TAPESTREA: A New Way to Design Sound

Ananya Misra
Princeton University
35 Olden St.
Princeton, NJ 08540, USA
amisra@cs.princeton.edu

Ge Wang
Stanford University
660 Lomita Dr.
Stanford, CA 94305, USA
ge@ccrma.stanford.edu

Perry R. Cook
Princeton University
35 Olden St.
Princeton, NJ 08540, USA
prc@cs.princeton.edu

ABSTRACT

TAPESTREA is a sound design and composition framework that facilitates the creation of new sound from existing digital audio recordings, through interactive analysis, transformation and re-synthesis. During analysis, sound templates of different types are extracted using a variety of techniques. Each extracted template is transformed and synthesized independently, allowing specialized transformations on each template based on its type. The user interacts with TAPESTREA via a set of graphical interfaces that offer parametric control over every stage of analysis, transformation and re-synthesis. Synthesis is further controlled through ChucK scripts. These combined techniques form a workbench for completely transforming a sound scene, dynamically generating soundscapes, or creating musical tapestries by weaving together transformed elements from different recordings. Thus, TAPESTREA introduces a new paradigm for composition, sound design, and sonic sculpting tasks.

Categories and Subject Descriptors

D.0 [**Software**]: General

General Terms

Design, Experimentation

Keywords

Sound design, composition, audio, multimedia, signal processing, real-time, open source software

1. INTRODUCTION

A sound designer or artist manipulating existing digital audio for purposes such as musical composition or sound scene creation often encounters difficulties because the existing sounds are not exactly as desired. A single recording is unlikely to have all desired sounds; unwanted sounds may overlap the wanted parts; the wanted part may not have the exact desired frequency, length, or other quality. Tools for transforming sounds are constrained either in the range of sounds to which they apply or in the manipulation paradigms and variety of results they offer. TAPESTREA aims to surpass these limitations by presenting a unified framework for creating new sound from any combination of existing audio, with expressive freedom in selecting both *what* to re-use and *how* to re-use it.

Given one or more recordings, TAPESTREA provides well-defined means to:

- Identify points of interest in a sound and extract them into re-usable *templates*
- Transform individual templates independently of the background and/or other events
- Continually re-synthesize background sound textures
- Controllably place event templates over backgrounds, using a novel graphical user interface and/or scripts written in the ChucK [6] audio programming language

In this way, it provides a new way to completely transform a sound scene and to compose and design sound by combining elements from different recordings [4].

2. TAPESTREA

TAPESTREA manipulates sound in several phases (see Figure 1). In the analysis phase, desired parts of a recording are interactively extracted into templates of different types. Pitched components are saved as *sinusoidal* templates [5]. Brief noisy events are captured as *transient* templates, using onset detection methods [1]. The *background* din is extracted by removing sinusoidal components in the frequency domain [5] and replacing transient segments in the time domain. Each template can then be transformed independently, including structural changes to sinusoidal templates [3]. Time and frequency transformations are available for sinusoidal and transient templates, while background din may be continuously generated via wavelet tree learning [2]. In the synthesis phase, these transformed templates can be combined using special *synthesis* templates to produce a new composite soundscape or composition. A set of graphical user interfaces offers interactive parametric control over each phase of analysis, transformation and re-synthesis. ChucK scripting provides additional control over the synthesis through the simultaneous, precise manipulation of many parameters, and also provides an interface for input from external devices or user-defined GUI elements.

The TAPESTREA software is open-source, cross-platform, and freely available at `http://taps.cs.princeton.edu/`. A package containing the source code, license, documentation, and multimedia examples and demo can be downloaded at `http://taps.cs.princeton.edu/tapestrea-acmm09.tgz`. User support includes online documentation, a wiki, and a mailing list with 178 members. Web logs indicate that the software has been downloaded approximately 10300 times

MM'09, October 19–24, 2009, Beijing, China.
ACM 978-1-60558-608-3/09/10.

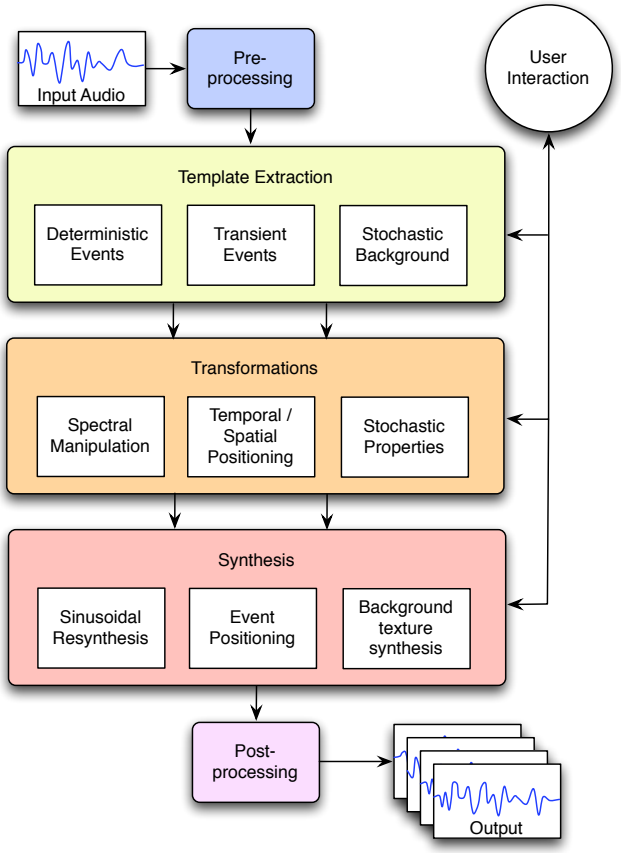

Figure 1: TAPESTREA sound design pipeline.

since April 2008; this statistic counts downloads of only the latest released version at any time, but does not account for bots or repeated downloads by the same person.

3. APPLICATIONS

While TAPESTREA intends to aid sound design and composition from existing sounds, its applications include:

- Analysis, information extraction: Parametrically extracting a part of a recording, possibly gaining a better understanding of the sound through the analysis techniques and parameters that best capture it,
- Synthesis, sound design, composition: Creating sound scenes for entertainment (such as games or video), and musical compositions (such as musique concrète),
- Pedagogy: Enhancing understanding of digital audio, signal processing, and computer music concepts via the interactive analysis, synthesis, and audiovisual display.

An informal survey of users on the TAPESTREA mailing list collected usage information. Voluntary responses to a series of yes/no questions are summarized in Figure 2. These provide an idea of how TAPESTREA has actually been used. All respondents reported having used it to "play around and explore," while the majority of users also reported having used it to extract sound parts and to transform and synthesize templates. Half the users have used it to "create a piece that [they] shared with others." A few have also used it for more specific arenas such as enhancing

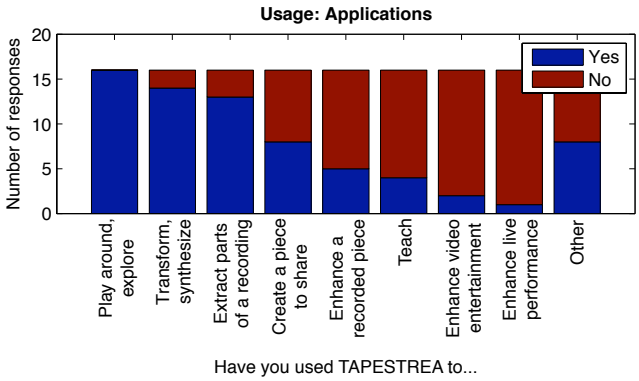

Figure 2: Applications for TAPESTREA usage.

a recorded piece, teaching, enhancing video entertainment (like games or animations) and live performance.

4. CONCLUSIONS AND FUTURE WORK

TAPESTREA provides a new way to design sound through highly flexible manipulation of existing recordings. The tools and interface together provide great freedom in selecting which parts of a sound to re-use and how to transform and combine selected elements. The interactive coupling of analysis and synthesis techniques and paradigms make TAPESTREA stronger than the sum of its parts, leading to a comprehensive framework for manipulating existing audio. Thus, TAPESTREA presents a powerful tool for sound design, composition and pedagogy. Areas for further work include continuing to integrate user feedback to improve the software, as well as expanding it to provide even more powerful scripting, intelligent parameter suggestions via machine learning, and a meaningfully searchable database of extracted templates.

5. REFERENCES

[1] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5), 2005.

[2] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Synthesizing sound textures through wavelet tree learning. *IEEE Computer Graphics and Applications*, 22(4), 2002.

[3] T. Lieber, A. Misra, and P. R. Cook. Freedom in TAPESTREA! Voice-aware track manipulations. In *Proceedings of the International Computer Music Conference*, 2008.

[4] A. Misra, P. R. Cook, and G. Wang. Musical tapestries: Re-composing natural sounds. In *Proceedings of the International Computer Music Conference*, 2006.

[5] X. Serra. *A System for Sound Analysis / Transformation / Synthesis based on a Deterministic plus Stochastic Decomposition*. PhD thesis, Stanford University, 1989.

[6] G. Wang and P. R. Cook. ChucK: A concurrent, on-the-fly, audio programming language. In *Proceedings of the International Computer Music Conference*, 2003.