

UnderScore: Musical Underlays for Audio Stories

Steve Rubin*, Floraine Berthouzoz*, Gautham J. Mysore†, Wilmot Li†, Maneesh Agrawala*

*University of California, Berkeley

{srubin, floraine, maneesh}@cs.berkeley.edu

†Advanced Technology Labs, Adobe

{gmysore, wilmotli}@adobe.com

ABSTRACT

Audio producers often use *musical underlays* to emphasize key moments in spoken content and give listeners time to reflect on what was said. Yet, creating such underlays is time-consuming as producers must carefully (1) mark an emphasis point in the speech (2) select music with the appropriate style, (3) align the music with the emphasis point, and (4) adjust dynamics to produce a harmonious composition. We present UnderScore, a set of semi-automated tools designed to facilitate the creation of such underlays. The producer simply marks an emphasis point in the speech and selects a music track. UnderScore automatically refines, aligns and adjusts the speech and music to generate a high-quality underlay. UnderScore allows producers to focus on the high-level design of the underlay; they can quickly try out a variety of music and test different points of emphasis in the story. Amateur producers, who may lack the time or skills necessary to author underlays, can quickly add music to their stories. An informal evaluation of UnderScore suggests that it can produce high-quality underlays for a variety of examples while significantly reducing the time and effort required of radio producers.

Author Keywords

Radio; music; audio editing; storytelling.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces - Graphical user interfaces (GUI)

INTRODUCTION

Radio shows, podcasts and audiobooks often use music to emphasize key moments in the spoken content. One common technique is to create a *musical underlay*, which fades in music before the *emphasis point* in the speech, then pauses the speech while the *music solo* plays at full volume, and finally fades out the music as the speech resumes (Figure 1). Professional radio producers use such underlays to enhance the mood of the story and to give listeners time to reflect on the key moment in the speech [3, 4].

Creating an underlay involves four main steps: (1) marking the emphasis point in the speech, (2) selecting music with the appropriate style, mood, tempo, etc., (3) aligning the music

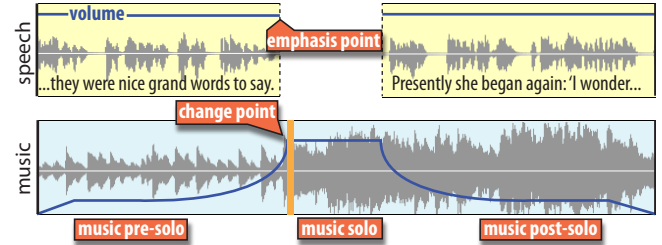


Figure 1. A musical underlay highlights an *emphasis point* in an audio story. The music track contains three segments; (1) a *music pre-solo* that fades in before the *emphasis point*, (2) a *music solo* that starts at the *emphasis point* and plays at full volume while the speech is paused, and (3) a *music post-solo* that fades down as the speech resumes. At the beginning of the solo, the music often changes in some significant way (e.g. a melody enters, the tempo quickens, etc.) Aligning this *change point* in the music with a pause in speech and a rapid increase in the music volume further draws attention to the *emphasis point* in the story.

solo with the *emphasis point*, and (4) adjusting the dynamics to achieve a harmonious composition of music and speech (Figure 1). Each of these steps can be time-consuming as producers must carefully refine the timing, alignment and dynamics to generate a high-quality underlay. Moreover, producers often try out several underlay compositions using different music before settling on the best one.

We have studied a variety of radio programs [2, 5, 23] and publications on radio production [1, 3, 4, 9, 22] to identify the properties of high-quality underlays. We find that the most effective underlays often introduce a significant change in the music (e.g. in melody, tempo, etc.) at the *emphasis point* in the speech to further draw attention to that point and underline the emotional tone of the story. This *change point* in the music marks the beginning of the solo. In addition, audio producers carefully adjust the dynamics (volume) of the music to further emphasize the *change point* while ensuring that it does not interfere with the speech.

In this paper we present UnderScore, a semi-automated system for adding musical underlays to audio stories. To create an underlay the producer simply marks an *emphasis point* in the speech and selects a music track. UnderScore then applies a sequence of tools that automatically refine the *emphasis point*, select a suitable *change point* in the music, align the music with the speech and adjust the dynamics of the composition. The resulting underlay appears in a timeline-based audio editing interface that allows the producer to further tweak any aspect of the underlay as necessary.

UnderScore allows producers to focus on the high-level design of the underlay; they can quickly try out a variety of music and test what it sounds like to emphasize different points in the story. Our automated tools handle the time-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'12, October 7–10, 2012, Cambridge, Massachusetts, USA.

Copyright 2012 ACM 978-1-4503-1580-7/12/10...\$15.00.

consuming, low-level details of aligning the tracks and adjusting their dynamics. Such automation allows amateur producers, who may not have the time or skills to author underlays from scratch, to quickly add music to their stories.

However, our automated results cannot always match the audio quality of underlays created by expert producers. Thus, UnderScore also offers a user-in-the-loop mode in which producers can manually apply each of our underlay tools and immediately decide if the results are acceptable. The producer can always tweak or undo the results of our tools. This mode allows advanced producers to exercise finer-grained control over the design of the underlay.

We demonstrate the effectiveness of UnderScore with an informal evaluation in which three independent experts rated the timing, dynamics and overall quality of our automatically generated underlays. For both timing and dynamics, the vast majority of ratings (89% or more) indicated little or no perceived problems in the underlay, and for overall quality, 86% of the ratings indicated satisfaction with the results (3 or higher on a 5 point Likert scale). We also interviewed both amateur and expert producers who indicated that UnderScore addresses both challenging and time-consuming parts of the editing process. These results suggest that our tools can produce high-quality underlays for a variety of examples while significantly reducing the time and effort required of audio producers.

RELATED WORK

Audio researchers have developed algorithms and interaction techniques to help users edit and compose audio files. Although none of these methods are designed to facilitate the production of audio stories, we consider the set of interfaces and techniques most relevant to our work.

Researchers have developed a variety of low-level techniques to help professional producers edit speech and music recordings [26]. Many of these methods, including automatic dynamics adjustment [11] and noise reduction [12], were introduced several decades ago and have since become standard tools in commercial audio editing software such as ProTools and Adobe Audition. While these tools can help improve the quality of the speech and music tracks in an audio story, they do not directly address the higher level challenge of how to compose these tracks into musical underlays. Thus, we view this existing research as complementary to our work.

Other audio researchers have focused on methods to segment audio into semantically meaningful constructs such as verse/chorus or speech/music [7, 21, 24, 27, 30]. These techniques typically model the global characteristics of each segment and then use the model to detect the transition points between segments. Our system also identifies change points in the song that can then be aligned with the emphasis point in the spoken content. However, we target more localized changes in features such as volume or timbre. Thus, our approach does not require sophisticated models and can instead use a local windowing approach.

The most relevant previous work investigates user interfaces and interaction techniques that simplify various audio editing

workflows. Much of this research focuses on the task of creating music and proposes graphical representations [19, 36] and tangible multitouch interfaces [8, 10, 13, 31] that support music composition and live performance. Other researchers have developed tools that leverage metadata and structure (either from automatic audio analysis or manual tagging) to enable users to edit speech and music at a more semantic level (e.g., editing speech via transcripts [15], mixing and matching multiple takes from a recording session [16, 20]) rather than directly working with waveforms or spectrograms. There has also been previous work that supports vocalized user input for selecting specific sounds in audio mixtures [34] and for creating musical accompaniments to vocal melodies [33]. Most of this research adopts the general strategy of identifying the requirements and constraints of specific audio editing tasks in order to design user interfaces that expose only the most relevant parameters or interaction methods to the user. We follow a similar strategy for the task of creating musical underlays.

DESIGN GUIDELINES FOR MUSICAL UNDERLAYS

In developing UnderScore, we follow the approach of Agrawala et al. [6] and first identify a set of design guidelines for creating high-quality musical underlays. We draw on publications describing the best practices of radio production [1, 3, 4, 9, 22] and analyze high-quality radio programs [2, 5, 23] in order to extract guidelines for each step of the underlay creation process.

Marking speech emphasis points. Audio stories usually include a few important segments that present the central ideas, introduce new characters, or set the mood. Producers often emphasize the endpoints of these segments with an underlay. A short break in the speech allows listeners to process the content of the story and can separate long passages into shorter chunks that are easier to understand [3].

Selecting music. Music serves several functions in an underlay. It augments the emotional content of the story and often builds tension leading up to the emphasis point in the story [4]. The music usually exhibits a significant change at the emphasis point to further draw the listener's attention. This change point can correspond to a melody entering, the tempo quickening, new instrumentation beginning, or a strong downbeat [3].

Aligning music and speech. Aligning music to speech requires choosing the entry and exit points of the music, positioning the change point in the music with respect to the emphasis point in the speech and determining the length of the music solo. The entry point of the music pre-solo often corresponds with a rising action or change of tone in the speech. Abel and Glass [3] suggest that the pre-solo should start 12 seconds before the emphasis point in the speech. The change point in the music usually appears slightly after the emphasis point in the speech. This gap ensures that the change in music does not interfere with the speech and can add to the dramatic effect of the change in music. Finally, Abel and Glass suggest that the music solo should last about 6 seconds before the speech resumes. The length of the music post-solo is less consistent. In some cases the end of the music signals another emphasis point in the speech.

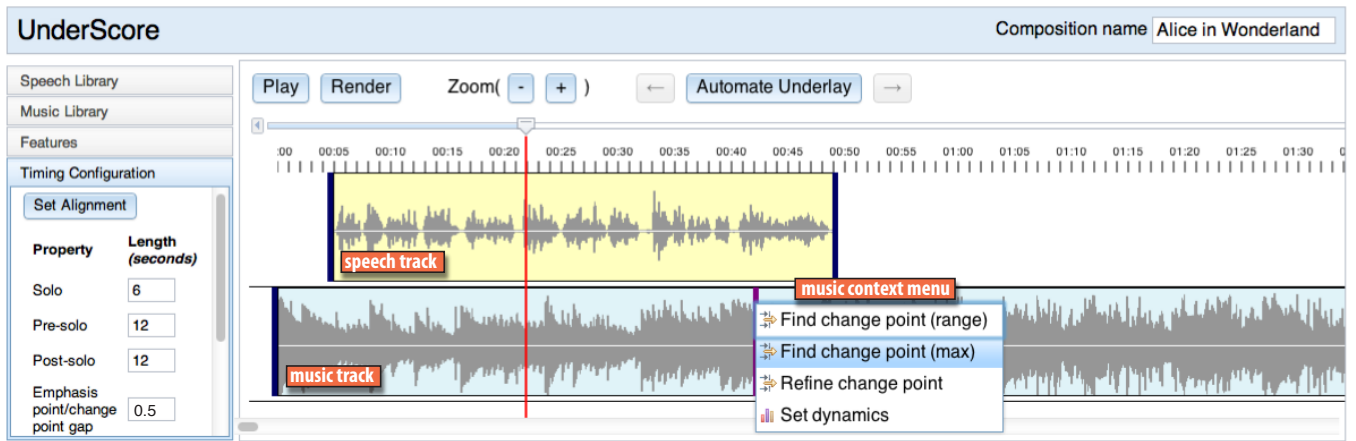


Figure 2. UnderScore is a semi-automated system for adding musical underlays to audio stories. It includes a multitrack timeline-based editor with context menus that provide tools to automate different steps of the underlay creation process. The timing configuration menu allows users to set lengths of the music pre-solo, solo, and post-solo segments to be used in automatic alignment.

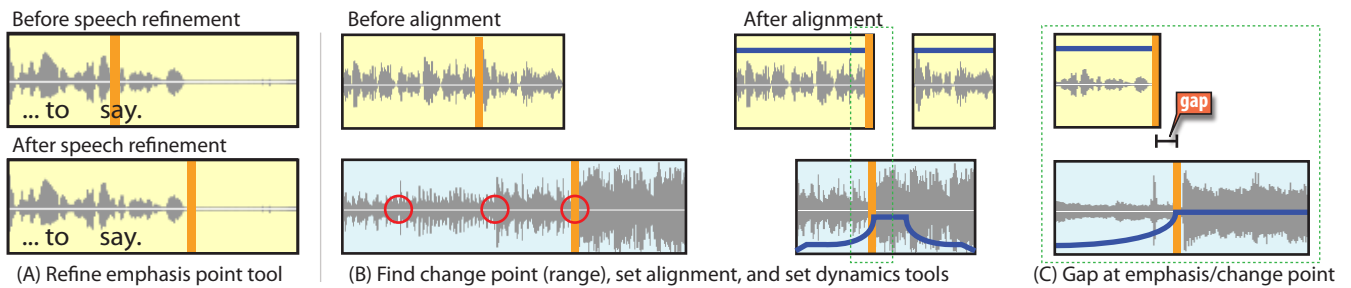


Figure 3. Tools for automating underlay creation. The refine emphasis point tool (A) adjusts a coarsely marked emphasis point in the speech to avoid speech clipping and excessive trailing silence. The find change point (range) tool (B, before) indicates the strongest change points in the music with red circles. The set alignment tool aligns the change point in the music with the emphasis point in the speech and sets the lengths of the audio segments. The set dynamics tool then adjusts the dynamics to ensure that both the music and the speech remain audible. The set alignment tool ensures that there is a small gap between the emphasis point in the speech and the change point in the music, while the set dynamics tool rapidly increases the volume of the music in this gap.

Adjusting dynamics. The goal of adjusting the underlay dynamics is to further heighten the impact of the change point in the music while ensuring that the speech is clearly audible. During the pre-solo the music plays softly, gradually increasing in volume. Near the emphasis point the volume quickly increases and reaches its maximum at the change point in the music to create a strong auditory transition. Finally, just before the speech resumes the music quickly fades down for the post-solo.

AUDIO EDITING INTERFACE

Our audio editing interface (Figure 2) offers a set of tools designed to streamline the process of creating musical underlays and transitions. We first describe these tools in the context of a typical editing session for an audio story (Figure 3). We then present the algorithms underlying these tools in the section on Algorithmic Methods.

Step 1: Marking Speech Emphasis Points

The first step in creating an underlay is to select a speech track and mark an emphasis point in the speech. In our example, the producer selects a reading from Alice in Wonderland [14] from the speech library and inserts a mark after the phrase “but thought that they were nice grand words to say”

(Figure 3A). Although audio tracks are usually sampled at 44.1 kHz, producers typically work with tracks at the granularity of seconds. At this coarse granularity the mark is often slightly too early and the speech is clipped or slightly too late, which adds a trailing silence to the speech. To mark the speech accurately, the producer can manually zoom in on the waveform to work at finer granularity and then carefully find the end of the sentence.

UnderScore provides a *refine emphasis point tool* that automates this process. The producer accesses this tool through the right-click context menu of the speech track and it automatically shifts the producer’s coarse granularity mark to a nearby point that does not clip the speech and minimizes trailing silence (Figure 3A). This tool allows producers to roughly mark the emphasis point without worrying about its sub-second accuracy.

Step 2: Selecting Music

The producer next selects a track from the music library that is stylistically appropriate for the speech and contains a significant change point. Here he picks Damiak’s “Tenuous Gears.” If the producer knows the music well, he can manually mark the change point and then use UnderScore’s *re-*

fine change point tool to automatically shift the mark to the strongest change point within a small range.

If the producer is less familiar with the music, he can apply UnderScore’s *find change point (range) tool* which scores the strength of the change at each point in the music and indicates the strongest change points with red circles (Figure 3B). The producer can then listen to each of these potential change points and select the best one, rather than laboriously searching for the best change point throughout the entire song. UnderScore also includes the fully automated *find change point (max) tool* that directly selects the highest scoring change point. This tool further reduces the time required to select the change point. However our fully automated tool may not always find the perfect change point, and in these cases the producer can always use the range tool or manual approach. All of these tools appear in the right-click context menu of the music track.

Step 3: Aligning Music and Speech

The third step is to align the music with the speech and set the lengths of the audio segments. Like many timeline-based interfaces UnderScore allows producers to manually translate audio segments, set the endpoints of a segment and split segments, by clicking and dragging. In manual mode the producer first translates the music so that the change point appears just after the emphasis point in the speech. The small gap between the emphasis point and the change point reduces interference between the music and speech (Figure 3C). UnderScore facilitates this manual alignment process by snapping the change point in place when it is close to the emphasis point. The producer then chooses the lengths of the music pre-solo, solo and post-solo by dragging the endpoints of the music. Finally the producer splits and separates the speech at the emphasis point to create the pause that is filled by the music solo.

UnderScore also provides a *set alignment tool* that automatically performs each of these adjustments. This tool uses default lengths of 6 seconds for the music solo, 12 seconds for both the pre- and post-solo, and 500 ms for the gap between the emphasis and change points. The producer can adjust these default values in the *timing configuration menu*. The resulting underlay segments also appear in the timeline and the producer can further tweak them if necessary.

Step 4: Adjusting Dynamics

To adjust the dynamics the producer can manually set the volume at keyframes within an audio segment and interpolate between them using either a linear or exponential volume transition. In our example (Figure 3B), the producer uses a short linear fade-in at the start of the pre-solo, and then applies an exponential transition so that the music gradually increases in volume at first and then rapidly reaches its maximum at the music change point. Figure 3C shows that the rapid increase in volume falls primarily in the gap between the emphasis point in speech and the change point in music. The producer similarly creates a quick fade-out just before the speech resumes.

UnderScore provides two automated tools that facilitate setting the dynamics. The *set dynamics tool* automatically sets the key frames and volume transitions to create the exponential fades in our example. In some cases large differences in volume levels between the music and speech make it difficult to build a transition in which both tracks remain audible. The *refine dynamics tool* automatically adjusts the overall energy (perceived loudness) of the music and speech to ensure that they fall within a similar range. This tool often improves the clarity of the music when it softly enters during the pre-solo. Both of these tools are available in the right-click context menus of the speech and music tracks.

Automation and Iteration

In each step of the underlay creation process, UnderScore gives producers the option of doing the task manually or leveraging tools that automate the task. This flexibility allows our system to support two main usage modes. In the user-in-the-loop mode, the producer can apply a tool and then manually tweak the results using UnderScore’s timeline interface. This mode allows producers to seamlessly jump back-and-forth between using our automated tools and manually exercising finer-grained control. UnderScore also provides a batch mode in which the producer simply marks the emphasis point in the speech and selects one or more music tracks. UnderScore then automatically applies our tools in sequence (refine emphasis point, find change point, set alignment and timing, and set dynamics) to generate an underlay for each selected music track.

Based on informal user feedback (see Results section), we have found that the fully automated batch mode results are often of sufficiently high quality to be used directly (i.e., without any further refinement) by amateur producers. Thus, UnderScore’s batch mode gives amateurs a quick and easy way to create underlays without requiring much expertise. Expert producers can also use the batch mode to rapidly iterate through underlays spanning several songs and multiple change points within those songs. These experts can either use our automated results as-is, or treat them as a starting point for further adjustments using our user-in-the-loop mode. In this way, UnderScore provides workflows for both amateurs and experts to create high quality underlays.

ALGORITHMIC METHODS

UnderScore relies on a variety of audio analysis algorithms to facilitate underlay creation. Audio analysis is a well studied area [29] and the algorithms typically operate on features that capture specific properties of music and speech such as volume, harmonic content, and timbre. The features are usually computed on standard 44.1 kHz pulse code modulated (PCM) wave files with their left and right channels summed together. Our tools use the following features:

RMS energy. We measure the perceived loudness of speech and music using root mean squared (RMS) energy, which is defined as $\sqrt{(\sum_n x_i^2)/n}$ where x_i is the value of the i th audio sample in a window of size n .

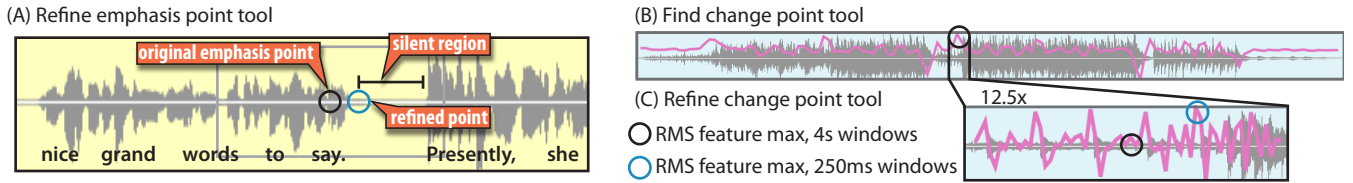


Figure 4. (A) To refine an emphasis point in speech, our system finds the longest region of silence in a 1 second window centered at the marked emphasis point. It then refines the point to the beginning of that silent region. To generate rough music change point estimations, our system finds the maxima of the RMS energy feature using 4 second subwindows that overlap by 50% and span the entire music track (B). To refine a change point, our system applies the same approach but with 250 ms subwindows that span an 8 second window about the coarse point (C). This two step process first large scale changes in volume and then find the strongest downbeat at the local scale.

Chroma. We measure the harmonic content of music using chroma vectors [35], which are the projection of the time-frequency spectrum of a song onto the twelve semitones of the chromatic scale ($C, C^\sharp, B^\flat, \dots$). We use the implementation of Ellis [18].

MFCCs. We measure the timbre of music (which often correlates with instrumentation) by computing its mel frequency cepstral coefficients (MFCCs) as described by Logan et al. [25].

While many of our tools depend on RMS energy, we have found that chroma and MFCCs are useful for finding change points.

Refine Emphasis Point Tool

The refine emphasis point tool is designed to move a coarsely marked emphasis point in the speech to the start of a nearby silent region (Figure 4A). To find this region, we consider a 1 second window centered at the coarsely marked emphasis point. We divide this window into 62.5 ms subwindows with 50% overlap and then compute the RMS energy of each one. We then find the longest span of subwindows whose energies all fall within 10% of the minimum energy of any subwindow.

Refine Change Point Tool

The refine change point tool adjusts a coarsely-marked change point in the music to the strongest change within an 8 second window (Figure 4C). To identify this change, we first compute all three features on 250 ms subwindows with 50% overlap. We then compute the distance between adjacent subwindows using signed difference for RMS energy, and Euclidean distance for both chroma and MFCCs. For each feature, we identify the pair of windows yielding the maximum distance, and set the refined change point to the beginning of the second window in this pair.

Find Change Point Tool

The find change point tool coarsely identifies change points in music. This tool uses exactly the same algorithm as the refine change point tool but with window size set to the length of the entire song and 4 second subwindows (Figure 4B). We then further adjust the coarse change points by applying the refine change point tool.

The find change point (range) tool returns a set of refined change points corresponding to the n strongest coarse changes in the music ($n = 3$ by default). If any of the n

refined change points lie within 250 ms of each other, we replace the weaker point with the next strongest change point. The find change point (max) tool returns the refined point with the strongest coarse change in the music. We have found that RMS energy often works well for finding and refining change points because it identifies large changes in volume at the coarse level and strong downbeats at the fine level. We use it as our default feature, but we have found that chroma and MFCCs are useful for music whose variation is primarily in harmonics or timbre rather than in volume.

Set Alignment Tool

The set alignment tool moves the change point in music to just after the emphasis point in speech. It then adjusts the length of the pre-solo, solo, and post-solo. Finally, it splits and translates the speech track to create the pause. The lengths of the resulting segments as well as the gap are specified by the timing configuration.

Set Dynamics Tool

The set dynamics tool adjusts the volume of the music pre-solo, solo, and post-solo. The pre-solo dynamics are a short linear fade-in followed by an exponential increase in volume. The solo plays at full volume, and the post-solo volume adjustment mirrors that of the pre-solo.

Refine Dynamics Tool

The refine dynamics tool further adjusts the volume of the music with respect to the speech to ensure that both tracks are clearly audible. More specifically, we scale the music by $k = \sqrt{RMS(speech)/RMS(music)}$ so that the music and speech are within a similar range of volume. In this case, we compute the RMS energies using only the audio segments within the underlay.

RESULTS

We have created a number of underlays with the user-in-the-loop mode of UnderScore for 3 different audiobook speech clips (“Alice in Wonderland” [14] by Lewis Carroll, “Great Expectations” [17] by Charles Dickens and “Me Talk Pretty One Day” [32] by David Sedaris) representing both male and female voices and a range of emotional tones. We selected the music from tracks commonly heard on the radio documentary program “This American Life” [23]. The resulting underlays are included in our project website¹. To demonstrate the importance of our automated tools, the website also

¹<http://vis.berkeley.edu/papers/underscore/>

includes examples of results generated with and without using our set alignment and dynamics refinement tools. While all of our tools reduced time and effort, the find change point (range) tool significantly reduced the amount of time we spent searching through the music. The set alignment and dynamics tools eliminated much of the tweaking involved in manual creation of underlays.

To evaluate the automated batch mode of UnderScore, we used a speech clip from David Sedaris’s “Me Talk Pretty One Day” audiobook [32] and selected 26 songs that are commonly used on the radio documentary program “This American Life.” [23]. For each combination we used the find change point (range) tool with the RMS energy feature and automatically generated underlays corresponding to the 3 strongest change points in the music. We asked 3 independent experts who are familiar with “This American Life” to rate the timing, dynamics, and overall quality of each underlay on a 5-point Likert scale (Figure 5).

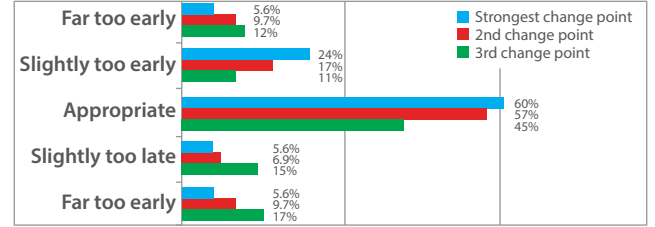
The underlays generated with the one strongest change point for each song are equivalent to the results of our find change point (max) tool. The ratings for these underlays correspond to the blue bars in Figure 5. With respect to the timing of the change point, the experts gave a rating of 3 (appropriate timing) in 60% of the responses, 89% were rated between 2 (change point arrived slightly too early) and 4 (slightly too late), and 11% were rated either 1 (far too early) or 5 (far too late). For the dynamics of the music, 59% of the ratings were 3 (appropriate volume), 94% were between 2 (slightly too soft) and 4 (slightly too loud), and 6% were rated 1 (far too soft) or 5 (far too loud). To measure overall quality, we asked the experts to evaluate the statement “I am satisfied with the overall quality of the underlay” on a scale from 1 (strongly disagree) to 5 (strongly agree); 50% of the responses were 4 (agree) or higher, 86% were 3 (neutral) or higher, and 14% were 2 (disagree) or lower. On the whole, these ratings suggest that our find change point (max) tool produces high quality underlays with good timing and dynamics.

The ratings for underlays using the second and third strongest musical change points for each song (red and green bars in Figure 5, respectively) are similar to the ratings for underlays using the strongest change point. However, the scores for timing and overall quality do decline somewhat as we move from the first to the third strongest change points. This trend suggests that our algorithm for finding a range of change points produces a useful ordering. For dynamics, there is no clear decline in ratings across the three change points. Note that our set and refine dynamics tools are independent of the strength of the change points. Thus, these tools can produce underlays with the appropriate relative dynamics between music and speech regardless of the strength of the change point.

We have found songs for which the chroma and MFCCs features find effective change points even when the RMS energy feature does not (Figure 6). Although we have found that the RMS energy feature is usually more effective than MFCCs and chroma, this example suggests that producers should consider the other features when RMS energy fails.

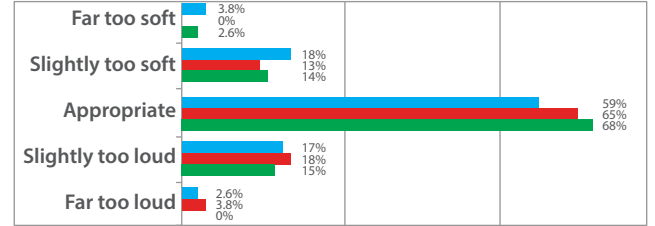
Timing

“Rate the timing of the musical change point in the underlay.”



Dynamics

“Rate the dynamics of the music in the underlay.”



Overall Quality

“I am satisfied with the overall quality of the underlay.”

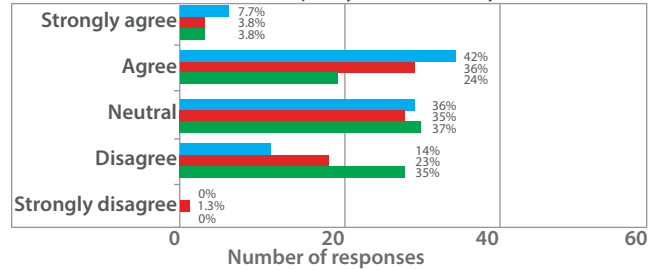


Figure 5. Three experts judged the quality of underlays automatically generated with the 3 strongest change points for each of 26 songs. The percentages are the ratio of a given rating to all ratings for each change point strength. While the strongest change point results in the best ratings, the second and third change points also perform well.

To get a sense for the usefulness of our system, we recruited 6 amateur users, who were either interested in creating audio stories or were casual podcasters, and 1 expert producer. All users were familiar with the technique of adding musical underlays to audio stories. We interviewed the participants about the process of creating musical underlays and showed UnderScore to them. All agreed that underlays enhance audio stories but are too much work to create manually with existing tools. They cited finding musical change points as the most time-consuming step, and most felt refining timing and dynamics would also be difficult. This informal feedback suggests that UnderScore’s automated tools address some of the main problems that users face when creating underlays. In addition, all amateurs said they would use our batch mode results without further refinement; the expert said our fully automated results were very good but liked having the option to refine them via our user-in-the-loop mode.

FUTURE WORK

While UnderScore’s fully automated tools usually produce good results, at times the change point comes too early or too late, and in rare cases, the music pre-solo can be too soft or too loud even after dynamics refinement. Thus, one direction

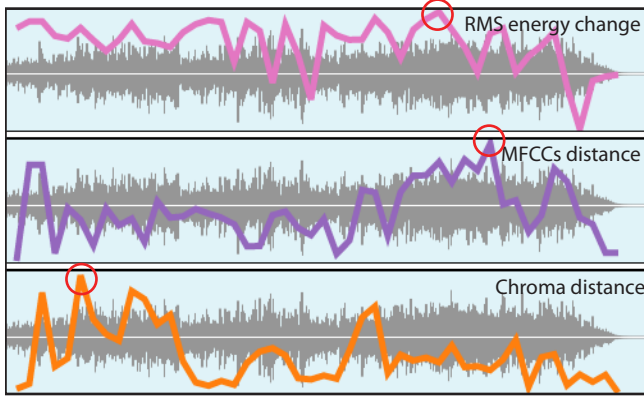


Figure 6. In “Scrapping and Yelling” by Mark Mothersbaugh [28], the maximum of the RMS energy change feature does not correspond to a strongly perceptible change in music. However, the maxima of both MFCCs and chroma distance features give a strong sense of change and lead to an effective underlay (see website).

for future work is to improve the audio analysis algorithms behind our automated system.

We see several other promising directions for further research:

Matching music to stories. Finding the appropriate music for an underlay can be a challenging task that involves searching a large library of songs to find one that best matches the semantic content of the story. Future work could explore and evaluate automated music suggestion techniques that analyze the sentiment of the speech and mood/emotion of the music.

Extended underlays. In some cases, producers may want to emphasize multiple speech points (or an entire story) with a continuous piece of music. Composing such extended underlays automatically would require algorithms that align multiple change points to multiple emphasis points, possibly by extending or contracting musical segments to fit the speech.

Underlays for live performance. In addition to enhancing recorded stories, underlays can emphasize key moments in live performances as well. Future work could investigate algorithms and interfaces that directly support this application.

Music for videos. Enhancing videos with music involves some similar challenges as creating effective underlays for audio stories (e.g., aligning musical changes to the video, balancing the dynamics, etc.). Designing tools that address these challenges for video production is another direction for future research.

Other production techniques. In general, underlays are just one of many production techniques that enhance audio stories. For example, producers sometimes incorporate ambient noise and sound effects to drive narrative development (e.g., in Radiolab [5]) or use music to distinguish multiple on-air performers based on their personalities and musical preferences. We believe that developing and evaluating interactive tools for applying such techniques represents an exciting opportunity for the HCI community.

CONCLUSION

Musical underlays can significantly enhance audio stories. We have identified a set of guidelines for producing high quality underlays. We have also instantiated these guidelines in a set of tools designed to automate the underlay creation process. This automation can help amateur producers who lack the necessary time and skills to add underlays to their stories. It also makes it easier for expert producers to quickly try out a variety of music and test different emphasis points in the speech. Our evaluations suggest that our automated tools create successful underlays on a variety of examples.

ACKNOWLEDGMENTS

This work was supported by NSF grant CCF-0643552.

REFERENCES

1. Stupid fade tricks. <http://transom.org/?p=7543>, September 2003.
2. Third coast audio library. <http://www.thirdcoastfestival.org/library>, April 2012.
3. Abel, J., and Glass, I. *Radio: An Illustrated Guide*. WBEZ Alliance Inc., 1999.
4. Abumrad, J. Music: A force for good (and sometimes evil). <http://www.thirdcoastfestival.org/library/450-music-a-force-for-good-and-sometimes-evil>, 2005.
5. Abumrad, J., and Krulwich, R. Radiolab. <http://www.radiolab.org/>, April 2012.
6. Agrawala, M., Li, W., and Berthouzoz, F. Design principles for visual communication. *Commun. ACM* 54, 4 (Apr. 2011), 6069.
7. Barthet, M., Hargreaves, S., and Sandler, M. Speech/music discrimination in audio podcast using structural segmentation and timbre recognition. In *Proc. of the 7th international conference on Exploring music contents*, Springer-Verlag (2011), 138–162.
8. Berry, R., Makino, M., Hikawa, N., and Suzuki, M. The augmented composer project: The music table. In *Proc. of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society (2003), 338.
9. Biewen, J. The transom review: John Biewen. <http://transom.org/?p=19659>, September 2011.
10. Bischof, M., Conradi, B., Lachenmaier, P., Linde, K., Meier, M., Pötzl, P., and André, E. Xenakis: combining tangible interaction with probability-based musical composition. In *Proc. of the 2nd international conference on Tangible and embedded interaction*, ACM (2008), 121–124.
11. Blesser, B. Audio dynamic range compression for minimum perceived distortion. *IEEE Transactions on Audio and Electroacoustics* 17, 1 (1969), 22–32.

12. Boll, S. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing* 28, 2 (1979), 113–120.
13. Carrascal, J., and Jordà, S. Multitouch interface for audio mixing. *Proc. of New Interfaces for Musical Expression* (2011).
14. Carroll, L. Alice’s Adventures in Wonderland, read by Kristen McQuillin. <http://librivox.org/alices-adventures-in-wonderland-by-lewis-carroll/>, Apr. 2012.
15. Casares, J., Long, A. C., Myers, B., Stevens, S., and Corbett, A. Simplifying video editing with silver. In *CHI ’02 extended abstracts on Human factors in computing systems*, ACM (2002), 672–673.
16. Dannenberg, R. An intelligent multi-track audio editor. In *Proc. of International Computer Music Conference (ICMC)* (2007), 89–94.
17. Dickens, C. Great Expectations, read by Mark F. Smith. <http://librivox.org/great-expectations-by-charles-dickens/>, Apr. 2012.
18. Ellis, D. Chroma feature analysis and synthesis. <http://labrosa.ee.columbia.edu/matlab/chroma-ansyn/>, April 2012.
19. Farbood, M., Pasztor, E., and Jennings, K. Hyperscore: a graphical sketchpad for novice composers. *Computer Graphics and Applications, IEEE* 24, 1 (2004), 50–54.
20. Fazekas, G., and Sandler, M. Intelligent editing of studio recordings with the help of automatic music structure extraction. In *Proc. of the Audio Engineering Society 122nd Convention, Vienna, Austria* (2007).
21. Foote, J. Automatic audio segmentation using a measure of audio novelty. In *International Conference on Multimedia and Expo (ICME)*, IEEE (2000), 452–455.
22. Glass, I. The transom review: Ira Glass. <http://transom.org/?p=6978>, June 2004.
23. Glass, I. This American Life. <http://www.thisamericanlife.org/>, April 2012.
24. Jensen, K. Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Advances in Signal Processing* 2007 (2007).
25. Logan, B. Mel frequency cepstral coefficients for music modeling. In *Proc. of the International Symposium on Music Information Retrieval* (2000).
26. Louizou, P. C. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.
27. Maddage, N. C., Xu, C., Kankanhalli, M. S., and Shao, X. Content-based music structure analysis with applications to music semantics understanding. In *Proc. of the 12th annual ACM international conference on Multimedia, MULTIMEDIA ’04, ACM (New York, NY, USA, 2004)*, 112–119.
28. Mothersbaugh, M. Scrapping & yelling, The Royal Tenenbaums soundtrack, December 2001. Audio recording.
29. Müller, M., Ellis, D., Klapuri, A., and Richard, G. Signal processing for music analysis. *Selected Topics in Signal Processing, IEEE Journal of* (2011).
30. Paulus, J., Muller, M., and Klapuri, A. Audio-based music structure analysis. In *Proc. of the International Symposium on Music Information Retrieval* (2010), 625–636.
31. Roma, G., and Xambó, A. A tabletop waveform editor for live performance. *Proc. of New Interfaces for Music Expression* (2008).
32. Sedaris, D. *Me talk pretty one day*. audiobook. Little, Brown and Company, 2000.
33. Simon, I., Morris, D., and Basu, S. MySong: automatic accompaniment generation for vocal melodies. In *Proc. of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM (2008), 725–734.
34. Smaragdis, P. User guided audio selection from complex sound mixtures. In *Proc. of the 22nd annual ACM symposium on User interface software and technology*, ACM (2009), 89–92.
35. Wakefield, G. Mathematical representation of joint time-chroma distributions. In *International Symposium on Optical Science, Engineering, and Instrumentation, SPIE*, vol. 99 (1999), 18–23.
36. Wright, J., Oppenheim, D., Jameson, D., Pazel, D., and Fuhrer, R. Cyberband: A “hands-on” music composition program. In *Proc. of International Computer Music Conference* (1997), 383–386.