LoopMaker: Automatic Creation of Music Loops from Pre-recorded Music

Zhengshan Shi
CCRMA, Stanford University
Stanford, CA, USA
kittyshi@ccrma.stanford.edu

Gautham J. Mysore
Adobe Research
San Francisco, CA, USA
gmysore@adobe.com

ABSTRACT
Music loops are seamlessly repeatable segments of music that can be used for music composition as well as backing tracks for media such as videos, webpages, and games. They are regularly used by both professional musicians as well as novices with very little experience in audio editing and music composition. The process of creating music loops can be challenging and tedious, particularly for novices. We present LoopMaker, an interactive system that assists users in creating and exploring music loops from pre-recorded music. Our system can be used in a semi-automatic mode in which it refines a user's rough selection of a loop. It can also be used in a fully automatic mode in which it creates a number of loops from a given piece of music and interactively allows the user to explore these loops. Our user study suggests that our system makes the loop creation process significantly faster, easier, and more enjoyable than manual creation for both novices and experts. It also suggests that the quality of these loops are comparable to manually created loops by experts.

ACM Classification Keywords
H.5.5. Sound and music computing: Methodologies and techniques: I.5.5.Implementation:Interactive systems

Author Keywords
Music loop creation; Interactive audio editing; Audio applications.

INTRODUCTION
A music loop is a segment of music that can be played repeatedly such that the transition from the end of the segment back to the beginning of the segment sounds natural and seamless. Continuously playing a loop essentially sounds like a continuous piece of music [18]. Loops are regularly used by both professionals and novices (people with little experience in music or audio editing) for a variety of applications. Musicians use loops to compose themed layers in their musical pieces and DJs often use loops to customize pre-existing music for their specific needs. Editors of media such as video, webpages, and games often use music as a backing track [6]. Such editors are often novices with respect to music and audio editing. One of the reasons that a loop is well suited for background music is that it can be played for an arbitrary length of time and sound continuous. This is particularly useful for webpages and games since the length of time that a user spends on a given section is not pre-determined. It is also helpful for deployment, as a loop can often be as short as a few seconds and requires little storage.

To create music loops, one can either compose them, or create them from pre-recorded music. The focus of this paper is the latter. A typical workflow to create such loops [1, 11, 22] using audio editing software is as follows:

1. Listen to a significant portion of a song or a whole song.
2. Roughly identify a part that would be a candidate for a music loop.
3. Find precise beginning and end points of the loop such that there is no abrupt sounding transition when played repeatedly.
4. Use low level audio tools to crop this selection, creating the loop.

Our user studies suggest that this loop creation process can be challenging and time-consuming, especially for novices. We present LoopMaker, an interactive system to assist users in creating loops from pre-recorded music. The goal is to automate the tedious aspects of this task (which our user study suggests in step 3 and 4 above), allowing users to be involved in the more creative aspects of the task (step 1 and in some instances, step 2). Based on this, LoopMaker has the following two modes of operation:

- *Semi-automatic* — The user listens to a song and finds the rough locations of a desired loop (steps 1 and 2 in the work flow). Our system then performs steps 3 and 4 to refine and create the loop.
- *Automatic* — The user simply provides a song and the desired loop length. Our system automatically performs all four steps and creates multiple loops that fit the length criterion. Our system provides an interface for the user to explore the created loops and select the ones that are most suitable for the desired task.

The processing time in both modes is on the order of seconds. We performed a user study with both novices and experts to

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

DOI: https://doi.org/10.1145/3173574.3174028

Figure 1. Interface for LoopMaker. It can automatically create loops using the *Find me Loops* button. It can assist users in creating loops in semi-automatic mode using the *Refine* button after users roughly select a region using the cursor.

determine the usefulness of our system by comparing loops created using our system to manually created loops. We compared both difficulty in creating loops and quality of the created loops. Our study suggested that using our system was significantly easier and faster than manual creation of loops and the quality of these loops were comparable to those created by experts.

RELATED WORK

Automation of tedious tasks that are inevitably part of the creative process, can allow users to focus their efforts on being creative [9]. Moreover, it can allow novices to be creative in ways that could be prohibitive based on their skills. This has previously been demonstrated on applications such as painting and sketching [2, 25].

Interactive music and audio editing systems have previously been built for various applications such as DJing [13], text-based speech editing [19], and interactive audio source separation [5].

However, to the best of our knowledge, there is little existing work in creating systems to assist in music loop creation from pre-existing music. Ong and Streich present an algorithm [17] to automatically create music loops. A fundamental limitation of their algorithm is that it creates loops only from segments that repeat in a song. Although such segments can lead to compelling loops, only a subset of all natural sounding loops in a song follow this principal. They also present a system to visualize existing loops [21] based on various criteria. Kitahara et. al [15] proposed a loop sequencer that automatically selects music loops from a given set of existing loops based on a temporal evolution of excitement, which is specified by the user. Neither of these systems were created to assist in the loop creation process.

The development of audio software has made it easier for amateurs to produce music [23]. Certain audio editing software packages such as Ableton Live[1] and Recycle[2] provide semantic information and high level representations of music such as tempo and beat locations. Although this can make the process easier for users that are familiar with such representations, it can still be a time-consuming process. Moreover, using these representations requires a certain level of knowledge of music and proficiency in audio editing, which novices typically lack.

[1]https://www.ableton.com/en/

[2]https://www.propellerheads.se/recycle

Figure 2. After clicking the *Find me loops* button, music loops extracted by LoopMaker are shown as orange rectangles with the rows ranked in order of confidence. The degree of confidence of each loop is indicated by the opacity value of the corresponding rectangle.

LOOPMAKER INTERFACE

We present the interface to our system in this section and the underlying algorithms in the following section. Our system is a web application based on the Wavesurfer web audio API [3] with our algorithm running as a back end in Python through the flask framework [4]. First, the user uploads a song from which he or she would like to create loops and is then presented with the interface that is shown in Figure 1. The waveform of the given song is displayed in pink.

Semi-automatic Mode

In this mode, the user selects the approximate region of the desired loop, which is shown in green in Figure 1. The user then simply clicks on the *Refine* button and the system refines the selection boundaries to create a seamless loop. The user can listen to the loop in repeat mode by pressing the *Play* button. If satisfied with the result, the loop can be saved by clicking the *Export Region* button. The user can create multiple loops in this way.

[3]https://wavesurfer-js.org

[4]http://flask.pocoo.org/

Automatic Mode

In this mode, the user clicks on *Find me Loops*, and the system finds and displays the location of a number of loops that occur after the cursor position, as shown in Figure 2. The user can listen to a loop by clicking on it.

Our system sometimes finds multiple variations of the same loop, which we define as loops with similar boundaries (less than one second of each other). These loops are collapsed into a single orange rectangle and can be expanded into the individual loops by clicking on it, as shown in Figure 3.

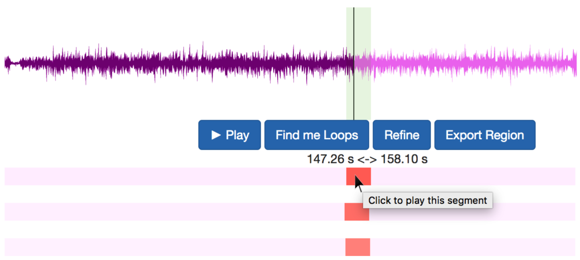

Figure 3. When clicking on a collapsed set of loops, the individual loops unfold and can be individually listened to.

ALGORITHM DESIGN

In this section, we describe our algorithm to automatically find segments of music that could serve as loops. We also describe how we assign a confidence score to each such detected segment. The goal of our algorithm is to automatically and semi-automatically find segments of music that sound natural and seamless when played continuously. This particularly means that the end of a given loop needs to seamlessly transition back to the beginning of the loop.

Consider the illustration of a song in Figure 4. The detected loop is given by A. It consists of parts a and b. Part c immediately proceeds part b. If we assume that the song played unaltered sounds natural and seamless, then the transition from part b to part c will be natural and seamless. Therefore, the transition from part b to another part identical to part c should also sound natural and seamless. Our goal is to transition from part b to a part that sounds as similar to part c as possible, as the above argument would imply that this could create a natural and seamless loop. More generally, for each chunk a in a given song, we detect all similar chunks c. We can create a distinct loop with each match.

We therefore frame the problem of loop detection as automatically detecting similar chunks. We define this similarity in terms of harmony, timbre, and energy [4] and define the units of music as beats [10] as commonly done in the music informational retrieval literature. Such forms of music similarity have previously been used for applications such as music retargeting [19], infinite length music playback [24], and creating mashups [8].

Our algorithm requires us to specify the size of chunks that it compares. The size of a detected loop would by definition be larger than a chunk so the minimum loop size is determined by the chunk size. We therefore favor relatively small chunks. On the other hand, a single beat does not contain enough temporal information for a meaningful comparison. Four beats is the most common length of a bar of music [3]. We therefore chose a chunk size of four beats. Our informal listening showed that it generally worked well even with music whose time signatures contain three beats per bar, but not as well as using chunks of size three.

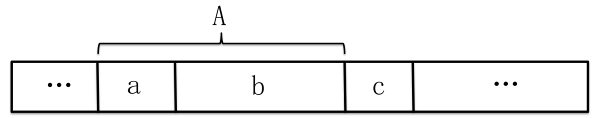

Figure 4. Illustration of music segments. Letters *a*, *b*, and *c* represent different short segments in the song. Segment *a* and segment *c* are in the same length (defined in beats). If segment *a* is additionally musically similar to segment *c*, then *A* could be a seamless sounding loop.

For each beat in a given chunk, we compute features for a window of samples around the beat. Specifically, we compute chroma vectors, Mel-frequency cepstral coefficeints (MFCCs), and root-mean-square energy (RMS) as a representation of harmony, timbre, and energy respectively. We compute the distance between two chunks that start with beat b_i and beat b_j respectively as shown below. The similarity between the two chunks is simply the inverse of this distance.

$$\begin{aligned} L(i,j) = \alpha \sum_{k=0}^{3} D_c(C[i+k], C[j+k]) \\ + \beta \sum_{k=0}^{3} D_m(M[i+k], M[j+k]) \\ + \theta D_r(R_{i:i+4}, R_{j:j+4}) \end{aligned} \quad (1)$$

where D_c denotes the cosine distance between two chroma vectors $C[i]$ and $C[j]$, D_m denotes the Euclidean distance between two MFCC vectors $M[i]$ and $M[j]$, and D_r denotes RMS energy. Our system uses $\alpha = 1$, $\beta = 0.6$, and $\theta = 0.2$ that were obtained through informal listening tests and the authors' musical judgment when creating loops. The distance functions we chose are based on classic music signal processing methods that have been applied on many applications [14, 20].

In the automatic mode of operation, our system compares every chunk of four beats to every other chunk of four beats respecting the user's constraints of search region and loop size. We define matching chunks as those with a distance below an adaptive threshold that filters out the top N loop candidates. For each matching chunk a and c as shown in Figure 4, we create a loop A using chunk a and b with a cross-fade between the end of b and the beginning of a.

In the semi-automatic mode, we simply compare chunks in the region of the user selection. Specifically, we first detect the closest beat location to the start and end points of the user selection. We then compare all chunks that would allow the loop to be within two beats of these detected start and end beats.

EXPERIMENTS AND RESULTS

We performed a user study on both novices and experts to determine if creating loops using our system is easier and faster

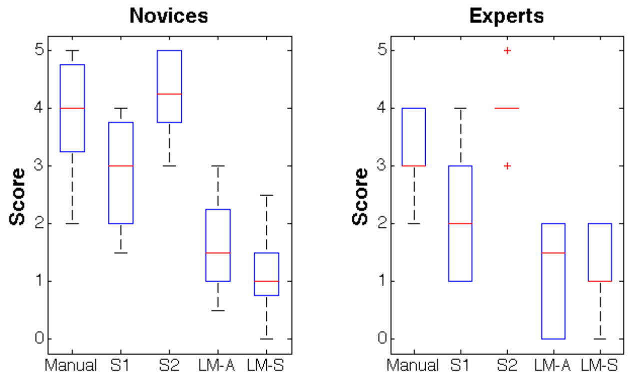

Figure 5. Level of difficulty for loop creation rated by the subjects on a scale of 0 to 5 (5 being extremely difficult). The labels are defined as — S1 and S2 are stage 1 and 2 of manual loop creation, LM-A and LM-S are LoopMaker using automatic and semi-automatic mode.

than manual loop creation. We then performed independent listening tests to compare the quality of the loops created using our system to the manually created loops.

User Study

For this study, we recruited 14 participants from the age of 24 to 53. The participants were a mixture of experts (6) and novices (8). We define an expert as someone who claimed to be both knowledgeable in the use of audio editing software and possess at least a basic music theory background (i.e. play a musical instrument or having taken a music theory class). We define all non-expert participants as novices. We asked each participant to manually create three loops from each of three different songs (total of nine loops). They were allowed to use any audio editing software package they like for this creation. We asked participants with no prior audio editing experience to use Audacity. They were allowed to use any online tutorial of their choice in order to gain a basic understanding of Audacity and audio editing. Two participants were not able to complete the task, as they found the task and use of the software too challenging, but all other participants completed the task.

We asked each participant to rate the difficulty of creating loops on a scale of 0 to 5 (0 referring to extremely easy and 5 referring to extremely difficult). We also asked them to individually rate the two stages of the process. The first stage is to listen to the song and roughly find the location of each loop (steps 1 and 2 mentioned in Section 1). The second stage is to refine the boundaries and create the loop (steps 3 and 4 mentioned in Section 1). The first stage also corresponds to the user input when using the semi-automatic mode of operation.

We then asked each participant to automatically create loops using LoopMaker with both modes and rate the level of difficulty (using the above scale). The music files we used in our experiment were instrumental music files that were randomly selected from the RWC Music Database [12].

We noted the time taken to complete each task, the reported difficulty levels, and comments left by the participants. The length of the manually created loops varies from 3 seconds to 13 seconds with an average of 5.2 seconds. Both novices and experts take about 1 minute per loop to find the approximate location (stage 1). To refine the location of the loops (stage 2), novices took a median of 7.5 minutes and experts took a median of 3 minutes per loop. The time taken for this task using LoopMaker in both automatic and semi-automatic (after completing stage 1 manually) modes is within milliseconds as it happens at the press of a button and has very low computational complexity.

As shown in Figure 5, the median level of difficulty reported by novices was 4 for manual loop creation (3 for stage 1 and 4.25 for stage 2). The median level of difficulty reported by experts was 3 (2 for stage 1 and 4 for stage 2). This is consistent with the amount of time taken for the same things. The difficulty in creating loops using LoopMaker (both automatic and semi-automatic modes) was reported to be significantly lower.

Some novices preferred using our system to find all of the loops automatically, and most of the participants felt that our semi-automatic loop creation is very helpful because it automates stage 2, which both novices and experts found to be difficult. Some experts found it helpful to see the beat location in their favorite music editing software package during stage 2. However, they reported that the task was still difficult.

We found that several experts use the following general procedure to perform the task — they preserve regularity of beats, then detect a transition point by comparing similarity of loudness, consistency in instruments playing and similarity in harmony. This corresponds closely to the way our algorithm functions.

Listening Tests

We compared the quality of the manually created loops to those that were created by LoopMaker, using listening tests. In order to perform an unbiased comparison, we had third parties perform the comparison using Amazon Mechanical Turk (AMT). We define the quality of a loop in terms of how natural and seamless it sounds. This is to say that a loop that sounds like continuous music without unnaturally abrupt transitions when played repeatedly is considered to be a high quality loop. Given the inherently ambiguous nature of this definition, we performed three different kinds of listening tests to judge quality.

The total number of manually created loops was 108 (12 participants that completed the task $\times$ 9 loops/participant). LoopMaker can automatically create a large number of loops per song. However, in order to make the comparison as comparable as possible, we use automatically created loops that were at about the same location as the manually created loops. In order to do this, for each manually created loop, we automatically found the corresponding location in the given song using dynamic time warping [16], and then used the semi-automatic mode of LoopMaker to find the closest possible loop. We therefore have a loop that was created by LoopMaker corresponding to each of the 108 manually created loops.

We recruited 1337 subjects on AMT. We screened each subject with a listening test as was previously done for other AMT based audio evaluation tasks [7]. Specifically, they were asked to listen to three audio segments. Each segment had a different

Type	*Novices*	*Experts*	*Overall*
LoopMaker	**3.864**	3.759	**3.786**
Manual	3.358	**3.788**	3.517

Table 1. Average naturalness rating on a scale of 1(Very Abrupt) to 5(Very Natural).

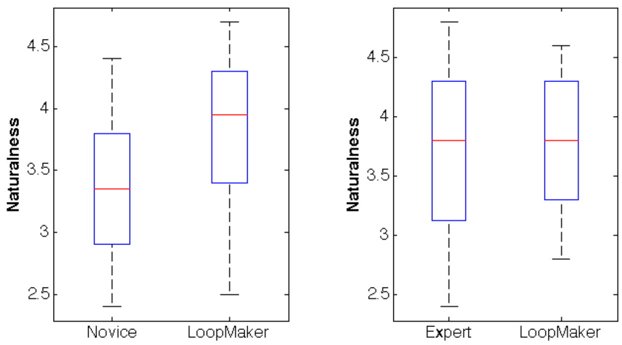

Figure 6. Naturalness rating.

number of sine tones of varying frequencies. They were asked to report the number of sine tones. If the reported number of tones was off by more than one, we discarded the response of the given subject. All but four subjects passed the screening.

We asked each of these subjects to compare 6 manually created loops with the 6 corresponding automatically created loops. They were randomly provided with 6 out of the 108 loops. Below, we describe the three listening tests that were performed by each subject. For each test, we presented the subject with the 12 loops mentioned above. Moreover, we played each loop continuously four times so that they can hear each transition multiple times. This task took each subject a total of about 10 minutes and we paid them for $1.50 for the task.

Naturalness Rating

The goal of this test was to determine how natural the transitions sound to subjects. We asked each subject to rate how natural each loop sounds on a scale of 1 to 5 where 5 means very natural, 4 means moderately natural, 3 means acceptable, 2 means not so pleasant, and 1 means very abrupt. We specifically mentioned that changes in sections or parts of music do not qualify as abrupt changes.

As shown in Table 1, the loops that were created by LoopMaker were rated higher than those created by novices (t-statistic:3.3526, p-value:0.001415) and comparable to those created by experts (t-statistic:2.4579,p-value:0.01525). However, as shown in Figure 6, the variance for those created by LoopMaker is less than the variance of those created by experts.

Abruptness Rating

The goal of this test was to determine if subjects could hear abrupt changes. For each of the 12 loops, we asked them click a button when the hear a change. They were allowed to click the button multiple times per loop. We simply tallied

Type	*Novices*	*Experts*
LoopMaker	**41.19%**	**45.25%**
Manual	57.07 %	57.97%

Table 2. Percentage of abrupt changes detected.

Natural Preference	*Novices*	*Experts*	*Overall*
LoopMaker	**41%**	**39%**	**41%**
Manual	31%	30%	29%
Both Natural	16%	17%	17%
Both Abrupt	12%	14%	13%

Table 3. Preference of naturalness.

the total number of abrupt changes that they could hear and report this as percentage of the total number of transitions. We hypothesize that they would hear fewer abrupt changes in a more natural sounding loop.

As shown in Table 2, the subjects reported less abruptness in loops that were created by LoopMaker when compared to those created by both novices (t-statistic:-2.3843, p-value:0.02042) and experts(t-statistic:-2.1512, p-value:0.03465).

Comparison to Manual Loops

The goal of this test was to directly compare loops that were created by LoopMaker to manually created loops. We presented each subject with each of the 6 pairs of corresponding loops. For each pair, we asked the subject to select one of the following: Clip A sounds more natural, Clip B sounds more natural, Both sound natural, and both sound abrupt.

As shown in Table 3, on average, subjects preferred loops that were created by LoopMaker to those created by both novices and experts.

All three listening tests suggest that loops that were created by LoopMaker are of higher quality than those created by novices and at least comparable to those created by experts. Note that we used LoopMaker to create loops that were comparable in location to those that were manually created. We did not use the loops with the highest confidence score. We argue that a real-world use case would not have this inherent bias against our method, and believe that users would be more likely to use loops with higher confidence scores. Therefore, we think that the results would be of even higher quality in practice. To test this, we examined the subset of loops generated by LoopMaker that received a naturalness rating of 4 or 5 and found that only 25% of the transitions in these loops had abrupt transitions, whereas 57% of the transitions from manually created loops were abrupt.

Finally, we simply asked each participant in the study if they feel that our system is useful and if they would like to use it. 80% of the participants answered this question positively.

CONCLUSION

We have presented an interactive system to assist novices and experts in creating music loops. Our user studies helped identify tedious and challenging parts of the loop creation process. Our system automates these parts, allowing users to concentrate on the more creative aspects of the process. These studies suggest that loops created using our system are significantly easier to create and are of comparable quality to those created by experts. We aim to explore other aspects of audio content creation that enable users to concentrate on the creative aspects of the process.

ACKNOWLEDGMENTS

We thank our colleagues at Adobe Research for participating in our user study, and Karrie Karahalios for providing valuable feedback on our paper.

REFERENCES

1. Jim Aikin. 2001. Making loop music in a computer. *Keyboard* 27, 2 (02 2001), 134–135.
2. Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. 2014. Painting with bob: Assisted creativity for novices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 419–428.
3. William E Benjamin. 1984. A theory of musical meter. *Music Perception: An Interdisciplinary Journal* 1, 4 (1984), 355–413.
4. Adam Berenzweig, Beth Logan, Daniel PW Ellis, and Brian Whitman. 2004. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal* 28, 2 (2004), 63–76.
5. Nicholas J. Bryan, Gautham J. Mysore, and Ge Wang. 2014. ISSE: an interactive source separation editor.. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.
6. J. Brzezinski. 2009. Game sound: an introduction to the history, theory, and practice of video game music and sound design. *Choice* 46, 7 (03 2009), 1323.
7. Mark Cartwright, Bryan Pardo, Gautham J Mysore, and Matt Hoffman. 2016. Fast and easy crowdsourced perceptual audio evaluation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 619–623.
8. Matthew E.P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. 2014. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 12 (2014), 1726–1737.
9. Nicholas Davis, Holger Winnemöller, Mira Dontcheva, and Ellen Yi-Luen Do. 2013. Toward a cognitive theory of creativity support. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*. ACM, 13–22.
10. Daniel PW Ellis. 2007. Beat tracking by dynamic programming. In *Journal of New Music Research*, Vol. 36. 51–60.
11. Bill Gibson. 2005. *The S.M.A.R.T. Guide to Producing Music with Samples, Loops, and MIDI*. Artistpro, Boston, MA, USA.
12. Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. 2002. RWC Music Database: Popular, Classical and Jazz Music Databases.. In *ISMIR*, Vol. 2. 287–288.
13. Tatsunori Hirai, Hironori Doi, and Shigeo Morishima. 2015. MusicMixer: computer-aided DJ system based on an automatic song mixing. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*. ACM, 41.
14. Ning Hu, Roger B Dannenberg, and George Tzanetakis. 2003. Polyphonic audio matching and alignment for music retrieval. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* IEEE, 185–188.
15. Tetsuro Kitahara, Kosuke Iijima, Misaki Okada, Yuji Yamashita, and Ayaka Tsuruoka. 2015. A loop sequencer that selects music loops based on the degree of excitement. *Proceedings of the 12th International Conference in Sound and Music Computing, SMC 2015* (2015), 435–438.
16. Meinard Müller. 2007. Dynamic time warping. In *Information Retrieval for Music and Motion*. 69–84.
17. Bee Suan Ong and Sebastian Streich. 2008. Music loop extraction from digital audio signals. In *IEEE International Conference on Multimedia and Expo*. 453–469.
18. David E. Reese, Lynne S. Gross, and Brian Gross. 2009. *Audio production worktext: concepts, techniques, and equipment*. Taylor & Francis.
19. Steve Rubin, Floraine Berthouzoz, Gautham J. Mysore, Wilmot Li, and Maneesh Agrawala. 2000. Content-based tools for editing audio stories. In *Proceedings of International Society of Music Information Retrieval*. ACM, 113–122.
20. Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. 2008. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing* 16, 6 (2008), 1138–1151.
21. Sebastian Streich and Bee Suan Ong. 2008. A Music Loop Explorer System. In *International Computer Music Conference Proceedings*.
22. Adobe Creative Team. 2012. Extending and Shortening Musical Selections in Adobe Audition CS6. (2012). `http://www.adobepress.com/articles/article.asp?p=1867758&seqNum=7`.
23. Newspapers Tribune. 2004. Making music; Software makes it easy for amaterus to compose tunes. (04 2004).
24. Brian Whitman. *The Infinite Jukebox*. `http://labs.echonest.com/Uploader/index.html`.
25. Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch: Face sketching assistance for novices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 407–417.