Integration of Freesound content in creative applications and sound interfaces: a survey of use cases

Panagiota Anastasopoulou
panagiota.anastasopoulou@upf.edu
Music Technology Group, Universitat Pompeu Fabra
Barcelona, Spain

Frederic Font
frederic.font@upf.edu
Music Technology Group, Universitat Pompeu Fabra
Barcelona, Spain

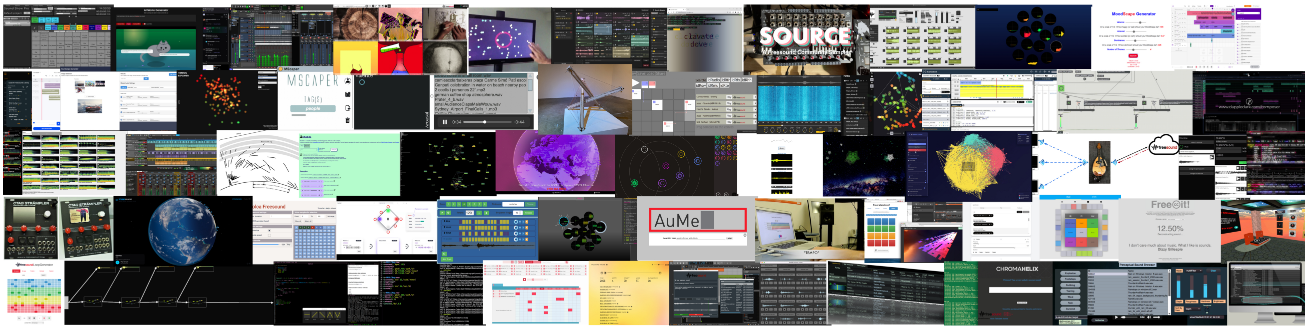


Figure 1: Collage of Freesound integration use cases

Abstract

Freesound is one of the most prominent collections of user-contributed audio samples, providing a large and heterogeneous collection of sounds for creative, educational, and research purposes. In addition to its web-browser interface, Freesound features an API that enables programmatic access to search features and sound resources, including descriptive metadata and content-based audio descriptors. As a result, third-party developers and researchers can access and integrate Freesound content in advanced ways, with applications in various fields including (but not limited to) music making, sound design, machine learning, and multimodal systems. This paper surveys creative applications and sound interfaces that integrate content from Freesound, addressing a wide range of use cases that utilize it interactively. We outline their technical capabilities and use of Freesound resources, and explore emerging research directions. We focus on how these systems interact with Freesound, how its content is integrated within their overall design, and how they engage users. The study shows that Freesound has become an infrastructural and inspirational resource for creative applications and sound interfaces. Nevertheless, the study reveals that many of Freesound's API capabilities remain underused, pointing to emerging opportunities for further development of applications and modes of interaction.

Keywords

audio retrieval, sound collection, interface, creative, use-case, Freesound

1 Introduction

Large, open, and heterogeneous sound corpora actively support music creation, sound research, audio technology development, among other applications. Freesound, an online collaborative collection of sounds released under Creative Commons licenses, is a widely used and continuously growing example of such a resource [28]. It currently hosts over 720,000 sounds and has more than 16 million registered users. Although the web interface remains a primary mode of interaction, additional access methods provided by the Freesound API enable the integration of Freesound content into third-party applications. Moreover, unlike traditional curated datasets, Freesound provides access to a heterogeneous collection of sounds with varying quality, content, and metadata, contributed by a global community. Thus, it supports research and development across applications such as computational creativity, web-based applications, hardware platforms, human-computer interaction, machine learning and audio modeling systems, sound design and music production tools, and intelligent sound retrieval, with projects spanning creative practice, research, education, and interactive installations.

Over its twenty years of operation, Freesound has been integrated into a wide range of projects, serving both as a resource and as a platform for sound exploration. Existing integrations vary in whether they foreground exploratory interaction or direct access, rely on semantic or content-based retrieval, support minimal or more open-ended interaction, and in the means of interaction with the database. This work proposes a set of shared analytical axes to examine a selection of these integrations, focusing on how different design choices shape user interaction with Freesound's corpus (Sec. 3). The scope of this work is limited to *creative applications and sound interfaces*, defined as audio-centric systems that support creative engagement with sound or its manipulation as a primary material. We therefore exclude systems that do not support these forms of creative interaction, particularly a range of Freesound use cases such as plain search tools (e.g. MuSST[1], CCMusicSearch[2]), video-centered applications and games (e.g. Blender Add-on[3], Freesound Audio Search

NIME '26, June 23–26, 2026, London, UK

[1]https://labs.freesound.org/apps/2019/01/13/jamendo-musst.html
[2]http://ccmusicsearch.com
[3]https://github.com/iwkse/freesound

Adobe plugin[4], Reward Circuit[5]), non-interactive works or systems limited to fixed curated palettes (e.g. 28x28[6]), libraries (e.g. simple-freesound[7], py4pd-freesound[8]), and archival/dataset resources (e.g. Sons da Praia[9], Freesound-based datasets[10]).

Through a systematic comparative analysis, the study aims to identify recurring design patterns, underlying assumptions, and inherent trade-offs between accessibility, expressivity, and exploratory potential (Sec. 4). By situating these projects within a common framework, the analysis highlights how Freesound has functioned not only as a repository of audio files but also as an evolving infrastructure for sonic research, creativity, and experimentation. The findings emphasize Freesound's dual role as an infrastructure and an inspiring resource for creative sound-based systems. However, the analysis also shows that significant portions of the API remain underutilized, revealing potential for expanded application design and emerging modes of interaction (Sec. 5).

2 Freesound content and access

2.1 Freesound web interface

The Freesound web interface centers on text-based search, which operates on user-provided metadata including tags, titles, and textual sound descriptions. The search input box is prominently displayed on the landing page, while in the search page it is accompanied by an optional set of advanced search filters (e.g. duration). After an initial query is carried out (which can be an *empty* query), users can apply further facet filters that include licence type, category, tags, and user, as well as audio file properties (e.g. sample rate, number of channels, etc.). On individual sound pages, a similarity button enables the discovery of related sounds. Additional browsing features include a random sound option and an interactive map for navigating geolocated audio content.

2.2 Freesound API

The Freesound API is a REST-based interface that allows programmatic access to the Freesound database through a variety of client libraries (e.g. Python, JavaScript, JUCE, SuperCollider, Max/MSP, Common Lisp). It enables access to sound resources and search functionalities, including audio files, user-contributed metadata (e.g. tags, geolocation, license), a broad collection of audio-content descriptors[11] (e.g. pitch, brightness), and other data such as audio file properties, number of downloads and upload date[12].

The API was first introduced in 2011 and was updated to APIv2 in 2014, adding improvements and new features such as sound uploads and the ability to combine sound metadata with audio descriptors during search. A major update in 2025 further enhanced retrieval and search capabilities, among others, by restructuring metadata retrieval, introducing new sound similarity spaces and audio descriptors, and supporting Freesound's newly introduced categorization scheme based on the Broad Sound Taxonomy [9].

[4]https://exchange.adobe.com/apps/cc/108490/freesound-audio-search
[5]https://labs.freesound.org/apps/2014/12/05/rewardcircuit.html
[6]http://28x28.xavierbonfill.com/
[7]https://ircam-rnd.github.io/simple-freesound/
[8]https://github.com/charlesneimog/py4pd-freesound
[9]https://bancosonoroamazonico.vercel.app/sonsdapraia
[10]https://labs.freesound.org/datasets/
[11]https://freesound.org/docs/api/analysis_docs.html
[12]The full list is available at https://freesound.org/docs/api/resources_apiv2.html#sound-instance.

Currently, the API offers a unified *search* endpoint that supports multiple retrieval strategies. More specifically, these include text-based queries over sound metadata; content-based retrieval using low-level audio features (e.g. temporal descriptors such as *duration* and *onset rate*, spectral descriptors such as *spectral centroid, rolloff, flux*, and *MFCCs*, perceptual attributes like *loudness*, *brightness*, and *roughness*); higher-level semantic descriptors (e.g. *sound categories*, *sources*, and *detected events*); and combinations thereof. Before the changes introduced in 2025, these features were split between the *text search*, *content search*, and *combined search* endpoints. The API also supports a number of sorting options (e.g. relevance, number of downloads) and similarity-based retrieval methods over acoustic and semantic similarity spaces. In addition, dedicated endpoints for *users*, *sounds*, *packs*, and *collections* enable structured access to grouped sound resources.

2.3 Web-based vs API access

By analyzing Freesound's web access and API logs, we estimate the ways in which Freesound data are accessed and integrated. The web interface features 12 million weekly search queries, while the Freesound API handles approximately 1 million search queries per week[13]. Even though not all such requests originate from the type of creative applications and sound interfaces which are object of our study –for instance, a common use case which is outside this scope is the use of the API for creating training datasets and batch-downloading sounds– these numbers suggest that the API is a major means of accessing, viewing, and saving content. As a result, it plays a central role in how many users experience and interact with Freesound content.

The vast majority of API search requests target the *text* search endpoint, while the former *content* and *combined* search endpoints are used much less frequently, with the latter being extremely rare (N≈750/month and N<50/month, respectively). As mentioned above, since the latest API update, these three endpoints are combined into a single *search* resource. We observe, nevertheless, that requests to the new *search* endpoint that take advantage of the integration of content-based descriptors still account only for a very small fraction of queries (peak N≈2,000/month for a single descriptor). It is important to note that these statistics cover all types of API usage, and may not directly reflect the usage originating from applications included in our study. The analysis on the following sections focuses on individual uses of Freesound content from third-party applications, with a focus on creative and sound-based interfaces.

3 Analytical framework

3.1 Overview

Freesound is accessed through multiple interfaces that reflect different patterns of use. Although we have observed that the web interface remains the primary entry point for browsing, listening, and manually downloading of sounds, a significant portion of Freesound usage occurs through the Freesound API, which offers alternative ways to interact with the sound database and expand creativity. Freesound content is integrated in many creative applications and sound interfaces, but the way it is integrated varies considerably. In some cases, Freesound is a central element of the interaction, while in others it functions as a secondary resource that complements a broader workflow. These differences

[13]The data are reported as weekly averages aggregated over an approximately three-month period (November–January).

in functionality shape both how users search for sounds and how they engage with the collection.

In order to get insights on Freesound integration usage patterns, we systematically analyze all relevant creative applications and sound interfaces identified through our data collection proces and characterize them along analytical axes defined for this study. In the following sections, we first describe how we selected the applications that are object of our study (Sec. 3.2), and then define the different analytical axes used to characterize them (Secs. 3.3–3.7).

3.2 Selection of creative applications and sound interfaces

To begin the collection of relevant projects, we examine the entries listed in *Freesound Labs*[14]. Freesound Labs is a listing of projects, hacks, applications, workshops, and research initiatives that either make use of Freesound content or are built on top of the Freesound API. At the time of our study, the Freesound Labs main page contained 56 entries, either gathered by the Freesound team or submitted by community members. For the purposes of this study, we only considered 42 of these entries, excluding entries such as workshops, installations, and other applications that did not align with the scope of this work (Sec. 1).

To complement this source, we analyzed Freesound API usage logs to identify applications that use the platform programmatically. We examined *API clients* that accumulated over 10,000 total queries during a five-year period (2021-2025). In many cases, it was not possible to obtain relevant information about the API clients originating such requests (i.e. no links to project pages or source code), preventing their potential inclusion in the study. In addition, we searched public GitHub[15] repositories for references to Freesound or its API in the project documentation, focusing on occurrences in README files. Altogether, these methods allowed us to further add 18 projects that integrate Freesound and meet our criteria for inclusion in the study.

The resulting set includes 60 creative applications and sound interfaces that interact with Freesound and actively engage with sound for purposes like search, retrieval, processing, or exploration.

3.3 Centrality of sound corpus

The first axis of evaluation addresses the centrality of the sound corpus within each application, defined by the role the sound collection plays in the user's workflow. In some systems, exploration and manipulation of the collection content drive the primary musical or sonic activity, while in others it functions as a supporting resource. Drawing on human-computer interaction and interactive information retrieval frameworks [34, 80], we conceptualize the corpus as a mediating resource whose structural role shapes creative activity. Accordingly, we assess Freesound integration in terms of its functional centrality: the degree to which it scaffolds, constrains, or enables the user's musical or sonic exploration. Based on this perspective, we identify the following categories:

- **Peripheral (0)**: Freesound is a supporting element; the primary task could function without it.
- **Foundational (0.5)**: Freesound provides the main sound database but is not the conceptual focus of the interface.
- **Central (1)**: The interface is built around Freesound; exploration and manipulation of its content constitute the main purpose of the application.

This distinction separates the structural importance of the corpus from the degree of interactivity and freedom it affords, allowing us to evaluate how deeply Freesound shapes the overall user experience.

3.4 Interactivity

Interactivity is the dynamic quality of input-feedback behaviour between the user and the system [12]. In the context of a sound collection, it describes the extent to which users can engage with its content and receive immediate, responsive results. Highly interactive systems enable users to iteratively test ideas, adjust parameters, and observe real-time outcomes, fostering exploratory behaviour that supports creativity and discovery. Interactivity is not limited to visible user-interface actions; behind-the-scenes (backend) processes can also improve responsiveness, making interaction feel fluid and dynamic. Even when hidden from the user, these system capabilities contribute to exploration by providing feedback and refinement in the background. We operationalize interactivity as the degree to which the system enables interaction with the database [10, 19]. This includes frontend interaction access, time flexibility, and responsiveness, as well as backend processes that support dynamic interaction. Specifically, interactivity refers to the range and effectiveness of options available for searching, filtering, navigating, and describing sound content, focusing on user control and exploration rather than general interface responsiveness or creative manipulation features.

We define interactivity levels as follows:

- **None (0)**: No user control over the retrieval or manipulation of sounds from the database.
- **Limited (0.5)**: Restricted control, with only basic filtering options or limited user-controlled search functionality available.
- **Basic (1)**: The system offers at least one user-controlled type of search (e.g. keyword search) and up to four metadata filters.
- **Advanced (2)**: The system supports two or more search types.

3.5 Retrieval and filtering

We analyze how different types of sound descriptors are used for retrieval in interactive applications. The available sound retrieval strategies are fundamental in sound-based interface design, as it determines how users access and manipulate audio content [23, 38]. We focus on three main retrieval strategies, text-based metadata, content descriptors, and sound similarity:

- **Acoustic / content (a):** Relies on extracted audio features such as timbre, pitch, rhythm, and spectral properties, allowing users to find sounds based on sonic characteristics regardless of semantic labeling.
- **Text-based / semantic (t):** Relies on textual metadata, including tags, descriptions, and user-provided annotations. Semantic search supports goal-directed exploration, enabling users to find sounds according to conceptual categories or intended meaning.
- **Similarity-based (s):** Relies on representations of sounds, such as feature embeddings derived from acoustic or semantic information. Sounds are retrieved based on their

[14]https://labs.freesound.org/
[15]https://github.com/

proximity in this representation space to a similarity target, and it supports exploratory workflows by grouping perceptually or conceptually related sounds together.

Retrieval processes can be combined, resulting in a multi-choice system where users can select or combine strategies [76]. Additionally, we may consider **random (r)** as a retrieval strategy, where sounds are retrieved without specific control over the selection process. While not user-driven, this strategy could potentially foster exploration by introducing variability, especially in large sound collections [72].

There are also several available filters that refine sound retrieval. These filters can be applied to control or constrain the result set, offering users more control over the exploration process. Available filters include: **license (l)**, **duration (d)**, **number of samples (n)**, **username (u)**, **pack name (p)**, **tag (t)**, **date added (da)**, **number of channels (ch)**, **file type (ft)**, **file size (fs)**, **explicit sound discard (e)**, and **geolocation (g)**[16]. In addition, the **sorting (s)** strategy (e.g. by relevance score, date added, or downloads in ascending or descending order) is relevant for applications that display sound content in a specific order. In certain instances, specific filters may assume the role of retrieval strategies, as their substantial influence on the organization and navigation of content enables them to function beyond traditional filtering, effectively shaping the retrieval process itself in the same way as acoustic/semantic-based constraints.

Lastly, there is the option to retrieve metadata associated with sounds (e.g. tags, descriptors, or other attributes) for later use or display, without altering search results during retrieval. This functionality, provided through Freesound, enables the collection of sound contextual information without narrowing the dataset, thereby enabling further tasks (e.g. display, meta-analysis) subsequently.

3.6 Input mechanism

Input mechanisms are conceptualized as interaction modalities that mediate the user's engagement with the sound database. Drawing on multimodal interaction theory and embodied interaction [18, 54], we categorize systems according to the modalities through which users influence retrieval and database-related manipulation processes. In the context of digital musical instruments and creative systems, input modalities are known to shape expressive affordances and compositional constraints [37, 51]. To systematically categorize the input mechanisms used across the systems, we define the following types based on interaction modality and musical functionality:

- **Typing/Clicking (t)**: Input via computer keyboard typing, mouse clicking, mobile tapping, or menu selections.
- **Gesture (g)**: Bodily or spatial input, such as physical movements or gestures.
- **Voice command (v)**: Vocal input used specifically for command or control purposes.
- **Audio (a)**: Sound signal as input, typically used for content analysis or similarity, either via file or microphone.
- **Hardware (h)**: Musical controllers or keyboards that manipulate events or parameters.
- **Other mode (o)**: Alternative modalities (e.g. image or video input).

We restrict this dimension to modalities that directly affect interaction with the Freesound corpus (e.g. retrieval, filtering, navigation). While many systems also include input mechanisms that control synthesis, or broader interface control, we follow stratified models of interaction in information retrieval [40, 63] to distinguish resource interaction from general interface control (e.g. audio or voice input used for purposes other than querying and interacting with the system).

3.7 Type of manipulation

Once sounds have been retrieved from the system, they can undergo different forms of manipulation, depending on the system's capabilities and the user's intentions. These manipulations are crucial for understanding how users interact with and modify the sound content for their creative or practical purposes [75]. We define the following categories of sound manipulation, based on how sounds are organised, structured, activated, and transformed within a system.

- **Access (acc)**: The sound is retrieved and saved, available for use or further exploration, without allocation, organization, or modification.
- **Access+ (acc+)**: The sound is retrieved and saved, and immediately placed within an interactive workspace or system environment for further exploration or use.
- **Arrangement (arr)**: Sounds are organised within a structured system for controlled playback, where playback behaviour can be interactively shaped by the user (e.g. looping, layering, panning, or modulation), without altering the audio content.
- **Arrangement (arr+)**: Extends arrangement with additional control over individual audio content, allowing interactive or direct alteration of its sonic properties (e.g pitch shifting, cropping, filtering, effects).
- **Transformation (tr)**: Sounds are organised or reconfigured by the system, where the structure, relations, or material organisation of sound is automatically generated or significantly altered through computational processes.

The first type of material manipulation is common in scenarios where the sound database is used to expand an existing sound library. The other types instead support more active interaction, where users are provided with specific tools to organise or modify sounds. While only access establish a more open relationship with the sound material, the latter four introduce more structured forms of exploration, enabling engagement through predefined or system-guided organisation and transformation of sounds.

4 Results

[h]

Table 1 shows the manual categorization and comparison of the applications according to the analytical axes described in the previous sections. The columns are labeled as follows: *inp.* for input method; *inter.* for interactivity; *cent.* for centrality, *type* for type of manipulation; and *retr.*, *f.u.*, and *f.q.* for the retrieval and filtering axis, referring respectively to search methods, user-accessible filters, and predetermined query filters[17]. The data is sorted chronologically.

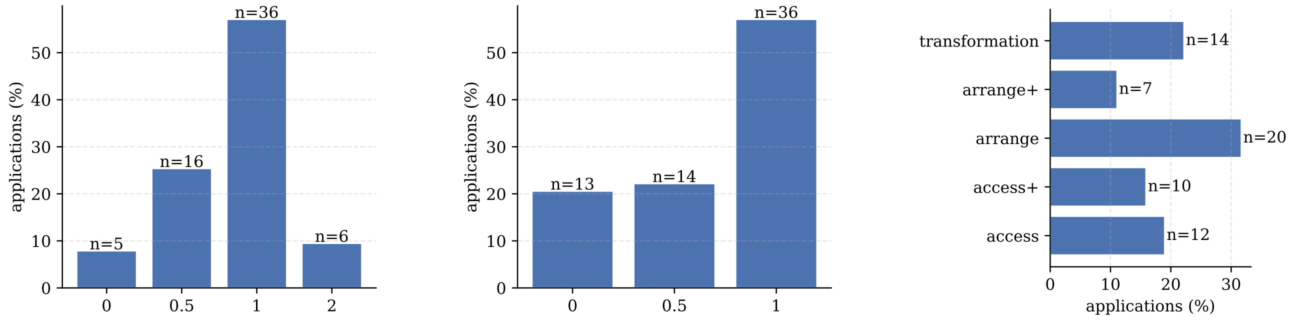
As shown in Fig. 2b, more than half of the applications analyzed position Freesound as a central resource. Specifically, 57.1%

[16]Section 2 provides an overview of Freesound filters; here we include only those discussed in the remainder of the paper.

[17]The columns use the abbreviations introduced in Sec. 3. Items marked with an asterisk denote uncertain classification due to insufficient data. Blank entries indicate unknown information; square brackets [] indicate optional modes; parentheses () in the retrieval column denote predetermined retrieval.

Table 1: Creative applications and sound interfaces using Freesound content

name	date	keywords	inp.	inter.	retr.	cent.	f.u.	f.q.	type
Freesound Radio [60]	2008	standalone, graph	t	1	t	1	-	-	arr
Freesoundscapes [5]	2011	web, 2d	t	1	t	1	-		arr
Free(cc)it! [30]	2012	web, generation	a	0.5	(a,t)	0.5	-	d	tr
Audio Metaphor [71]	2012	web, generation	t	0.5	t*	0.5	-		tr
Overtone [3]	2012	live-coding, language	t	1	t	0	open	-	acc+
Minority Freeport [29]	2013	standalone, 2d, sound-pool	g	0.5	(t)	1	-	d	arr
plexure.js [46]	2013	web, 2d	t	1	t	1	-	d	arr
Music Contextualizr [49]	2013	web (local), lyrics	a	0.5	(t)	0.5	-	d	tr
idRadio [79]	2014	web, sound-pool	-	0	-	0.5	-	-	arr
Elevator Music Generator [11]	2014	maxmsp, generation	-	0	(t)	0.5	-	ft,ch,t	tr
Free Maschine! [6]	2014	web, drum-machine	t	2	t,s,r	1	-		arr
Fsmic [62]	2015	web, semi-interactive, qbe	a	1	s	1	-		acc
Floop [61]	2015	interactive, loop, sound-pool	t	0.5	-	1	-	-	arr
Cloud Browser [65]	2015	plugin, daw	t	1	t	0	l		acc
Beatpush [22]	2016	web, drum-machine	t	1	t	0	-		acc+
Hands-Free Sound Machine [41]	2016	web, drum-machine	v,g	1	t	1	-	d	arr
Web Audio Sequencer [45]	2016	web, sequencer, drum	t	1	t	0.5	-		arr
Ardour [16]	2016	daw	t	1	t	0	-	-	acc
MIRLC [78]	2016	live-coding, extension	t	2	t,a,s,r	1	-	-	arr+
Infinite Orcherstra [64]	2016	music-making, collaborative	t	1	t	1	-	d	arr*
Soundly [43]	2016	daw	t	1	t	0	l	-	acc
Soundscape Generator [50]	2017	web-app, image	t,o*	0.5	t*	0.5	-		tr
Perceptual Sound Browser [15]	2017	standalone, browse, sound-pool	t	0.5	-	1	-	-	acc
Freesound Explorer [26]	2017	web, 2d	t	1	t	1	s,n,d	t	arr
Freesound Timeline [27]	2017	web, chronologic	t	0.5	da	1	-	-	tr
Freesound Trip [14]	2017	web, 3d	t	1	t	1	-	d	arr
Loskop Radio [44]	2017	web, generative	-	0	(t,s)	1	-	d	tr
Mixcraft 8 [4]	2017	daw	t	1	t	0	l	-	acc
AudioTexture Free [66]	2018	daw, plugin, loop	t	1	t	0.5	-	-	tr
Freemix [48]	2018	web, movement	g,t	1	t	1	u,d		arr+
MoodScape Generator [55]	2018	web, soundscape	t	0.5	(t)	1	-		tr
Multi Web Audio Sequencer [21]	2018	web, sequecer	t	1	t	1	d	-	arr
Smart Mandolin Freesound System [73]	2018	hybrid-instrument, mobile	t,a	0.5	(t)	0.5	-		arr
Volca-Freesound [36]	2018	hardware, app	t,h	0.5	r	1	d	-	acc+
Soundtrap [67]	2018	daw	t	1	t	0	-	l	acc
Playsound [17]	2018	web, grid, music-making	t	2	t,s	1	-		arr+
EarSketch [53]	2018*	music coding, education	t	1	t	0	-	-	acc
CTAG Strämpler [52]	2019	hardware, modular, sampler	h	1	t	1	-	-	arr+
Timbral Explorer [57]	2019	web, 2d	t	1	t,a*	1	-		acc+
FreesoundSimpleSampler [59]	2019	plugin, sampler, juce	t,[h]	1	t	1	-	-	acc+
Mscaper [70]	2019	plugin, soundscape, ableton	t	1	t	1	-		acc
Freesound Loop Generator [24]	2020	web, drum-machine, loop	t,[h]	2	t,s,r	1	l	d	arr
GifDub [42]	2020*	web, audiovisual	t	0.5	(t*)	0.5	-		tr
Freesound Player [31]	2020	digital-instrument, maxmsp, midi	t,[h]	2	t,a	1	d,n		arr+
SOURCE [25]	2021	hardware, sampler	h	2	t,a,s,r	1	d,l	-	arr+
Lockdown Composer 7 [68]	2022	daw	t	1	t	0	-	l,d	acc
Shabda [32]	2022	live-coding, extension, web	t	1*	t,(s)	1	l	d	acc+
hydra-freesound-auto [35]	2022	live-coding, extension, visual	-	0	(t)	1	-		tr
MelodyLine [8]	2022	live-coding, extension	t	1	t,(a)	1	-	-	acc+
Imaginary Machinescapes [7]	2022	installation, generation	a	1*	t*	0.5			tr
FreesoundVR [74]	2022	vr, daw, composition, visual	g	0.5	-	1	-	-	arr+
Sound Show [39]	2023	soundboard	t	1	t	0	-	-	arr
Hathor [56]	2024	web, drum, sound-pool	-	0	-	0.5	-	-	tr
FollowWeb [69]	2024	standalone, visual, semantic	t	1	t	1	u,p,t	-	acc+
MPC program creator	2024	standalone, editing, hardware export	t	1	t	1	d	d	acc+
Earth song [13]	2024	web, 3d, geospatial	t	1	g	1	-	g	arr
Audio Forge [2]	2025	web, soundboard	t	1	t	0	-	s	arr
Audio Editor App [77]	2025	android, editing	t	1	t	0	s	s*,da	acc
Soundpot [20]	2025	web, 2d	t	0.5	r	0.5	-	d,l,e	arr
MLA Labs' The playground [47]	2025	web, processing	t	1	t	0	-	fs	acc+
AI Movie Generator [1]	2025	web, audiovisual, generative	t	0.5	(t)	0.5	-	d	tr
Freesound Rack [33]	2025	plugin, standalone, sampler	t,[h]	1	t	1	-	d*	arr
EthnoSphere [58]	2026	web, 3d, geospatial	t	0.5	g	1	-	d	acc



(a) Interactivity **(b) Centrality** **(c) Type of manipulation**

Figure 2: Distribution of interactivity, centrality, and type of manipulation across applications.

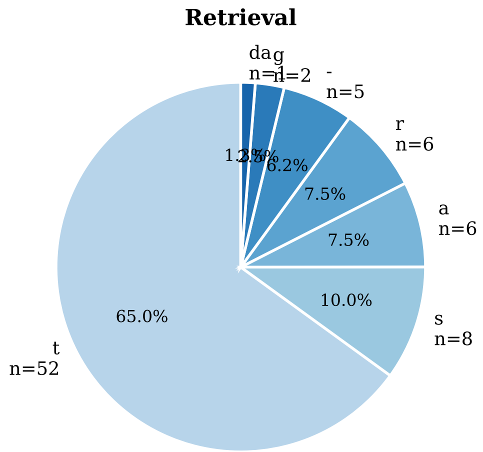


Figure 3: Retrieval usage across applications

Note: values are aggregated; inside = occurrence frequency, labels = system coverage.

Table 2: Filters in user interfaces and backend querying

filter	f.u. (n)	f.u. (%)	f.q. (n)	f.q. (%)
(none)	47	74.6	22	34.9
duration (d)	7	11.1	16	25.4
license (l)	6	9.5	3	4.8
number samples (n)	2	3.2	—	—
sort (s)	2	3.2	2	3.2
username (u)	2	3.2	—	—
(not identified)	1	1.6	18	28.6
pack name (p)	1	1.6	—	—
tag (t)	1	1.6	2	3.2
open	1	1.6	—	—
file type (ft)	—	—	1	1.6
date added (da)	—	—	1	1.6
number of channels (ch)	—	—	1	1.6
file size (fs)	—	—	1	1.6
no explicit sound (e)	—	—	1	1.6
geolocation (g)	—	—	1	1.6
Total	70	100	69	100

Note: Values indicate both absolute counts and percentages of systems in which each filter type appears.

Table 3: Distribution of input types

input	app. (n)	app. (%)	occ. (%)
t	49	77.8	68.1
h	7	11.1	9.7
-	5	7.9	6.9
a	5	7.9	6.9
g	4	6.3	5.6
o	1	1.6	1.4
v	1	1.6	1.4

are classified as *central*, meaning that exploration or manipulation of Freesound content constitutes the main purpose of the interface. A further 22.2% are categorized as *foundational*, where Freesound serves as the primary sound database but is not conceptually foregrounded. Applications such as Soundscape Generator, Free(cc)it!, Imaginary Machinescapes, AudioTexture Free, Audio Metaphor, Soundscape Generator, Elevator Music Generator and AI Movie Generator rely on Freesound to generate a final composition, soundscape, or design outcome. However, they do not typically support free-form exploration or individual triggering and manipulation of retrieved sounds (i.e. the most common type of manipulation is *transformation*). In these systems, Freesound functions as a structured material source embedded within a higher-level generative or compositional process. Some borderline cases illustrate this distinction further, such as Hathor, which incorporates a substantial number of Freesound samples, yet the corpus does not constitute the exclusive or conceptual focus of the system, as its emphasis lies on drum pattern generation rather than sound corpus exploration.

By contrast, 20.6% integrate Freesound *peripherally*, as a complementary resource that supports other core functionalities. This interaction mode is particularly common in Digital Audio Workstations (DAWs) and editing tools, such as Ardour, Soundtrap, Cloud Browser, Sound Editor App, and Lockdown Composer 7, as well as in sound organization or post-production tools like Soundly, SoundShow, and Audio Forge. In such cases, Freesound usually serves as a convenient repository of Creative Commons sounds suitable for academic or commercial projects, having heightened correlation with the sound *access* system types. This is also reflected in more large and complete projects, such as Overtone and EarSketch. Additionally, the interface in such integrations is typically minimal, often limited to a basic text-based search (see later). This pattern suggests that when Freesound is integrated as a supplementary resource, the design emphasis lies on efficiency and seamless workflow integration rather than exploratory interaction, also reflected across other analytical axes.

In terms of interactivity with the Freesound corpus, most systems provide at least a basic level of user control. The *basic (1)* category receives over half of all applications (see Fig. 2a), the highest among all categories. These systems typically offer a single search modality, most often keyword-based semantic search accompanied by a limited number of filtering options. At the lower end, 7.9% of the systems exhibit *no (0)* interactivity, meaning users have no control over the search process. In such cases, retrieval can occur automatically in the background, and Freesound functions as a material source for generative processes rather than as an interactively explorable corpus. Examples include hydra-freesound-auto, a self live-coding system that fetches sounds algorithmically; Hathor, an automatic rhythm generator, Elevator Music Generator for infinite music generation, and radio-based projects such as Loskop Radio. A *limited* but higher degree of interactivity (25.4%) appears often in generative systems, where access to the database may be mediated through backend translation rather than direct querying. For example, MoodScape Generator allows users to input perceptual values, which are then internally mapped onto Freesound queries. While this additional layer may create a distinct experiential interface, the internal constraints and the lack of sound information display limit transparent or iterative exploration of the corpus itself. Other systems with limited interactivity permit simple, one-off queries but with delayed or non-iterative responses. Music Contextualizr, AudioMetaphor, and Free(cc)it!, for instance, retrieve acoustically or semantically related sounds given an input audio file, yet response times and the absence of continuous refinement reduce the immediacy required for sustained interactive exploration. Some systems in this category rely on predefined sound categories (e.g. Perceptual Sound Browser, Smart Mandoline Freesound System, FreesoundVR), often in prototyping contexts to ensure heterogeneity without full interaction yet, while it can also be an aesthetic constraint guiding discovery (e.g. GifDub).

Conversely, *highly (2)* interactive systems (9.5%) include the Freesound Player, which supports text search combined with acoustic search; the Freesound Loop Generator and Fressmaschine!, where users can choose among three strategies (text, similarity, randomness); and MIRLC and SOURCE, which implement all four main retrieval strategies. In contrast to MIRLC, SOURCE can actively combine retrieval strategies in parallel. An interesting case is Overtone: although it appears to offer only basic interactivity, it enables open-ended filter definition by allowing users to manually specify filter names and values. Historically, Overtone did not support content search, but recent API updates have made this possible through this field. This is an example of how API evolution can significantly affect ease of implementation and expand interaction opportunities.

We notice that increasing interactivity requires more retrieval or filtering options. As evidenced in Fig. 3, most applications rely on text-based searches, using keywords to capture semantic aspects (e.g. sound source, action, location). This is reflected in the fact that more than 50% of applications use only text search, while it appears in 82.5% of all applications. Random search (N = 6) appears as the only modality in 2 cases, Volca-Freesound and Soundpot. Both limit retrieved sounds to short duration for sampling, albeit with limited control over precision. Acoustic search appears only in 6 applications (4 user-controlled, never as the main focus). Such behavior is observed in MelodyLine and Shabda, which rely on interactive text search while their auxiliary search remains predetermined. Similarity search (N = 7) is predominantly used to replace retrieved sounds, appearing in combination with another modality in 6 cases; the exception, Fsmic, employs it in isolation to match sounds to microphone input but without supporting further creative use.

That being said, some applications introduce an additional interaction step through post-retrieval analysis, occurring independently of the original query before results are displayed. Locally computed in most cases, this information may alternatively be derived from Freesound's acoustic metadata; Floop, for example, opts for the former despite the latter being available. Minority Freeport, Freesound Explorer and Timbral Explorer retrieve content-based descriptors but to compute similarity relations locally, with the first two also using this information for display purposes (e.g. mapping sound point colors to tristimulus values). Interestingly, none of all applications use Freesound's built-in similarity for visual display, despite many use visualization techniques (as seen later). Several applications perform their own analyses because of lack of such information in Freesound[18]: Mscaper, Perceptual Sound Browser, and Moodscape Generator all use custom analyzers as additional after-retrieval filters, while Audio Metaphor performs tasks such as foreground/background classification that does not exist in Freesound.

Additionally, a minimal amount of applications (N = 3) retrieve acoustic information not for filtering or post-analysis but for displaying, or potential later use. Specifically, Lockdown and EarSketch use *BPM*, while Mixcraft 8 also uses *key* and *scale*. By contrast, retrieving text metadata for display is more common, typically focusing on title, tags, sound ID, description, and license. Some applications, like SoundShow, display this information on hover, while Freesoundscapes integrates it more seamlessly as a decorative element within the interface.

Among the available filters, 74.6% of the systems do not offer any user-controlled options (see Table **??**). When offered, *duration* and *license* are the most common (N = 6–7, each), while the remaining 6 filtering or sorting options observed appear seldom (N = 1–2), making uncommon for users to exert further influence. Filters are often presented as drop-down menus or lists (e.g. SOURCE low/mid/high, Locker Composer 7 license list), while sliders may appear for duration and content-based filtering/search (e.g. Freesound Player, Freesound Explorer). This design approach likely reflects the goal of these integrations: to provide rapid, low cognitive-load access to Freesound content without requiring users to leave the host system. This is also supported by the low proportion of 1+ filters within systems, which accounts for only 9.5% (N = 6). Alongside, at least 33.3% of systems use no predefined filters[19], while a surprising 25.4% use duration. This may be because users tend to select short sounds to fit within looping and sequencing scenarios. Additionally, licenses are used 3 times to exclude more restrictive ones, while 8 other filters are used only once or twice each.

[18]at least at the time of application creation
[19]information for 28.6 was not accessible

Applications that allow *access* without an additional allocation (19%) correlate with those that Freesound is not central. Another 15.9% of applications allow enhanced *access+* by allocating tha audio inside the environment and remaining available for smooth integration in the workflow. Correlating the prevous one with DAWs (earlier), it is now this, common i live coding systems (shabda, melodyine, overtone as a real-time compositional technique but without restricting the usual pipeline of the peorformer. offline vs real time Sometimes though it does give new tools Mirlc In the case of *arrangement*, a very common strategy is to place sounds into a structured grid where the user can define relationships, as seen in Free Maschine!, Hands-Free Sound Machine, Beatpush, Freesound Rack, Freesound Player, idRadio, Web Audio Sequencer, GifDub, and Playsound, Freesoundscapes, Earth song. The position in space may be fixed or the user can position sounds working as a triggering ssytem or sometimes spatial or volumn organization (utility or relational afect on sound). On the other hand, Timbral Explorer and FollowWeb, meaningful (perceptual and semantic) allocation without allowing compositional

Similar systems with additional capabilities to *arrange+* are 11.1%, showing a tendency to preserve the sounds are they are. That could be because of some 'purer' sonic discovery

Other prominent choices include 2D spaces (Minority Freeport, Floop, Freesound Explorer, Freesoundscapes, Soundpot), 3D spaces (Freesound Trip fly manually through space; map-like EthnoSphere, Earth song; network graph FollowWeb), inserting more structure flexibility than the grid. All arrangements in the space happen from local analysis, except Earth song that uses the Freesound geotag data to organize sounds. Lastly, Freesound Radio follows sequential linear playback arrangement.

Freesoundscapes, Soundpot, and Earth song adhere also to *transoframtion* (immediate alteration or manipulation) as it organizes and triggers in space but the space transform, panning, loudness, creating a soundscape. Applications provide optional editing techniques for each sound, like cropping (e.g Audio Editor) and pitch-shifting (e.g. The playground). In contrast, AudioTexture Free creates an infinite loop, which is more extreme and inherently structural transformation, as it alters the sounds' internal structure. As mentioned, applications that aim to produce a final product (e.g. AudioMetaphor, Free(cc)it!, Music Contextualizr) focus on transformation, either generatively creating infinite output or synthesizing from existing material. Interestingly, only about one quarter of applications alter the initial material at all; the rest involve no dedicated tools for modification.

The majority of applications are controlled via typing and clicking, maybe due to their web-based nature, and account for 78% of all inputs. Approximately 8% of applications do not provide user-controlled input. Alternative modalities include gestures (N = 3), voice control (N = 1), audio content input (N = 4) –3 times input file, while one to record–, dedicated hardware (N = 3), and optional hardware supporting MIDI mapping via a computer keyboard or MIDI controller (N = 4). Image input appears once, in combination with text typing. Overall, other accessibility options are limited, indicating that multimodal or assistive inputs are uncommon.

5 Discussion and conclusion

Exploration in Freesound-based creative applications and sound interfaces can emerge from the relationship between two independent dimensions: the *centrality* of Freesound within an application's workflow and the *interactivity* through which access and manipulation of its content is realized; how important Freesound is to the application's purpose and how much control the user has over accessing and shaping its content. Applications where Freesound is central and interactivity is high offer the greatest potential for user-driven exploration. Our analysis shows that although Freesound is central in many applications, high interactivity is comparatively less common. In particular, text-based search dominates as the main access paradigm, often with limited use of advanced filters, acoustic descriptors, relational metadata, or similarity-based retrieval. Centrality, therefore, does not necessarily imply rich user-driven exploration. Applications may rely heavily on Freesound as a content source while still constraining how users meaningfully navigate or interrogate the collection. This tendency aligns with broader API usage patterns, where a relatively small subset of available functionalities is most frequently used. While the ecosystem of applications exhibits considerable diversity in interface design and creative intention, many systems converge around a narrow interaction model. As a result, substantial parts of the Freesound infrastructure remain underutilized.

Importantly, this pattern cannot be attributed solely to conceptual design choices. Exploration is also shaped by practical constraints. In hardware-based systems, live performance contexts, or time-critical interactive setups, extensive querying and browsing may be impractical due to latency, efficiency, or cognitive load. Under such conditions, interaction often shifts from active database exploration toward looping, transforming, or recombining preselected material. What appears as limited exploration may therefore reflect performance-oriented optimization rather than a lack of ambition.

At the same time, newer interfaces do not automatically expand exploratory possibilities. The emergence of additional tools does not necessarily correspond to greater interactivity or broader engagement with metadata structures. This raises an open question: are limitations driven primarily by usability concerns, technical complexity, unfamiliarity with the API, or deliberate aesthetic choices? The answer likely varies across contexts, but the overall pattern suggests that exploration is frequently framed within a small subset of available retrieval strategies. Using Freesound's capabilities, such as precomputed audio descriptors, can also help reduce computational time, something important in many creative settings.

Freesound's heterogeneity further complicates this landscape. Variability in recording quality, metadata completeness, and licensing conditions introduces friction into exploratory workflows. While licensing or duration is often explicitly addressed in interface design, inconsistencies in audio quality or descriptive metadata are less frequently mitigated, potentially narrowing practical exploration even when technical affordances exist. Recent additions to the Freesound infrastructure –such as similarity spaces, category structures, richer metadata fields, and acoustic or perceptual descriptors– offer opportunities to support more multifaceted interaction models. Beyond expanding conceptual access, these mechanisms may also reduce computational overhead in creative contexts by enabling more targeted retrieval strategies. Leveraging such features could support forms of exploration that are both computationally efficient and semantically richer.

Even though the data collection methods may be constrained by poorly documented systems, personal projects, systems not using API keys, earlier high-traffic applications, and missing information for identifying certain API clients, they provide a broad and complementary, if partial, view of the ecosystem and its trends, with parts of the community potentially underrepresented. The circulation of open-source data across platforms provides a useful initial indication of how large open collections are approached, used, and leveraged for creativity and interaction.

Overall, our findings indicate that the exploratory potential of Freesound-based applications remains only partially realized. Despite considerable creative diversity, interaction paradigms frequently converge around limited retrieval modes. This suggests emerging opportunities for future work in designing interfaces that support more multifaceted, *open-ended* engagement with sound collections. Strengthening this connection between infrastructural capacity and interface design could enable more dynamic, transparent, and exploratory relationships with large and heterogeneous sound collections.

6 Ethical Standards

This research adheres to the ethical guidelines established by the NIME community. The study did not involve any human participants, and all data was collected ethically. The authors declare no financial or non-financial conflicts of interest.

Acknowledgments

This research was partially funded by the Generalitat de Catalunya (2025FI-100252, Joan Oró program), the IA y Música Cátedra (TSI-100929-2023-1, Cátedras ENIA 2022, SE Digitalización e IA, EU NGEU), and the IMPA project (PID2023-152250OB-I00, MCIU, AEI, co-funded by EU). We would also like to thank all developers who have created applications using Freesound.

References

[1] 2025. AI Movie Generator. https://websim.com/@frog/ai-movie-generator.
[2] 2025. Audio Forge. https://slashpaf.com/audioforge/. Retrieved 2026-02-01 from Slashpuf.
[3] Sam Aaron et al. 2012. Overtone. https://github.com/overtone/overtone. Retrieved 2026-02-01 from GitHub.
[4] Acoustica. 2017. Mixcraft 8. https://labs.freesound.org/apps/2017/01/01/mixcraft.html. Retrieved 2026-02-01 from Freesound Labs.
[5] Javi Agenjo. 2011. Freesoundscapes. https://labs.freesound.org/apps/2015/04/29/freesoundscapes.html. Retrieved 2026-02-01 from Freesound Labs.
[6] Javi Agenjo, Bram de Jong, and Frederic Font. 2014. Free Maschine! https://labs.freesound.org/apps/2015/04/27/free-maschine.html. Retrieved 2026-02-01 from Freesound Labs.
[7] Esteban Agosin. 2022. Imaginary Machinescapes. https://estebanagosin.cl/machinescapes.html. Retrieved 2026-02-01.
[8] Panagiota Anastasopoulou. 2022. *Audio Retrieval from Big Databases by Melodic Movements: A Virtual Agent for Live Coding*. Master's thesis. National and Kapodistrian University of Athens. https://doi.org/10.5281/zenodo.19699506
[9] Panagiota Anastasopoulou, Alastair Porter, and Frederic Font. 2025. The Freesound API: Advances in Audio Search and Retrieval. In *Proc. Web Audio Conference (WAC)*.
[10] Marguerite Barry and Gavin Doherty. 2017. How We Talk About Interactivity: Modes and Meanings in HCI Research. *Interacting with Computers* 29, 5 (2017), 697–714.
[11] Stefan Brunner. 2014. Elevator Music Generator. https://labs.freesound.org/apps/2015/04/28/elevatormusicgenerator.html. Retrieved 2026-02-01 from Freesound Labs.
[12] Erik P. Bucy and Chen-Chao Tao. 2007. The Mediated Moderation Model of Interactivity. *Media Psychology* 9, 3 (2007), 647–672. https://doi.org/10.1080/15213260701283269
[13] Jeff T Byrd. 2024. Earth Song. https://earthsong.world/. Retrieved 2026-02-01.
[14] CJ Carr, Daniel Lopez, Emilio Molina, Mónica Rikić, Lefteris Stamellos, Edu Fonseca, Xavier Favory, and Dmitry Bogdanov. 2017. Freesound Trip. https://labs.freesound.org/apps/2017/01/09/freesound-trip.html. Retrieved 2026-02-01 from Freesound Labs.
[15] Albin Andrew Correya. 2017. Perceptual Sound Browser. https://labs.freesound.org/apps/2017/08/07/perceptual-sound-browser.html. Retrieved 2026-02-01 from Freesound Labs.
[16] Paul Davies and the community. 2016. Ardour. https://labs.freesound.org/apps/2016/01/01/ardour.html. Retrieved 2026-02-01 from Freesound Labs.
[17] Ariane de Souza Stolfi, Miguel Ceriani, Luca Turchet, and Mathieu Barthet. 2018. Playsound. Space: Inclusive Free Music Improvisations Using Audio Commons. In *Proc. New Interfaces for Musical Expression (NIME)*. 228–233.
[18] Paul Dourish. 2001. *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press.
[19] Edward J. Downes and Sally J. McMillan. 2000. Defining Interactivity: A Qualitative Identification of Key Dimensions. *New Media & Society* 2, 2 (2000), 157–179.
[20] Gorka Egino. 2025. Soundpot. https://asmatzaile.codeberg.page/Soundpot/. Retrieved 2026-02-01 from Codeberg.
[21] Xavier Favory and Xavier Serra. 2019. Multi Web Audio Sequencer: Collaborative Music Making. In *arXiv Preprint arXiv:1905.06717*. arXiv:1905.06717
[22] Evan Feenstra. 2016. Beatpush. https://labs.freesound.org/apps/2016/04/01/beatpush.html, note = Retrieved 2026-02-01 from Freesound Labs.
[23] Miguel A. Fernández and Geraint A. Wiggins. 2020. Music Information Retrieval and Contemporary Music Creation. *Transactions of the International Society for Music Information Retrieval (ISMIR)* 3, 1 (2020), 126–136.
[24] Frederic Font. 2020. Freesound Loop Generator. https://labs.freesound.org/apps/2020/03/05/freesound-loop-generator.html. Retrieved 2026-02-01 from Freesound Labs.
[25] Frederic Font. 2021. SOURCE: A Freesound Community Sampler. https://labs.freesound.org/apps/2021/05/27/source-freesound-sampler.html. Retrieved 2026-02-01 from Freesound Labs.
[26] Frederic Font and Giuseppe Bandiera. 2017. Freesound Explorer: Make Music While Discovering Freesound!. In *Proc. Int. Web Audio Conference (WAC 2017)*.
[27] Frederic Font and Tony Martinez. 2017. Freesound Timeline. https://labs.freesound.org/apps/2017/11/23/freesound-timeline.html. Retrieved 2026-02-01 from Freesound Labs.
[28] Frederic Font, Gerard Roma, and Xavier Serra. 2013. Freesound Technical Demo. In *Proc. 21st Int. Conference on Multimedia (ACM)*. 411–412.
[29] Frederic Font and Álvaro Sarasúa. 2013. Minority Freeport. https://labs.freesound.org/apps/2015/04/29/minorityfreeport.html. Retrieved 2026-02-01 from Freesound Labs.
[30] Frederic Font and Stelios Togias. 2012. Free(Cc)It! https://labs.freesound.org/apps/2015/04/29/freccit.html. Retrieved 2026-02-01 from Freesound Labs.
[31] Austin Franklin, Dylan Burchett, and William Thompson IV. 2021. Composing and Improvising Using Sound Content-Based Descriptive Filtering. In *Proc. Web Audio Conference (WAC)*.
[32] Alexandre Gravel-Raymond. 2022. Shabda. https://labs.freesound.org/apps/2025/11/27/shabda.html. Retrieved 2026-02-01 from Freesound Labs.
[33] Behzad Haki. 2025. Freesound Rack. https://github.com/behzadhaki/freesound-rack. Retrieved 2026-02-01 from GitHub.
[34] Marc Hassenzahl and Noam Tractinsky. 2008. User experience – a research agenda. *International Journal of Design* 2, 1 (2008), 1–10. https://ijdesign.org/index.php/IJDesign/article/view/718/370
[35] Naoto Hieda. 2022. Hydra-Freesound-Auto. https://labs.freesound.org/apps/2022/02/10/hydra-freesound.html. Retrieved 2026-02-01 from Freesound Labs.
[36] Wouter Hisschemöller. 2018. Volca-Freesound. https://labs.freesound.org/apps/2018/12/28/freesound-volca.html. Retrieved 2026-02-01 from Freesound Labs.
[37] Andy Hunt and Marcelo Wanderley. 2002. Mapping Performer Parameters to Synthesis Engines. *Organised Sound* 7, 2 (2002), 97–108.
[38] Eugene Ie, Martin Rehn, Gal Chechik, Samy Bengio, and Dick Lyon. 2018. Large-Scale Content-Based Audio Retrieval from Text Queries. In *Proc. ACM Int. Conference on Multimedia Information Retrieval (MIR)*. ACM.
[39] Impronivers. 2023. Sound Show. https://impronivers.itch.io/sound-show. Retrieved 2026-02-01.
[40] Peter Ingwersen and Kalervo Järvelin. 2005. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer.
[41] Jordi Janer and Frederic Font. 2016. Hands-Free Sound Machine. https://labs.freesound.org/apps/2016/05/20/hands-free-sound-machine.html. Retrieved 2026-02-01 from Freesound Labs.
[42] John Johnston. 2020. GifDub. https://johnjohnston.neocities.org/. Retrieved 2026-02-01.
[43] Peder Jørgensen and Christian Schaanning. 2016. Soundly. https://labs.freesound.org/apps/2016/01/01/soundly.html. Retrieved 2026-02-01 from Freesound Labs.
[44] KeDiMouRa collective and Dimitri Aatos. 2017. Loskop Radio. https://labs.freesound.org/apps/2021/06/15/loskop-radio.html. Retrieved 2026-02-01 from Freesound Labs.
[45] Stuart Keith. 2016. Web Audio Sequencer. Retrieved 2026-02-01 from Freesound Labs.

[46] David Kirby. 2013. Plexure.Js. https://five23.github.io/plexure.js/index.html. Retrieved 2026-02-01 from GitHub.
[47] MLA Labs. 2025. The Playground. https://playground.mlalabs.xyz/. Retrieved 2026-02-01.
[48] Joseph Larralde. 2018. Freemix. https://labs.freesound.org/apps/2018/03/19/freemix.html. Retrieved 2026-02-01 from Freesound Labs.
[49] Quim Llimona. 2013. Music Contextualizr. https://labs.freesound.org/apps/2013/06/14/music-contextualizer.html. Retrieved 2026-02-01 from Freesound Labs.
[50] Alex MacLean and Greg Smith. 2017. Soundscape Generator. https://labs.freesound.org/apps/2017/12/01/soundscape-generator.html. Retrieved 2026-02-01 from Freesound Labs.
[51] Thor Magnusson. 2010. Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal* 34, 4 (2010), 62–73.
[52] Robert Manzke, Phillip Lamp, and Niklas Wantrupp. 2019. CTAG Strämpler. https://labs.freesound.org/apps/2019/05/02/strampler.html. Retrieved 2026-02-01 from Freesound Labs.
[53] Scott McCoid, Jason Freeman, Brian Magerko, Christopher Michaud, Tom Jenkins, Tom Mcklin, and Hera Kan. 2013. EarSketch: An Integrated Approach to Teaching Introductory Computer Music. *Organised Sound* 18, 2 (2013), 146–160.
[54] Sharon Oviatt. 1999. Ten Myths of Multimodal Interaction. *Commun. ACM* 42, 11 (1999), 74–81.
[55] Tayjo Padmini Vaduru. 2018. *Moodscape Generator: Automated Generation of Soundscapes.* Ph. D. Dissertation. Queen Mary University of London.
[56] Sergi Pastor. 2024. Hathor. https://hathor.uno. Retrieved 2026-02-01.
[57] Andy Pearce. 2019. Timbral Explorer. https://labs.freesound.org/apps/2019/01/30/timbral-explorer.html. Retrieved 2026-02-01 from Freesound Labs.
[58] German Puerto Rodriguez. 2026. EthnoSphere. https://trekar99.github.io/ethnosphere/. Retrieved 2026-02-01 from GitHub.
[59] Antonio Ramires and Frederic Font. 2019. FreesoundSimpleSampler. https://labs.freesound.org/apps/2019/11/14/juce-examples.html. Retrieved 2026-02-01 from Freesound Labs.
[60] Gerard Roma, Perfecto Herrera, and Xavier Serra. 2009. Freesound Radio: Supporting Music Creation by Exploration of a Sound Database. In *Workshop on Computational Creativity Support (CHI2009)*. Citeseer.
[61] Gerard Roma and Xavier Serra. 2015. Music Performance by Discovering Community Loops. In *Proc. Web Audio Conference (WAC)*.
[62] Gerard Roma and Xavier Serra. 2015. Querying Freesound with a Microphone. In *Proc. Web Audio Conference (WAC)*.
[63] Tefko Saracevic. 1997. The Stratified Model of Information Retrieval Interaction. *Journal of the American Society for Information Science* 48, 6 (1997), 509–525.
[64] Daniele Scarano and Chromahelix. 2016. Infinite Orcherstra. https://labs.freesound.org/apps/2016/05/26/infinite-orchestra.html. Retrieved 2026-02-01 from Freesound Labs.
[65] Stagecraft Software. 2015. Cloud Browser. https://labs.freesound.org/apps/2015/06/01/cloud-browser.html. Retrieved 2026-02-01 from Teixeira.
[66] Le Sound. 2018. AudioTexture Free. https://labs.freesound.org/apps/2018/11/29/audiotexture-free.html. Retrieved 2026-02-01 from Freesound Labs.
[67] Soundtrap. 2018. Soundtrap. https://labs.freesound.org/apps/2019/01/30/soundtrap.html. Retrieved 2026-02-01 from Freesound Labs.
[68] Dappledark Studios. 2022. Lockdown Composer 7. https://labs.freesound.org/apps/2022/12/20/lockdown-composer.html. Retrieved 2026-02-01 from Freesound Labs.
[69] FollowWeb Development Team. 2024. FollowWeb. https://visualise.music/. Retrieved 2026-02-01.
[70] Paulo Teixeira, Gilberto Bernardes, and Matthew Davies. 2019. Fostering the Database in Audio Production Environments by Affect Soundscape Retrieval. In *Hidden Archives, Hidden Practices: Debates about Music-Making*. 220–233.
[71] Miles Thorogood, Philippe Pasquier, and Arne Eigenfeldt. 2012. Audio Metaphor: Audio Information Retrieval for Soundscape Composition. In *Proc. Sound and Music Computing Conference (SMC)*. 277–283.
[72] Robert H. Tubb and Simon Dixon. 2014. The Divergent Interface: Supporting Creative Exploration of Parameter Spaces. In *Proc. Int. Conference on New Interfaces for Musical Expression.* https://doi.org/10.5281/zenodo.1178967
[73] Luca Turchet and Mathieu Barthet. 2018. Jamming with a Smart Mandolin and Freesound-based Accompaniment. In *Proc. Conf. Open Innovations Association (FRUCT)*. 375–381.
[74] Luca Turchet, Marco Carraro, and Matteo Tomasetti. 2023. FreesoundVR: soundscape composition in virtual reality using online sound repositories. *Virtual Reality* 27 (2023), 903–915.
[75] George Tzanetakis. 2002. *Manipulation, Analysis, and Retrieval Systems for Audio Signals.* PhD thesis. Princeton University.
[76] Gabriel Urbain, Christian Frisson, Alexis Moinet, and Thierry Dutoit. 2016. A Semantic and Content-Based Search User Interface for Browsing Large Collections of Foley Sounds. In *Audio Mostly: A Conference on Interaction with Sound*. 272–277.
[77] Photo App World. 2025. Audio Editor App. https://play.google.com/store/apps/details?id=com.photoappworld.audio.professionalaudioeditor. Retrieved 2026-02-01 from Google Play.
[78] Anna Xambó, Gerard Roma, Alexander Lerch, Mathieu Barthet, and György Fazekas. 2018. Live Repurposing of Sounds: MIR Explorations with Personal and Crowdsourced Databases. In *Proc. New Interfaces for Musical Expression (NIME)*. 364–369.
[79] Xavier Bonfill. 2014. idRadio. https://labs.freesound.org/apps/2015/04/08/idradio.html. Retrieved 2026-02-01 from Freesound Labs.
[80] William Y. Arms Yang. 2000. Evaluation of Digital Libraries. In *Digital Libraries*. MIT Press, Chapter 10. https://www.cs.cornell.edu/wya/DigLib/text/Chapter10.html