Authoring augmented soundscapes with user-contributed content

Jordi Janer*
Music Technology Group
Universitat Pompeu Fabra

Gerard Roma†
Music Technology Group
Universitat Pompeu Fabra

Stefan Kersten‡
Music Technology Group
Universitat Pompeu Fabra

ABSTRACT

Augmented reality audio is an area still not sufficiently explored. In this article we address the creation of soundscapes to augment the acoustic information in a physical location. In particular, we focus on authoring tools that make use of user-contributed content. To facilitate the authoring process, our tool integrates the access to Freesound.org, an online repository with more than 120,000 sounds under a Creative Commons license. The sound search combines the traditional text-query with content-based audio classification. The automatic classification allows searching according to a taxonomy of environmental sounds (e.g. drip, impact, wind, etc.). Finally, we implemented a complete augmented soundscapes system that , in an autonomous and continuous manner, spatializes virtual acoustic sources in a geographic location.

1 INTRODUCTION

The use of audio features in augmented reality (AR) is still scarce, and today only a few applications are available to the general public (e.g. soundwalks [11] provide guided tours on popular tourist cities). At the same time, it is important to observe how more and more people cover their transportation needs in modern cities permanently equipped with headphones and portable music players or mobile phones.

When thinking of sound in AR, also referred to as ARA (Augmented Reality Audio) [7, 6], a difference should be made between speech, music and environmental sounds. From our perspective, sound information can go beyond verbalizing the textual or graphic content with a text-to-speech (TTS) synthesizer, as typically found in visual AR applications.

We are interested in how environmental sounds can augment the existing soundscape of a real location. Our system is not regarded only as a location-based media service, since sounds are meant to augment physical world elements (historical buildings, public spaces, etc.) with acoustic information that is not present. In game environments, most of the sound content is diegetic (the narrated events occur), and principally synchronized with game interactive events. In contrast, we envision an augmented soundscape as an autonomous sound generation system, whose primary mission is to produce a sonic ambiance that spatially covers a given geographic location. Our strategy is to place virtual sound objects in a geographic area. Users can explore in an interactive manner the soundscape generated in real-time.

Within this context, we can think of several potential use-cases. It can improve immersion in mixed reality (e.g. virtual 3D models) as in [1]. Also from a mixed-reality perspective, the generated soundscape can be employed as an auditory display for real-world data, e.g. sonifying real-time traffic or weather information. Specifically in the AR context, we can think of tourism guides (e.g. in open air museums), where our system could allow users to visit a city listening to a soundscape of a past age (e.g gladiators sounds when visiting the Coliseum in Rome, a supporters chants in the surroundings of a football stadium, people in a demonstration when visiting Berlin's Wall, etc.). More creative usages could be adapted for entertainment or artistic scenarios (e.g. in amusement parks or art festivals), as found in [8] to create musical paths. Another area of interest is found in the accessibility domain, for example [5] use sounds to help the navigation of blind people inside a building.

In this paper, we describe an authoring tool that allows the creation of an augmented soundscape from a user-contributed sound repository [12]. First, we briefly introduce the complete system of augmented soundscape generation in section 2. The steps involved in the authoring process are addressed in section 3. Finally, a use-case of the system is presented in section 4.

This authoring tool is desktop-based, and thus designed to be used by a sound designer during the content creation workflow. In the current system, users have a limited control over the sound content reproduced in the physical world, which is mainly bound to user's position and orientation. Although it is beyond the scope of this paper, a future direction is to consider mobile devices for "on-site" authoring. This scenario will present new possibilities and challenges both in authoring and user interaction.

2 AUGMENTED SOUNDSCAPES

2.1 User-contributed content

With the rapid growth of social media, large amounts of sound material are becoming available through the web every day. In contrast with traditional audiovisual media, networked multimedia environments can exploit such a rich source of data to provide content that evolves over time. These multimedia environments follow the trend towards user-centered technologies and user-generated content[1]. We present tools around Freesound, an online repository of user-contributed sounds. Created in 2005, it stores today more than 120,000 sounds under a Creative Commons license, with over two million registered users.

Considering user-contributed content for augmented reality brings some advantages. First as a way of keeping the content continuously up-to-date, since as long as the community is alive and active, new sounds will be uploaded. Further, for some applications it is necessary to provide evolving content, in order to attract users that access the application repeatedly.

2.2 System overview

A main characteristic of the presented work is the authoring with user-contributed content. However, to illustrate the relevance of this approach, we describe here the complete augmented soundscapes system, which consists of three components: an authoring tool, a real-time soundscape generation engine, and a server-side platform that supports multiple simultaneous clients.

Figure 1 shows the block diagram of the proposed architecture. In the authoring stage, the designer (bottom right) creates a new soundscape, taking content from Freesound. The real-time *audio generation engine* outputs a stereo signal taking into account the listener position inside the soundscape area, which is managed

*e-mail: jordi.janer@upf.edu
†e-mail:gerard.roma@upf.edu
‡e-mail:stefan.kersten@upf.edu

[1] Some popular online repositories are among others Flickr.com (photos), YouTube (video), Google 3D-Warehouse (3D models), Freesound (sounds).

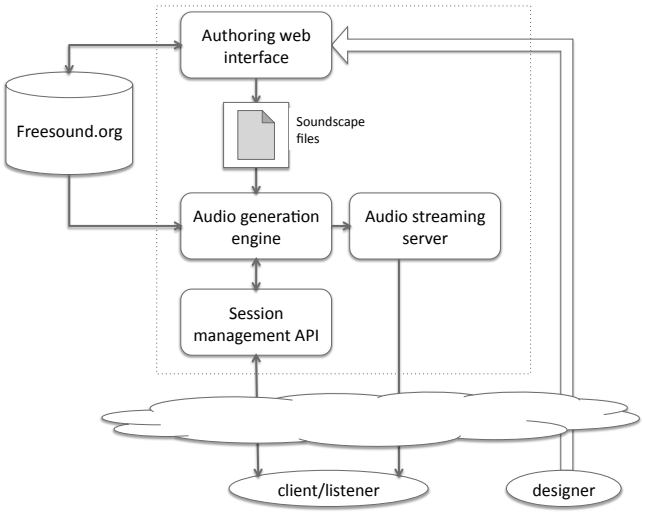

Figure 1: Block diagram showing the architecture on top: authoring (with Freesound access), generation engine and streaming server. At the bottom the two user topologies: designer and client/listener.

Figure 2: Example of a soundscape map. Global area in blue, zones in green and concepts in yellow.

by the *session management* module. Finally, an *audio streaming server* sends the personalized soundscape to the client application by means of an MP3 stream.

An important characteristic of this system is that the synthesis engine runs in an autonomous manner, generating a stochastic sequence of sounds for each virtual object. The parameters of the synthesis engine allow to specify different acoustic characteristics such as the sound density, simultaneous sounds, loudness, etc. Regarding the spatialization of the synthesized sounds, the current system only supports 2D panning. Each listener can localize sounds coming from different locations, distance is modelled also as a point source law (6dB/double distance), and low-pass filtering is applied according to the orientation to distinguish among sound coming from the front or from the back.

3 AUTHORING STAGE

A principal objective of the presented system is to facilitate the authoring process from a designer point of view. In the literature, we find similar approaches [14], where the authoring tool integrates several panels containing a file browser, a waveform display and a map for positioning the virtual sound sources. Compared to other approaches, our principal feature is the direct access to a large sound repository, which can make the sound design process more efficient. Also, the developed synthesis engine already solves all technical aspects of the automated sound generation, therefore the designer can focus on the authoring steps. Three steps are involved: map definition, sound content retrieval and synthesis parameter settings. Next, we describe this process in more detail.

3.1 Geographic information

Our scenario involves the virtual sonification of a real geographical area. The geographic information is specified on a 2D coordinate system, and is structured in three levels: global, zone, and sound concepts. The global level defines the limits of the sonified area [13, 10]. A zone is defined by a polygon, which is populated by multiple sound concepts located in different positions. Each sound concept refers to a virtual acoustic source, and as described in the next section, will be associated with multiple sound samples.

In terms of the role of the acoustic source, a sound concept can be of three types: a) a fixed point source, b) a point source randomly located in a area or c) as an area source that is present in the whole area. The designer labels the sound concept, and defines its location and source type.

As shown in figure 2, to create the map we use the world browser Google Earth, which allows to design and export the designed map in KML format. Other applications that exporting standard KML could be equally employed.

For interoperability, this data is stored in KML format [3]. KML (*Keyhole Markup Language*) is an open format based on XML used to describe geographic data developed by Google. We use *Placemark*, a tag with associated geometry, to declare zones and concepts. *Placemark* has name, description and two types of geometry elements, *Point* models point sound sources and *Polygon* models area sound sources and zone geometries. Additionally, KML allows to create *Folder*s, container elements that we use to declare the *global soundscape*, which in turn contains the *zone* folders and each zone folder contains a collection of *sound concept* placemarks. Geographic location is stored as longitude and latitude in degrees.

3.2 Sound content retrieval

Once the sound concept locations are specified in the KML file, the following step associates sound samples to each sound concept.

In order to facilitate the sample search, we directly access Freesound through its web API. Searching in large unstructured audio databases such as Freesound has inherent limitations, due to the lack of organized metadata. The typical sound search functionality in Freesound is achieved by means of text-queries, searching the database for user-contributed tags and descriptions. As a consequence, poorly annotated sounds will be more difficult to find. Therefore, we argue that a combination of a taxonomy of sounds and automatic classification may improve both the querying and the ranking processes.

As suggested by Gaver [2], environmental sounds can be categorized according to their acoustic properties to identify the physical sources of sound production. This is referred to as *ecological acoustics* in the literature. Gaver defined a hierarchical taxonomy with three source categories (solid, liquid and gas), and eleven sonic events: deformation, impact, scraping and rolling (for solids); explosion, whoosh and wind (for gas); and, drip, pour, splash and ripple (for liquids). We adopt this taxonomy for searching on environmental sounds.

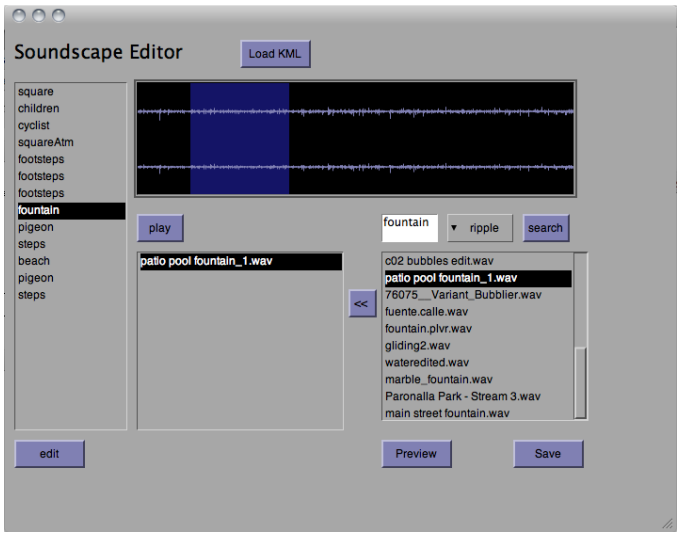

Figure 3: GUI of the synthesis parameters settings for each sound concept. The right panel contains a list of retrieved samples.


```
- <Placemark>
    <name>children</name>
  - <LookAt>
      <longitude>2.173519684377638</longitude>
      <latitude>41.40316147827902</latitude>
      <altitude>0</altitude>
      <heading>1.447342568145065</heading>
      <tilt>3.158027485318621</tilt>
      <range>448.8122098079919</range>
      <altitudeMode>relativeToGround</altitudeMode>
      <gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>
    </LookAt>
    <styleUrl>#msn_ylw-pushpin6</styleUrl>
  - <Point>
      <altitudeMode>clampToGround</altitudeMode>
      <gx:altitudeMode>clampToSeaFloor</gx:altitudeMode>

        <coordinates>2.173021098769079,41.4030310155222,0</coordinates>
    </Point>
  - <ExtendedData>
    - <SchemaData schemaUrl="#concept_id">
        <SimpleData
          name="conceptGeometry">44.723033973184,0,0,0</SimpleData>
        <SimpleData name="gain">1</SimpleData>
        <SimpleData name="psRandomGeneration" />
        <SimpleData name="continuous">0</SimpleData>
        <SimpleData name="multipleGenerativePath">1</SimpleData>
        <SimpleData name="probability">0</SimpleData>
        <SimpleData name="ar">1</SimpleData>
        <SimpleData name="recordingDistance">1e-12</SimpleData>
        <SimpleData name="listenedArea">1e-12</SimpleData>
        <SimpleData name="clone" />
      </SchemaData>
    </ExtendedData>
  </Placemark>
```


Figure 4: Example of segments in the KML file containing geographical data and synthesis parameters for each sound concept.

In a previous paper [9], we introduced a method for an automatic classification of audio events that is based on this taxonomy. A model for each category was trained with a Support Vector Machine algorithm, taking a set of audio features (e.g. MFCC) as input data. In the same article, we describe how we carry out a primary automatic classification of sounds into three main categories speech, music and environmental. The classification into the eleven environmental sound categories only considers those sounds that fall into the environmental class.

To evaluate the method, we carried out a user experiment that involved environmental sound retrieval in Freesound. Users had to validate the search results obtained with a text query, and those obtained when ranking with the content-based classifier. The precision obtained (number of relevant files divided by the number of retrieved files) in the content-based method was 77.22%, against the 57.54% obtained by the text-only based method currently available in Freesound.

In the authoring interface, the retrieval method combines content-based information with text queries. Queries are composed of two terms. The first term is entered by the user and is expected to define the object producing the sound. The second term is chosen from the taxonomy and represents an event produced by the object. For example in order to find sounds of a door slam, the user would enter ”door” and select ”impact” from the taxonomy. The first term is used for a regular text search in the database. Due to the nature of user-generated content, such text searches may return many unwanted sounds, e.g. a long field recording that contains the word ”door” in the description). Hence, the content based classifier is used to filter and rank the results specifically for the soundscape design application. For each sound, the content-based classifier determines the most probable class from the sound events taxonomy with an associated probability. The class is used for filtering the results, and the probability is used to rank them.

To integrate content-based audio retrieval in the authoring workflow, we developed a GUI prototype programmed in SuperCollider software environment [15], which imports the KML with the list of sound concept labels. For example, figure 3 shows a list of sounds retrieved from

3.3 Synthesis parameters

The synthesis of a sound concept is based on sample concatenation, and its temporal evolution is driven by a graph model. The graph model ensures an autonomous and non-repetitive generation. It is configured with a few control parameters (e.g. concept probability, regular vs arrhythmic triggering, randomness, or number of simultaneous samples). In other words, each graph model defines the sequencing of a number of events (samples). A detailed description of the sound synthesis engine is found in [10].

The soundscape editor loads a KML with placemarks that define the location of sound concepts. For each concept several parameters (e.g. loudness, density, etc.) can be edited. The process then consists in searching sounds and editing segment boundaries. Suitable segments are added as sound events of a particular concept (see Figure 3).

Finally, to export all this data we store two separate files: an extended KML file with localization and parameters (as described in [10]), and a audio files dataset in XML format with links to Freesound content. Figure 4 shows an example of the KML segments.

4 Application

The introduced system was originally developed in the context of virtual reality. Specifically, we developed a set of modules to add a customized sonification of a virtual island in Second Life. Its application in the context of augmented reality is however direct. The required adaptations were only related to the coordinates system. In the virtual worlds, coordinates (x, y) were referenced in meters to the island position. In the augmented reality, geographical coordinates are absolute values of longitude and latitude expressed in degrees. When importing the KML file in the authoring tool, we convert the longitude and latitude pairs to (x, y) pairs relative to a reference point, and express the values in meters.

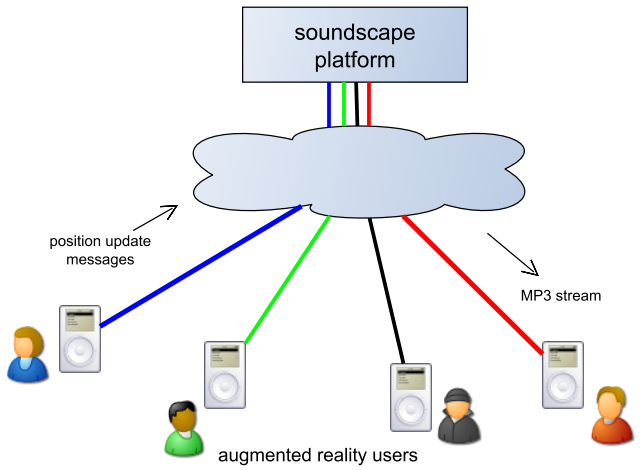

Figure 5: A server architecture allows multiple clients to send position updates as HTTP messages and obtain a personalized soundscape.

4.1 Server-side platform

As shown in Figure 5, the system supports multiple independent listeners that send position updates and receive an audio stream. Interaction with the soundscape generation system running on the server is done through a web API. This allows client applications to add listeners and obtain personalized streams given the coordinates of each listener. A web server, implemented using the Twisted[2] framework, provides an HTTP interface for external web clients, which translates to OSC (OpenSoundControl) calls for controlling the streaming server. The web server is also responsible of maintaining client sessions. In the current implementation, a fixed pool of streaming URLs is used, and so the number of clients is bounded. Once the listener has been added a session ID is returned. Then the client application can make requests to "position" and "rotation" resources, with the position and rotation as parameters, which are translated to the corresponding OSC calls.

The synthesized sound corresponding to a given listener in the synthesis server is streamed from a unique URL. Each output channel pair in the audio engine is connected to one instance of the real-time audio streamer *darkice*[3] through the audio application interconnection server *jack*[4]. These tools are widely used on Linux environments for managing low-level audio routing. The streaming application *darkice* in turn creates a mount-point corresponding to a listener id in an *icecast*[5] streaming server and streams the listener output produced by the soundscape generation in the MPEG1 Layer 3 format.

4.2 Client Mobile implementation

A simple Javascript client application running on the mobile is used to connect to the soundscape generation server. Note that the audio synthesis process takes places in the server, and is transmitted to the client as an MP3 audio stream. The client makes use of HTML5 technologies. On one hand, the geo-location API allows getting the current location using a mobile device. The coordinates are sent to the server for position updates. On the other hand, an HTML5 audio object plays the generated stream. The graphical interface simply shows a Google map of the current location.

[2] `http://twistedmatrix.com`
[3] `http://code.google.com/p/darkice/`
[4] `http://jackaudio.org/`
[5] `http://www.icecast.org/`

5 Conclusion

In this paper, we presented a system for producing augmented soundscapes, putting special focus on its authoring tool. It integrates Freesound, a user-contributed repository used as sound library for the sample-based synthesis engine. We showed that this collaborative authoring resource, when combined with automatic audio classification can simplify the sound asset management.

In the developed prototype, simultaneous mobile clients can communicate its position to the soundscape generation engine, receiving a personalized MP3 audio stream according to its position.

Regarding the future directions in augmented reality audio, we think that technologies are in general sufficiently mature. However, the main challenge is more on the side of studying user-interaction aspects and to implement applications that can benefit from the auditory information.

Acknowledgements

The authors wish to thank Mattia Schirosa. This research has been partially supported by TECNIO - Generalitat de Catalunya and the TECLEPATIA Project (TEC2010-11599) by the Spanish Ministry of Science.

References

[1] N. Finney. Autonomous generation of soundscapes using unstructured sound databases. Master's thesis, Universitat Pompeu Fabra, 2009.

[2] W. W. Gaver. What in the world do we hear? an ecological approach to auditory event perception. *Ecological Psychology*, 5:1–29, 1993.

[3] Google. Kml format. `http://code.google.com/apis/kml/`, 2010.

[4] Google. Google earth. `http://www.google.com/earth`, 2011.

[5] Y. Lasorsa and J. Lemordant. An Interactive Audio System for Mobiles. In *127th AES Convention*, New York, États-Unis, Oct. 2009.

[6] J. Lemordant and Y. Lasorsa. Augmented Reality Audio Editing. In *Proceedings of 128th AES Convention*, page paper 8143, London, Royaume-Uni, May 2010.

[7] E. Mynatt, M. Back, R. Want, and R. Frederick. Audio Aura: Leight Weight Audio Augmented Reality. In *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST 97*, New York, 1997.

[8] S. Park, S. Kim, S. Lee, and W. Yeo. Composition with path: Musical sonification of geo-referenced data with online map interface. In *Proceedings of the International Computer Music Conference, New York, NY, USA*, 2010.

[9] G. Roma, J. Janer, S. Kersten, M. Schirosa, P. Herrera, and X. Serra. Ecological acoustics perspective for content-based retrieval of environmental sounds. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.

[10] M. Schirosa, J. Janer, S. Kersten, and G. Roma. A system for soundscape composition, generation and streaming . In *Proceedings of the CIM - Colloquium for Musical Informatics, Turin, Italy*, 2010.

[11] Soundwalk. Soundwalk. `http://soundwalks.com/`, 2010.

[12] Universitat Pompeu Fabra. Freesound.org. `http://www.freesound.org`, 2005. Repository of sounds under the Creative Commons license.

[13] A. Valle, V. Lombardo, and M. Schirosa. *Auditory Display 6th International Symposium, CMMR/ICAD 2009, Copenhagen, Denmark, May 18-22, 2009. Revised Papers*, volume 5954, chapter Simulating the Soundscape through an Analysis/Resynthesis Methodology, pages 330–357. Springer, Berlin, 2010.

[14] B. N. Walker and K. Stamper. Building Audio Designs Monkey: An audio augmented reality designer's tool. In *Proceedings of the International Conference on Auditory Display (ICAD2005)*, Limerick, Ireland, 2005.

[15] S. Wilson, D. Cottle, and N. Collins, editors. *The SuperCollider Book*. The MIT Press, Cambridge, Mass., In Press.