explore the real world space through AR interfaces.

Plug-in to existing platforms SonifyAR could be integrated into established AR authoring tools. Given that the authoring pipeline of existing tools precedes the AR experiences, while our PbD (Programming by Demonstration) pipeline operates concurrently during these experiences, these two methods occupy distinct time segments and, as such, can complement each other effectively.

5 PRELIMINARY EVALUATION

To examine the usability and authoring performance of the SonifyAR system, we conducted a preliminary usability study across 8 participants (6 male and 2 female) age between 26 and 33. Among them, six had prior AR experience, and three had experience in AR authoring tools. Each participant operates the SonifyAR application with a given 3D model, trying to author matching sounds to its AR experiences. After usage, participants provided ratings for several questions about the system, and also elaborated on their general feedback and reasonings behind ratings.

All participants were able to complete the AR sound authoring task using SonifyAR without difficulty. Feedback on the SonifyAR app was also largely positive. All eight participants expressed a favorable impression of the tool. They were highly agreed that the SonifyAR tool would be helpful to AR authoring process (Q1,AVG=6, SD=0.93). There was also a general willingness for using SonifyAR in their own AR creation practice (Q2, AVG=6.25, SD=1.16). Participants agreed that the sound interaction involving the real world surfaces will improve the immersiveness of AR experiences (Q3, AVG=6.38, SD=1.06). Additionally, they preferred the automated sound authoring process over the manual search for sound assets (Q5, AVG=6.13, SD=0.99). The only area for improvement was the quality of the generated sound (Q4, AVG=4.75, SD=1.39). However, this could be enhanced with the introduction of a more advanced sound generative model, which we could readily integrate into our system. The full Likert-type results can be viewed in Figure 4b.

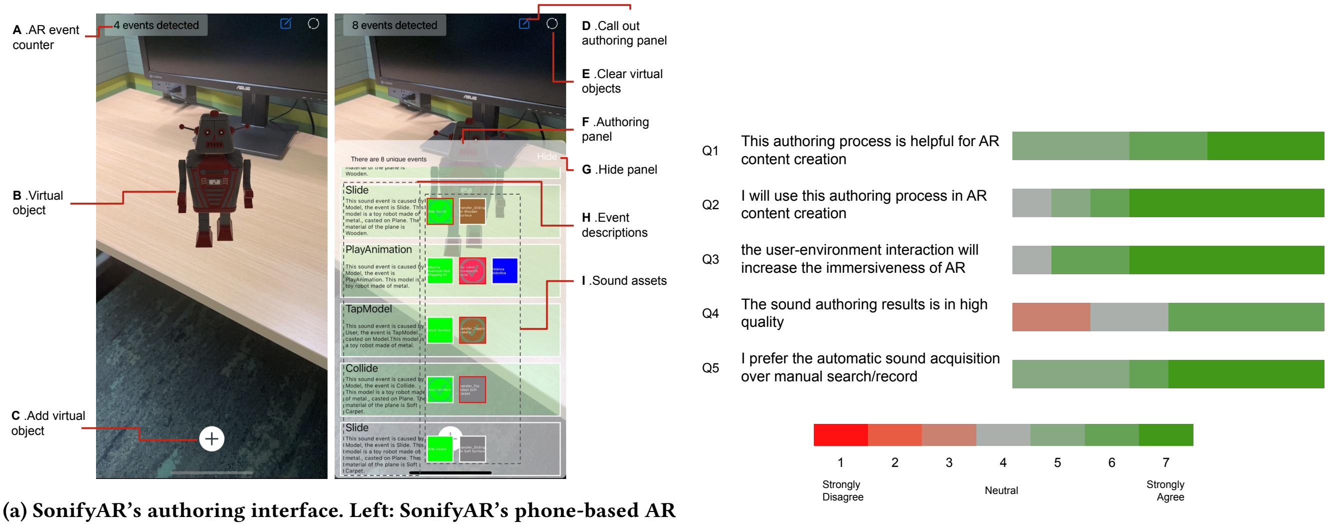
6 FUTURE ADVANCEMENTS

While SonifyAR enables automated AR sound authoring, there remains room for further advancements.

Firstly, our current SonifyAR app is a simplified tool compared to comprehensive AR design suites that cover every aspect of AR scene creation. It exclusively supports the authoring of AR sound and accepts pre-crafted AR scenes as input. For wider applicability that reflects the real-world intricacies of AR content creation, we recognize the need to extend its capabilities to accommodate a wider spectrum of AR events and interactions. Our future goals include integrating the SonifyAR framework with established tools like Reality Composer and Adobe Aero.

Secondly, SonifyAR adopts a straightforward process of collecting context information for virtual objects by using a text template to compile semantics. In future developments, we aspire to incorporate computer vision models for object detection and image understanding to enhance the context acquisition process with greater precision and adaptability.

Lastly, the current sound generation output of SonifyAR presents opportunities for further refinement. Recognizing the rapid advancements in the AI research domain, our system has been designed as a modular framework, ensuring that as more advanced models emerge, they can be seamlessly integrated. Every LLM/ CV/ generative model within SonifyAR is replaceable, allowing SonifyAR to easily adapt and stay at the forefront of innovation in this field.


A .AR event counter
4 events detected
8 events detected
B .Virtual object
C .Add virtual object
D .Call out authoring panel
E .Clear virtual objects
F .Authoring panel
G .Hide panel
H .Event descriptions
I .Sound assets
There are 8 unique events
Hide
Slide
PlayAnimation
TapModel
Collide
Slide
Q1 This authoring process is helpful for AR content creation
Q2 I will use this authoring process in AR content creation
Q3 the user-environment interaction will increase the immersiveness of AR
Q4 The sound authoring results is in high quality
Q5 I prefer the automatic sound acquisition over manual search/record
1
2
3
4
5
6
7
Strongly Disagree
Neutral
Strongly Agree

(a) SonifyAR's authoring interface. Left: SonifyAR's phone-based AR interface. Right: SonifyAR's authoring panel.

(b) Likert-type questions and results of usability study

Figure 4: SonifyAR interface and evaluation results

7 CONCLUSION

In this paper, we present SonifyAR, a context-aware sound authoring system designed for sonifying AR events. Our work implements a PbD authoring pipeline which enables sound asset acquisition using holistic context information. With SonifyAR, users could freely explore AR interactions and have AR sound effects automatically acquired. Our studies validate the usability and overall performance of our system. We believe SonifyAR can be applied in application fields like e-commerce and BLV assistance. We also envision it incorporated into existing AR authoring tools to strengthen their sound authoring capabilities. This research advances literature in AR sound authoring, while also opening up new research avenues for the application of LLM and generative models in AR systems.

REFERENCES

[1] [n. d.]. *ARVid - Augmented Reality.* https://apps.apple.com/us/app/arvid-augmented-reality/id1276546297 Accessed on September 24, 2023.

[2] [n. d.]. *Freesound.* https://freesound.org/ Accessed on September 24, 2023.

[3] [n. d.]. *Halo AR.* https://haloar.app/ Accessed on September 24, 2023.

[4] Adobe. 2023. *Adobe Audition Sound Effects Download Page.* https://www.adobe.com/products/audition/offers/AdobeAuditionDLCSFX.html Accessed on Date of Access.

[5] Adobe. Accessed September 11, 2023. Adobe Aero. https://www.adobe.com/products/aero.html.

[6] Apple. 2023. *Apple ARKit Documentation.* https://developer.apple.com/documentation/arkit/ Accessed on Oct 9th, 2023.

[7] Apple. 2023. *ARKit - Tracking and Visualizing Planes.* https://developer.apple.com/documentation/arkit/arkit_in_ios/content_anchors/tracking_and_visualizing_planes Accessed on Oct 9th, 2023.

[8] Apple. Accessed September 11, 2023. Reality Composer. https://apps.apple.com/us/app/reality-composer/id1462358802.

[9] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. 2023. VAST: A Vision-Audio-Subtitle-Text Omni-Modality Foundation Model and Dataset. arXiv:2305.18500 [cs.CV]

[10] Epic Games. 2023. *Unreal Engine.* https://www.unrealengine.com/en-US Accessed on Oct 9th, 2023.

[11] Dhruv Jain, Sasa Junuzovic, Eyal Ofek, Mike Sinclair, John R. Porter, Chris Yoon, Swetha Machanavajhala, and Meredith Ringel Morris. 2021. A Taxonomy of Sounds in Virtual Reality. In *Proceedings of the 2021 International Conference on Multimodal Interaction.* ACM, Montréal QC Canada, 80–91. https://doi.org/10.1145/3462244.3479946

[12] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Defossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2023. AUDIOGEN: TEXTUALLY GUIDED AUDIO GENERA-. (2023).

[13] Tessa A. Lau and Daniel S. Weld. 1998. Programming by Demonstration: An Inductive Learning Formulation. In *Proceedings of the 4th International Conference on Intelligent User Interfaces* (Los Angeles, California, USA) *(IUI '99).* Association for Computing Machinery, New York, NY, USA, 145–152. https://doi.org/10.1145/291080.291104

[14] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmose, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. In *The 34th Annual ACM Symposium on User Interface Software and Technology.* 626–637.

[15] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. 2023. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503* (2023).

[16] Kyzyl Monteiro, Ritik Vatsal, Neil Chulpongsatorn, Aman Parnami, and Ryo Suzuki. 2023. Teachable Reality: Prototyping Tangible Augmented Reality with Everyday Objects by Leveraging Interactive Machine Teaching. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* 1–15.

[17] A.-M. Oncescu, A.S. Koepke, J. Henriques, and Albanie-S. Akata, Z. 2021. Audio Retrieval with Natural Language Queries. In *INTERSPEECH.*

[18] Flávio Ribeiro, Dinei Florêncio, Philip A. Chou, and Zhengyou Zhang. 2012. Auditory augmented reality: Object sonification for the visually impaired. In *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP).* 319–324. https://doi.org/10.1109/MMSP.2012.6343462

[19] Dariusz Rumiński. 2015. An experimental study of spatial sound usefulness in searching and navigating through AR environments. *Virtual Reality* 19, 3-4 (2015), 223–233.

[20] Stefania Serafin, Michele Geronazzo, Cumhur Erkut, Niels C. Nilsson, and Rolf Nordahl. 2018. Sonic Interactions in Virtual Reality: State of the Art, Current Challenges, and Future Directions. *IEEE Computer Graphics and Applications* 38, 2 (March 2018), 31–43. https://doi.org/10.1109/MCG.2018.193142628 Conference Name: IEEE Computer Graphics and Applications.

[21] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580* (2023).

[22] Unity Technologies. 2023. *Unity.* https://unity.com/ Accessed on Oct 9th, 2023.

[23] Paul Upchurch and Ransen Niu. 2022. A Dense Material Segmentation Dataset for Indoor and Outdoor Scene Parsing. http://arxiv.org/abs/2207.10614 arXiv:2207.10614 [cs].

[24] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology.* 552–567.

[25] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671* (2023).

[26] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).

[27] Mark Yi-Cheon Yim, Shu-Chuan Chu, and Paul L Sauer. 2017. Is augmented reality technology an effective tool for e-commerce? An interactivity and vividness perspective. *Journal of interactive marketing* 39, 1 (2017), 89–103.

[28] Yongqi Zhang, Cuong Nguyen, Rubaiat Habib Kazi, and Lap-Fai Yu. 2023. PoseVEC: Authoring Adaptive Pose-aware Effects using Visual Programming and Demonstrations. In *ACM Symposium on User Interface Software and Technology*.

[29] Yuhang Zhao, Elizabeth Kupferstein, Brenda Veronica Castro, Steven Feiner, and Shiri Azenkot. 2019. Designing AR visualizations to facilitate stair navigation for people with low vision. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 387–402.

[30] Yuhang Zhao, Elizabeth Kupferstein, Hathaitorn Rojnirun, Leah Findlater, and Shiri Azenkot. 2020. The effectiveness of visual and audio wayfinding guidance on smartglasses for people with low vision. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.

[31] ZhiYing Zhou, Adrian David Cheok, Yan Qiu, and Xubo Yang. 2007. The Role of 3-D Sound in Human Reaction and Performance in Augmented Reality Environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 37, 2 (March 2007), 262–272. https://doi.org/10.1109/TSMCA.2006.886376 Conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.