

Sonification of Experimental Parameters as a New Method for Efficient Coding of Behavior

Andrea Ravignani¹, W. Tecumseh Fitch²

Department of Cognitive Biology, University of Vienna, Vienna, Austria.

¹andrea.ravignani@univie.ac.at, ²tecumseh.fitch@univie.ac.at

Abstract

Cognitive research is often focused on experimental condition-driven reactions. Ethological studies frequently rely on the observation of naturally occurring specific behaviors. In both cases, subjects are filmed during the study, so that afterwards behaviors can be coded on video. Coding should typically be blind to experimental conditions, but often requires more information than that present on video. We introduce a method for blind-coding of behavioral videos that takes care of both issues via three main innovations. First, of particular significance for playback studies, it allows creation of a “soundtrack” of the study, that is, a track composed of synthesized sounds representing different aspects of the experimental conditions, or other events, over time. Second, it facilitates coding behavior using this audio track, together with the possibly muted original video. This enables coding blindly to conditions as required, but not ignoring other relevant events. Third, our method makes use of freely available, multi-platform software, including scripts we developed.

Keywords. Sonification, blind coding, observational data, scoring behavior, experimental condition, behavioral observation, animal behavior, playback experiment, Python, ELAN.

Introduction

Experimental research in behavior and cognition often follows a typical methodological pattern. While a subject is exposed to several conditions, its reactions are videotaped. Afterwards, the researcher scores these data by blind coding specific behaviors from the video.

We developed a method that allows for an easy and more ergonomic video coding of behaviors and reactions, particularly suited for playback studies. While the person coding the video is busy with her visual task, she may need further information other than that visible in, or audible from, the original recording. Similarly, when a video is to be coded blindly, and its audio track must therefore be silenced, the coder nonetheless needs temporal guidance on what specific behaviors to code and at what point in time. In our approach, such information is broadcasted to the coder by the occurrence of specific sounds, played synchronously with the video.

Methods

The crucial innovation in our approach is to have an audio track where all aspects of the experiment relevant to video coding are mapped to sound (that is to say, “sonified”). Here we outline the step-by-step procedure that can be used to achieve this auditory-aided video coding (see Figure 1). The emphasis is on the methodology and the conceptual approach used, rather than the specific pieces of software employed. Therefore, in the rest of the section, the general description of each step is followed by a short description of the specific software solution we chose. Our choices, however, do not preclude alternative solutions.

1. While filming, the experimenter keeps a log of the events that, not being present on video, she would still like to include at the time of coding. Alternatively, the experimenter may use experimental software that automatically keeps a log with time and type of experimental conditions. For example, on Mac OS X platforms, Playback 3.0 [1] offers such a possibility for playback experiments (see reference [1] for download instructions), and its output are plain txt files. In any case, a log can be thought of as a list of different events whose time of occurrence is specified.

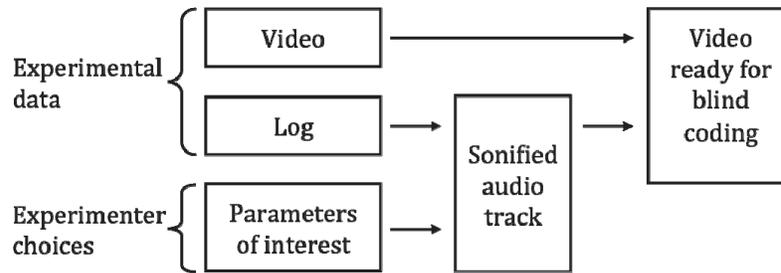


Figure 1. General procedure from experimental data to video annotation. The log of events collected during the study, together with the sounds the researcher decides to associate with particular parameters of interest, gives the sonified soundtrack of the experiment. This gets combined with the video and it is ready to be coded.

2. The experimenter decides what sound to associate with which type of event. Our first choice, dictated by perceptual saliency, has fallen on pure tones that differ by more than one octave in frequency. However, alternative solutions are easily implementable.
3. The log of events is transformed into an audio track, where each type of sound has a different meaning. Therefore the experiment has an “auditory counterpart” associated, spanning its entire duration in real time (see Figure 2). This is accomplished using a Python (Python 2.x, www.python.org) script we wrote, whose source code is available from the first author upon request. Python is a high-level, multi-platform programming language, built-in in Mac OS X and also available for download to Windows and Linux users. In particular, our script only makes use of libraries already included in Python, therefore avoiding additional installations.

What follows is the pseudocode on which our sonification script is based, enabling interested researchers to create an alternative version of the script using their preferred programming language.

- Create a silent audio file spanning the entire duration of the experiment.
 - Create a different sound for each relevant event.
 - For each time step in the audio file.
 - If an event requiring sonification is present.
 - Insert the corresponding sound in the audio file.
4. Audio and video are easily synchronized by manually trimming the onset of the first event in the video. For the case of playback experiments, this is equivalent to trimming the video up to the first stimulus onset. Synchronization accuracy depends on the quality and accurateness of the event log (see point 1). In studies requiring high synchronization accuracy, researchers should opt for a computer-based log-keeping system. An upper bound for the precision of the synchronization, on the other hand, is determined by the video frame rate. Trimming can be accomplished using any video editing software. iMovie (included in Mac OS X, our choice for testing the system), Windows Movie Maker (built-in in Windows) and Apple QuickTime (both platforms) are, for instance, uncomplicated, free solutions.
 5. Videos can be coded using a wide array of available software. We tested our system using the freely available video annotation software ELAN [2-3], which runs on Windows, Mac OS X and Linux platforms (see reference [2] for download). In ELAN, when simultaneously importing audio and video files, their onsets get automatically aligned. Therefore, once the video has been trimmed (see point 4), the audio-video synchronization is straightforward and automatic.
 6. An additional Python script (also available on request) can be used to check whether the behaviors and reactions coded satisfy the conditions originally imposed by the researcher and dictated by the audio

track. For instance, if behaviors are supposed to be coded within a given time window, the script can trim, discard or leave unaltered the annotations that exceed that time span, depending on the needs of the study. Additionally, when data from more than one coder is present, our script calculates the inter-rater agreement, both as a naïve agreement coefficient (joint probability) and using Cohen's kappa statistic [4].

In the next sections, we outline two possible research contexts where our method is particularly suitable.

Playback Experiments

In a playback experiment, different categories of sounds are played to an infant or a non-human animal. The subject's differential reactions among stimuli types are used to investigate aspects of her auditory cognition, such as discrimination capabilities, individual recognition, generalization within classes, etc.

While coding the video recording of the experiment, the researcher needs to know when a stimulus was played, so to look for reactions to code. However, to avoid biasing the coding, she must be blind to the specific sound played. Therefore, the original audio track is removed from the video and substituted with placeholder sounds. These (pure sine waves at different frequencies) signal when the playback started, when the coder can actually begin coding reactions and when no coding is allowed anymore.

This method is currently used at the University of Vienna (together with ELAN [2-3]) to code behavior in cognitive experiments in several primate species.

Observations of Naturally Occurring Behavior

Researchers in ethology usually focus on naturally occurring behaviors. Either observations are coded in real time or a focal subject (or a group of individuals) is filmed and the behavior coded at a later time. In the second case, several environmental and social phenomena may escape the camera, but are recorded by the researcher at the site. Such events may include, for instance, fights among other members of an animal group, extraneous human interactions, appearance of a predator, etc. Our method offers a way of integrating, at the time of coding, this manually recorded information with the video.

Conclusion

While coding specific behaviors from a video, two types of information are important. On the one hand, there are sounds that were recorded in the original video but may need to be concealed from the blind coder. On the other hand, there may be information not present in the video that is nonetheless essential while coding. The method

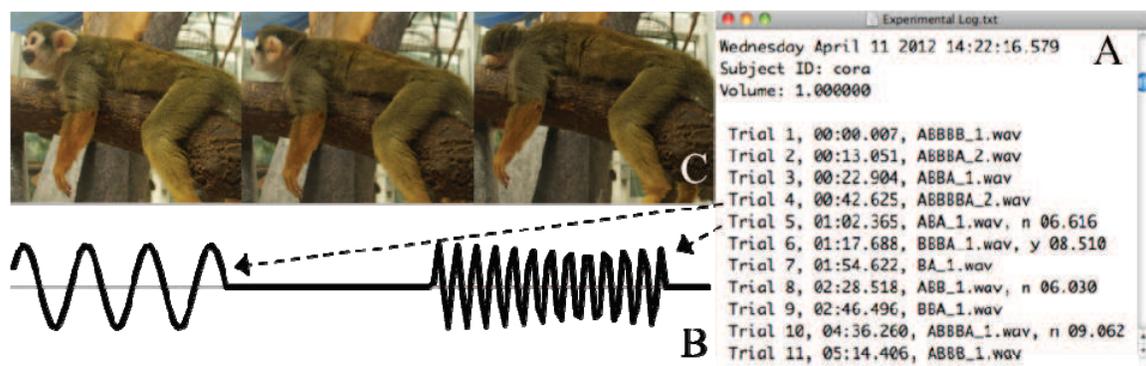


Figure 2. Sketch of the sonification procedure. Using a Python script, each event in the experimental log (A) is transformed into one or more sounds chosen by the researcher. The resulting soundtrack (B) is joined with the original video (C) and these are played simultaneously during behavioral coding. (Squirrel monkey pictures by Markus Boeckle)

we present here offers a simple solution to both issues at once. Information relevant to the experiment is sonified, that is, experiment-relevant events are transformed into sounds. This stream of sounds is then substituted to, or flanked by, the sound track of the original video. This allows coding blind to experimental conditions while keeping track of other events relevant to the study. We provided two practical examples of the many possible applications that this new method has in the study of human and non-human animal behavior and cognition. Finally, we show how the entire procedure, from performing the experiment to annotating behaviors, can be performed using freely available software.

Acknowledgments

The first author is supported by ERC Advanced Grant 230604 SOMACCA awarded to T. F. We thank Drasko Boko, Sarah Deventer, Maxime García, Bruno Gingras, Mauricio Martins, Stephan Reber and Ruth Sonnweber for testing the system and providing useful comments.

References

1. Jones, R., Fitch, T. (2009). Playback 3.0. Copyright 2009. Download requires git (multi-platform and freely available at <http://git-scm.com/>) and typing the following command: `git clone git://playback.git.sourceforge.net/gitroot/playback/playback` Accessed September 2011.
2. ELAN Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands. <http://www.la-mpi.eu/tools/elan/> Accessed March 2012.
3. Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., Sloetjes, H. (2006). ELAN: a Professional Framework for Multimodality Research. *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*, 1556-1559.
4. Cohen, J. (1960) A coefficient for agreement for nominal scales. *Education and Psychological Measurement* **20**, 37–46.