Conference Presentation

ACERemix: A Tool for Glitch Music Production and Performance

Cunningham, S., Weinel, J. and Griffiths, D.

This is a paper presented at the 9th Audio Mostly: A Conference on Interaction With Sound. 01-03 October 2014, Aalborg, Denmark

ACERemix: A Tool for Glitch Music Remix Production and Performance

Stuart Cunningham, Jonathan Weinel and Darryl Griffiths
Creative and Applied Research for the Digital Society (CARDS)
Glyndwr University
Mold Road, Wrexham, LL11 2AW, North Wales, UK
+44(0)1978 293583
{s.cunningham | j.weinel | griffiths.d}@glyndwr.ac.uk

ABSTRACT

In this paper we discuss the use of a recently developed audio compression approach: Audio Compression Exploiting Repetition (ACER) as a compositional tool for glitch composition and remixing. ACER functions by repeating similar sections of audio where they occur in a file and discarding the repetitive data. Thresholds for similarity can be defined using this approach, allowing for various degrees of (dis)similarity between materials identified as 'repetitive'. Through our initial subjective evaluation of ACER, we unexpectedly discovered that the compression method produced musically interesting results on some materials with higher levels of compression. Whilst listeners demonstrate this level of loss of fidelity to be unacceptable for the purposes of compression, it shows potential as a performance or production tool. When applied to pop songs the predicable form of the music was disrupted, introducing moments of novelty, while retaining the songs quantized rhythmic structure. In this paper we propose the use of ACER as a suitable method for producing sonic materials for 'glitch' composition. We present the use of ACER for this purpose with regards to a variety of materials that may be suitable for glitch or electroacoustic composition and using ACER in several different ways to process and reproduce musical audio.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *audio input/output.*

H.5.5. [**Information Interfaces and Presentation**]: Sound and Music Computing – *methodologies and techniques, signal analysis, synthesis, and processing, systems.*

General Terms

Algorithms, Design, Experimentation, Human Factors.

Keywords

Glitch music, remixing, DJ, performance, audio similarity, compression.

AM '14, October 01 - 03 2014, Aalborg, Denmark
Copyright 2014 ACM 978-1-4503-3032-9/14/10…$15.00.
http://dx.doi.org/10.1145/2636879.2636888

1. INTRODUCTION

In this work, we describe the inspiration and development behind a new tool: ACERemix, which is created to facilitate the production and performance of glitch music, that employs existing musical audio tracks as source material.

The system itself has evolved from a previous piece of research work that examined the potential of musical similarities, compositional and production repetition to be exploited for the purposes of data compression in digital musical audio [6, 7]. A by-product of this investigation was that the unwanted compression artefacts in the system often produced musically interesting pieces of audio, which were effectively remixes of the source material. Through further experimentation and observation, it was found that by deliberately altering the compression settings in specific ways; vibrant pieces of glitch music could be created and then subsequently mixed together and processed for production or performance scenarios.

The remainder of the paper is organised as follows: in section two we provide the reader with a brief overview of the field of glitch art and subset of glitch music; section three provides a summary of the original compression system and the key factors that are manipulated within the system for the purposes of generating glitch music; section four expands upon these parameters and discusses the approaches taken whilst experimenting to develop the ACER to produce glitch music; section five discusses the development and use of the ACERemix patch in MAX/MSP so that it can be used in a performance environment; and finally, in section six we offer a discussion of ACERemix in practice and its future development.

2. RELATED WORK

2.1 Glitch Art

Glitch art is primarily concerned with the creation of, often abstract, modernist, artistic works by employing technical deficiencies or flaws within a technological tool or environment. These errors, or glitches, usually introduce imperfections or what engineers might regard as noise into an otherwise satisfactory piece of art, be it an image, music, sound, and so on [10, 18]. With particular regard to the digital domain or environments that could be considered immersive, an alternate terms of 'strange ontologies' is also proposed, to encompass glitches or failure phenomena inconsistent with the real world [22].

Such is the contemporary interest and appeal of glitch art that there are several websites where users can upload image content and have glitches inserted. This is especially useful in providing examples of glitch art, as seen in Figure 1, which is a photograph of a street scene in Aalborg, Denmark processed using the *image glitch experiment* tool [12]. This particular example exhibits several noticeable glitches, including: misalignment of sections of the image; introduction of noise; banding effects; level manipulation; and colour masking.

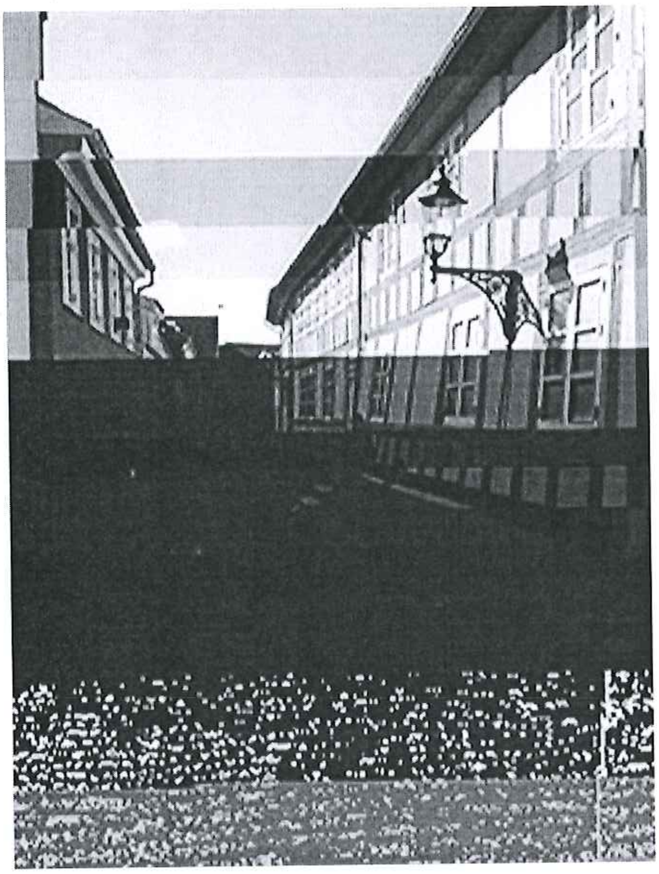

Figure 1: Photo of Aalborg, Denmark processed as Glitch Art

In many ways, the formulation of glitch art forms may be considered a process of reconstruction or remixing, insomuch as the artist is able to create glitch works either from material originally created by themselves or by working with material created by another. Though pre-dating the notion of 'glitch', William Burroughs's use of the 'cut-up technique' (a technique that involves cutting up and rearranging pre-existing text to produce new original output) during the 1950s and 1960s could be seen as an early form of glitch; a kind-of manual exploitation of the potential for linguistic messages to be disrupted using technology, in order to form new messages which adapt or pervert the original meaning [3].

It could be argued that the use of technology to introduce noise or otherwise disrupt the original source material enables an element of originality to be contributed; indeed, Burroughs was known to utilise this technique in the creation of various novels, which are now generally considered classic and important works of fiction. On the other hand, the use of such tools in an arbitrary, random or unskilled fashion, such as one might argue is offered by sites such as [4] and indeed, the various online 'cut-ups generators' [15] perhaps suggests a different view. This might be broadly considered to fit broadly with Barthes's notion *The Death of the Author* [1], but opens a debate about whether authorship and whether this resides with the technology itself or the artist operating the equipment. Such a debate goes beyond the scope of this paper.

Pushing a technology beyond its limitations or intended use often reveals glitch features. These artefacts are usually unwanted within the intended context but can then be represented or reused for other purposes, chiefly in the creation of artist works. There are examples of this being desirable within various artistic and performance contexts [19, 17, 8, 13]. In the case of audio and music, glitch-based experimentation and production for music is a well-established practice.

2.2 Glitch Music

In terms of glitch as it relates to music, the presence or failure of technological noise in music reproduction is not new or exclusive to the domain of digital audio. For example, analogue mediums such as vinyl records and the turntables used to play them incorporate a variety of glitch elements. These include: dust and scratches on the surface of the record, playback being stuck in a loop, and records being played at the wrong speed. These idiosyncrasies inherent in the technology, along with other potential for misuse (according to the manufacturers specification) became features that musicians and producers would activity exploit. For example, the technique of scratching involves the potential to play records incorrectly, using abnormal speeds and repetitions to form rhythmic sounds. Similarly, artists such as Public Enemy were known to deliberately mistreat vinyl that they intended to sample, in order to produce a more 'raw' sound [14]. Such techniques as these, and the incorporation of vinyl noise, records skipping or playing at the wrong speed, are also embedded in subsequent postmodern musical genres such as Trip-Hop[1]. A comprehensive discussion of glitch music can be found in [2].

Indeed, Cascone [5] cites contemporary glitch music as being rooted in DJ remixing and performance, that tools were available to bring together a range of sonic sources in a cohesive manner and using modern technology to bind together and represent this material in a range of unique ways. However, since our work here is concerned with digital glitches, it is that domain upon which we focus now. The digital domain offers a wide array of noise and failure sounds, and sound manipulations that can be integrated into musical works.

The aforementioned, key, seminal text in the field of glitch music comes in the form of Kim Cascone's article *The Aesthetics of Failure: 'Post-Digital' tendencies in contemporary computer music* [5]. Cascone's work describes the evolution of glitch music as being founded largely in the Machine Age and early 20th century, initially embracing the sounds of new technological transformation as tools to be foregrounded and explored by composers and listeners. A significant milestone of attracting listeners' attention to the usually unheard, unwanted musical sounds is cited as being John Cage's influential *4'33"* [4]. In more contemporary, computer-based, forms of glitch music, existing tracks are often reduced down to their basic, often microscopic, components and simplified. This is a key resonance with the approach of the ACERemix system.

[1] Trip-hop is a form of down-tempo electronic music originating in Bristol (UK) during the 1990s. For an example, please see Portishead's album *Dummy* [11]

Glitch music generally encompasses three interwoven themes and opportunities: the ability to create new sonic material from what Cascone would consider 'background' sounds; the ability to represent or remix existing sonic or musical audio; and the ability to produce music, or noise, through one or both of the previous two approaches. In our own work here, we are primarily concerned with the representation of existing musical audio in a form that might be considered music or noise.

3. MUSIC SIMILARITY PROCESSING

In previous research, the authors developed a system for the data compression of musical audio that exploits the presence of musical repetition and performance similarities in audio recordings of music, rather than adopt psychoacoustic models and seek to adaptively reduce quantisation levels in frequency sub-bands. This particular compression system is known as *Audio Compression Exploiting Repetition* (ACER) [6, 7]. We provide a brief explanation of the salient points about the operation of ACER in the remainder of this section, interested readers are directed to the full papers for a more complete explanation of ACER's operation.

Fundamentally, the ACER system performs a search for self-similarity within musical audio recordings, the aim being to remove musical sequences or segments that are repeated exactly elsewhere in the recording, thus removing redundant information, or by removing segments deems sufficiently similar, thus removing perceptually irrelevant information. As such, a significant part of this search process is informed by the duration of sequences that are searched for and the level of similarity between two segments that is set as being acceptable to the user or listener.

In terms of duration, the ACER system is typically configured to search for sequences of audio, that are identical or similar, with durations that are musical beat multiples. An example of identical sequence durations being identified is provided in Figure 2 and Figure 3, where the red patches highlight the first instance of a sequence and the green patches show subsequent repetition of this sequence. In the case of using ACER for compression, this is typically two, four, or eight beats, especially in the case of music conforming to 'standard' 4/4 tuning, where common sequence lengths contain 2^n beats, where $n \in \mathbb{Z}^+$.

Figure 2: Matched sequences with duration of a beat

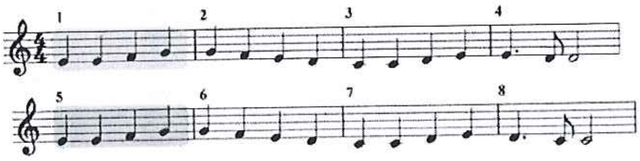

Figure 3: Matched sequences with duration of 4 beats

As for similarity threshold, this was determined following a series of listener experiments and scale of similarity identified that is used to operate the ACER system. This is a five-point scale with 'identical' on one end and 'different' on the other. When using ACER in the domain of data compression of musical audio, settings typically provide more satisfactory results closer to the 'identical' end of the scale, although this results in lower amounts of data reduction [6].

It was this similarity threshold that first led to the inspiration of ACERemix as a system for producing glitch music. During an initial listener testing phase, it was observed by the researchers that setting the ACER system to use a similarity threshold equal to, or greater than, the 'different' marker, extremely unpleasant and awkward sounding audio would subsequently be produced. This led to further experimentation with the similarity threshold and then to experimenting with varying sequence lengths in tandem; these tests led to the implementation of ACER as a tool for producing odd, glitch-type music: ACERemix.

4. ACERemix: GLITCH PRODUCTION

The primary ways that the ACER system has been adapted to produce glitch remixes is in the two aforementioned categories: the length or duration of the musical sequence that are searched for using ACER; and the level of similarity deemed acceptable between to sequences for them to be considered a match. To facilitate this, the ACER system was partially rewritten; with amendments to support the production of glitch remixes of the source content.

In terms of the duration of sequences, beat-related measures were again deemed desirable, but a conscious effort was made to use small segments of musical audio, typically less than a second in length. The purpose here was to provide more textural aesthetic to the resultant audio and this sequence choice also means that few words of melodic sections would ever appear in their entirety, giving the resultant music a much less cohesive feel and making it often difficult for the listener to identify structural cues in the music. This choice of particularly short segments of musical audio to create larger compositions broadly falls within the domain of microsound, which is often considered to be a precursor, or particular form, of glitch based music [21]. Whilst it is certainly true that using extremely small, varied fragments of musical audio result in glitch type music, glitch music could also consist of much larger segments of audio, re-arranged in a random way, and also qualify as glitch. As such, the definition itself is broad. Nevertheless, the use of short sound samples, or microsound, adds particular aesthetic and textual interest to the musical content produced using this modified ACER approach.

To this extent, the experimentation with ACER in creating glitch music focused upon selecting very small samples of audio but to attempt to maintain relation to the original piece and simplify the ACER search process, these short samples were also related to the tempo of the original music. As such, and inverse to the use of ACER for data compression, sequence lengths for 'standard' 4/4 music were now selected that were of duration 2^n beats, where n becomes the set of all negative integers $n \in \mathbb{Z}^-$. In musical terms, this produces remixed versions of music where the glitch segments lengths, relative to the tempo, are half notes; quarter notes; eighth notes; sixteenth notes; thirty-second notes, and so on. For each iteration of the ACER process to create glitch music the length of these sequences can be varied, meaning it is possible to create multiple glitch remix versions of the same source audio.

The approach to amending the similarity metric in the ACER system was also designed to reflect some of the unknown and unpredictable aspects of glitch music and glitch art, in that there should be an element of surprise and uniqueness each time the source is processed. To this extent, the similarity threshold was assigned a value, each time a match comparison took place, which was generated using a random number. This means that for any comparison that takes place between sequences, the system might look for matches that are absolutely identical to matches that are completely the opposite, and everything in between. To this extent, the boundary of possible values goes beyond those in the five-point perceptual similarity scale mentioned earlier, meaning it is actually more likely that extremely dissimilar matches will be found for any comparison. This meant that the ACER system produces a unique remixed, each time it is applied.

These amendments resulted in experimentation with a variety of musical material. In particular, it was found that interesting results would be acquired when using music that already exhibited simple, repetitive structures, little or no vocals, and which was highly quantised in terms of its rhythm. As such, it was found that music broadly conforming to the genres of pop; rap; dance; house; and dub-step were particularly successful when this modified ACER system was applied. Despite this, when applied to choral or chant music, traditionally perceived as fluid in terms of rhythm, the system would produce rhythmic, textured, almost dance-like, results.

Whilst it is difficult to appreciate the audio qualities in any visual example, to provide the reader with an insight into the resultant audio from the ACER system, consider the following waveform images, which are two second extracts from the same part of an original and ACER glitch version of a piece of chant music. Figure 4 shows the first two seconds of the original piece of musical audio. Figure 5 also shows the first two seconds of the same piece, but an ACER remixed version, which was created using a sequence duration of a thirty-second beat and the previously discussed random similarity threshold technique.

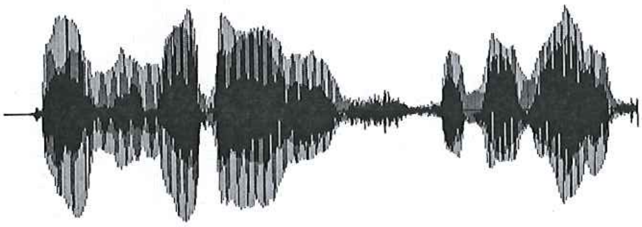

Figure 4: Two Second Chant Sample (Original)

Figure 5: Two Second Chant Sample (1/32 beat ACER)

There is a notable difference in the waveform shapes between these two examples, although it can be seen that they still broadly conform to a similar, macro level, structure. These changes are due to other segments of the piece being identified as matches for segments within this window and replacements being inserted accordingly, thus creating glitches around the structural and sequence elements of the audio. Another artefact that becomes evident is the introduction of distortion, which is introduced by the waveform writing function in ACER attempting to interpolate between short segments of audio that are sharply cut-off and the fact that neighbouring segments may now be completely unrelated. This can be illustrated by examining the spectrograms from the original sequence, shown in Figure 6, and the same time frame from the ACER remixed version using the thirty-second note segment durations, Figure 7, noting particularly the cross-frequency vertical bars, indicating brief bursts of noise.

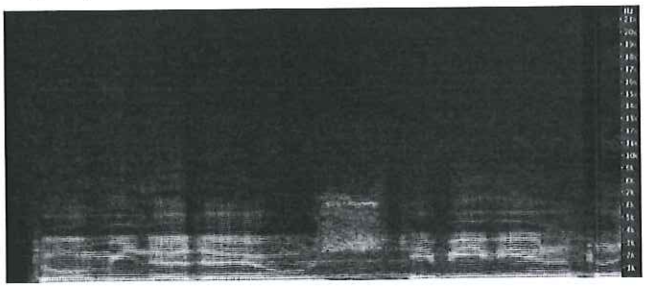

Figure 6: Spectrogram of Two Second Chant Sample (Original)

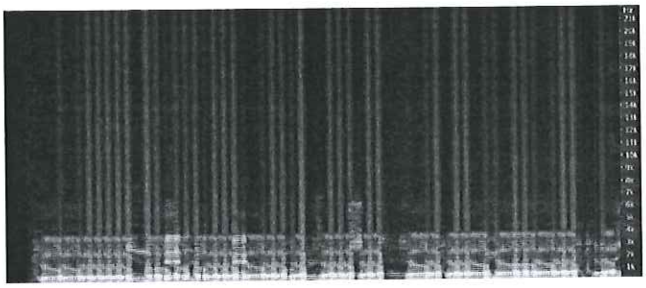

Figure 7: Spectrogram of Two Second Chant Sample (1/32 beat ACER)

This adds additional glitch properties, in particular the introduction of a rolling textural quality and very short noise or distorted elements.

5. ACERemix: GLITCH PERFORMANCE

After adjusting the ACER system to produce glitch type music, the next apparent step was to consider how it might be used in a performance or interactive way. Particularly given the nature of the source music that we were working with and the highly rhythmic nature of the results the system produces, the most obvious use for the system was as a performance or live remixing tool for DJs. This was especially true, given the obvious aesthetic connections to non-digital DJ techniques mentioned in section two such as scratching and looping of vinyl recordings. This led to the development of the performance version of the system: ACERemix.

A key consideration at this stage was how best to use the content that ACERemix could produce. One limitation of the original ACER system is that it is not suitable for real-time processing, since the search and matching procedure that it follows is computationally intense. As such, it was identified that the ACERemix tool, in its present incarnation, would have to make use of pre-remixed versions of the source material. However, given that the ACER system was able to produce glitch remixes that could vary in their sequence length by beat duration and that each remix is unique, this provides considerable diversity.

A patch was developed in the MAX/MSP software for the purposes of using ACERemix material for performance, the interface for which is shown in Figure 9. This contains traditional transport and level controls in addition to a number of features for performance of glitch remixes.

It is useful at this stage to note the work of Mason [16], which broadly discusses glitch art but, in particular, emphasises the value of remix as a particular type of glitch art form. Mason further argues that glitch remixed content is better appreciated by an audience when they have the original, un-glitched, media available to them and, even more so, if they have some indication as to the process that the media has gone through to arrive at its glitch remixed form.

With particular consideration to the need to be able to reference the source material, one of the primary functions of the software is the ability to switch between, in real-time and without interrupting the sequence of the piece, different beat duration ACERremix versions of the source material. This means that the source music is available as well as glitch versions that have been producing using whole note; half note, quarter note; etc. segments and that these are time synchronised in the software allowing instantaneous switching between them. This can be seen in the top-middle section of Figure 9 under the header 'ACER Quantize', where variations, down to sixteenth note versions, can be switched between.

In addition to allowing such comparisons between the original (unaltered) material and the glitch versions, the switching capability enables rhythmic performative switching between the respective audio streams to occur. This essentially extends a performance technique that originates in DJ mixing and turntablism: the DJ 'switch', where the DJ momentarily switches the audio to an alternate, synchronised track. This technique can be seen to serve as a momentary rhythmic divergence, which adds novelty and 'teases' the audience with a glimpse of an upcoming track.

As producers began to design records specifically for DJs to mix (in electronic dance music for example), sounds of the switch became incorporated into the production of records. Eventually, this became a pervasive feature of electronic dance music. This is perhaps best evidenced in current popular music by the technique used by dubstep/electro artists such as Skrillex, who rapidly switches the timbre of single bass lines between multiple different timbres within a single bar. This technique can be heard in Skrillex *Bangarang* [20] (for example, at 1:20) and various other tracks within this genre, such as Doctor P's *Sweet Shop* [9] (for example, at 0:30), the latter of which also demonstrates the use of a bit-crushing effect[2], itself a form of glitch.

The switch function of ACERemix can therefore be seen in such terms, as a possible tool for creating rhythmic switching between unaltered and various glitch material, both for the purposes of comparison, and of introducing momentary novelty and variation, as can be seen as appropriate in the context of the aforementioned sub-genres of electronic dance music. In the context of the control surface utilised (Figure 8), the top row of square buttons (usually utilised for muting or soloing tracks in a Digital Audio Workstation (DAW)) are mapped to the switch features of ACERemix, enabling momentary rhythmic tapping etc.

To provide the performer or DJ with additional ways to manipulate the musical output of ACERemix, a number of effects were added to an effects rack, visible along the bottom of the ACERemix interface. These effects constitute: volume; stereo pan; overdrive; ring modulation; frequency filter; delay; reverb; and a Doppler shift. The interface offers users the facility to control the levels of these effects within the overall mix as well as controls particular to the effect itself. The use of buttons on the control surface (the lower row of square buttons seen in Figure 8) also enables the rhythmic use of effects, also similar to the discussion of DJ switch aesthetics discussed previously.

In general the control surface was seen as a suitable way to provide performative interaction with the software. In addition to those specific features mentioned, the user is able to access a range of other features and parameters of ACERemix simultaneously and without having to point-and-click a mouse. This affords the opportunity for the user to 'play' using the tool, since they can tap the controls and manipulate the faders in time to the music or to create purposeful rhythmic or melodic effects. To prototype this concept, a Korg Nanokonrol MIDI USB device, depicted in Figure 8, was used and the controls mapped to ACERemix. This device more than adequately demonstrates the concept, though ACERemix utilises a standard MIDI implementation and as such could be used with a variety of other control surfaces.

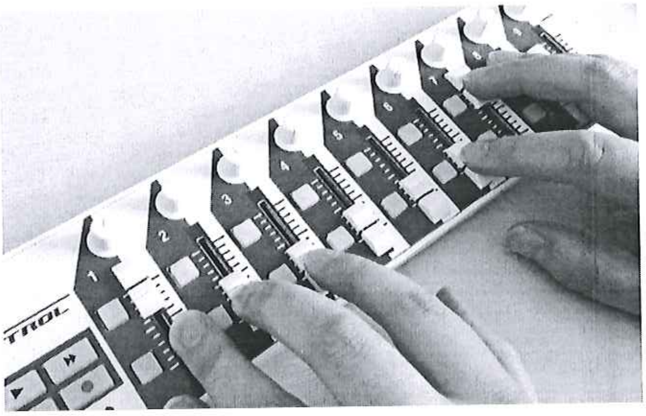

Figure 8: Control Surface use in ACERemix Performance: The transport controls were mapped to the appropriate transport controls in the ACERemix software, the top channel buttons were used to select which of the ACERemix duration sizes was playing at any given time, and the channel pan was mapped to the effect specific controls, channel fader controlled the effect level, and the remaining channel button used to switch each effect on and off.

6. DISCUSSION AND FUTURE WORK

The process that has been implemented works well in delivering a mechanism for producing glitch-based remixes of audio material. The range of settings that can be manipulated provide a diverse range of opportunity for production of unique remixes. The system is particularly well-suited to being used in a performance environment, as supported by the ACERemix software and associated control surface hardware. This also provides the opportunity for the system itself to have a wider, possibly more commercial appeal, since an audience do not have to listen to a piece of music that is entirely glitch-based, potentially failing to

[2] Bit-crushing effects simulate a reduction in bandwidth on the audio input source, and are often used to signify the quality of audio heard on outdated hardware such as 8-bit or 16-bit computer systems and video games consoles.

maintain their interest. Instead, the performer is able to dynamically switch to, and introduce, glitch based elements only where it serves to change or add some diversity to the original piece, for example, As with all performance tools of this nature there is something of a learning curve to learn which effects, ACER versions, and transitions work well to provide acoustically pleasing results.

In terms of future developments, the use of the ACER system in detecting similarity is being refined. The nature of using a random threshold means that the resultant glitch music can often be extremely repetitive. This is primarily due to the way the ACER search works, starting and the beginning of the audio and looking for matches later on. When using a random threshold this means that the first few sequences in the audio often end-up being repeated frequently throughout the file. There are several strategies that can be implemented to address this issue, but the most obvious one is to cap the number of times any given sample can be matched elsewhere; this value itself may even be determined randomly. This said, it is often difficult to criticise the resultant audio as unacceptable since this phenomenon is itself a type of glitch.

7. REFERENCES

[1] Barthes, R. 1977. The death of the author. *Image, music, text.*

[2] Bates, E. (2004), Glitches, Bugs, and Hisses: The Degeneration of Musical Recordings and the Contemporary Musical Work. In Washburne, C. and Derno, M. (eds), *Bad Music: Music you Love to Hate*. Routldege, New York, USA pp. 212-225.

[3] Burroughs, W. S., & Gysin, B. 1978. *The third mind.* Viking Press.

[4] Cage, J. 1993. *4' 33"*. London, UK: Edition Peters. (Original work composed 1952).

[5] Cascone, K. 2000. The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music. *Computer Music Journal*, 24, 4, 12-18.

[6] Cunningham, S., & Grout, V. 2013. Data reduction of audio by exploiting musical repetition. *Multimedia Tools and Applications*, 1-22.

[7] Cunningham, S., Weinel, J., Roberts, S., Grout, V. and Griffiths, D. 2013. Initial objective & subjective evaluation of a similarity-based audio compression technique. In *Proceedings of the 8th Audio Mostly Conference* (AM '13). ACM, New York, NY, USA, , Article 1 , 6 pages. DOI=http://doi.acm.org/10.1145/2544114.2544116

[8] den Heijer, E. 2013. Evolving glitch art. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design,* 109-120). Springer Berlin Heidelberg.

[9] Doctor P, 2010. On *Circus One Presented By Doctor P & Flux Pavilion* (2011). [CD]. Sweet Shop. London, UK: Circus Records.

[10] Downey, J. 2002. Glitch art. *Ninth Letter*, 118.

[11] Gibbons, B., Barrow, G., & Utley, A. 1994. *Dummy.* [CD]. London, UK: Go! Beat.

[12] Github, 2014. *image glitch experiment.* Available at: http://snorpey.github.io/jpg-glitch/ [Last accessed: 10th May 2014].

[13] Gross, S. 2013. Glitch, please: datamoshing as a medium-specific application of digital material. In *Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces* (DPPI '13). ACM, New York, NY, USA, 175-184. DOI=http://doi.acm.org/10.1145/2513506.2513525

[14] Hale, J. (Producer/Director), 2011. Public Enemy: Prophets of Rage. In P. Bullock, *Black Music Legends of the 1980s.* United Kingdom: British Broadcasting Corporation.

[15] Languageisavirus, 2014. *Cut-Up Machine.* Available at: http://www.languageisavirus.com/cutupmachine.html#.U3zuqihvDR4 [Last accessed 22nd May 2014].

[16] Mason, S. 2012. Glitched lit: possibilities for databending literature. In *Proceedings of the 2nd workshop on Narrative and hypertext* (NHT '12). ACM, New York, NY, USA, 41-44. DOI=http://doi.acm.org/10.1145/2310076.2310086

[17] Menkman, R. 2011. Glitch studies manifesto. *Video Vortex Reader II: Moving Images Beyond YouTube*, Institute of Network Cultures, Amsterdam, 336-347.

[18] Menkman, R. 2011. *The glitch moment (um).* Amsterdam: Institute of Network Cultures.

[19] Sheppard, R.M., Kamali, M., Rivas, R., Tamai, M., Yang, Z., Wu, W., and Nahrstedt K. 2008. Advancing interactive collaborative mediums through tele-immersive dance (TED): a symbiotic creativity and design environment for art and computer science. In *Proceedings of the 16th ACM international conference on Multimedia* (MM '08). ACM, New York, NY, USA, 579-588. DOI=http://doi.acm.org/10.1145/1459359.1459437

[20] Skillrex, 2012. Bangarang. On *Bangarang* [digital download]. US: Big Beat.

[21] Thomson, P. (2004). Atoms and errors: towards a history and aesthetics of microsound. *Organised Sound*, *9*(2), 207-218.

[22] Whitelaw, M., Guglielmetti, M., and Innocent, T. 2009. Strange ontologies in digital culture. *ACM Comput. Entertain.* 7, 1, Article 4 (February 2009), 13 pages. DOI = 10.1145/1486508.1486512 http://doi.acm.org/10.1145/1486508.1486512

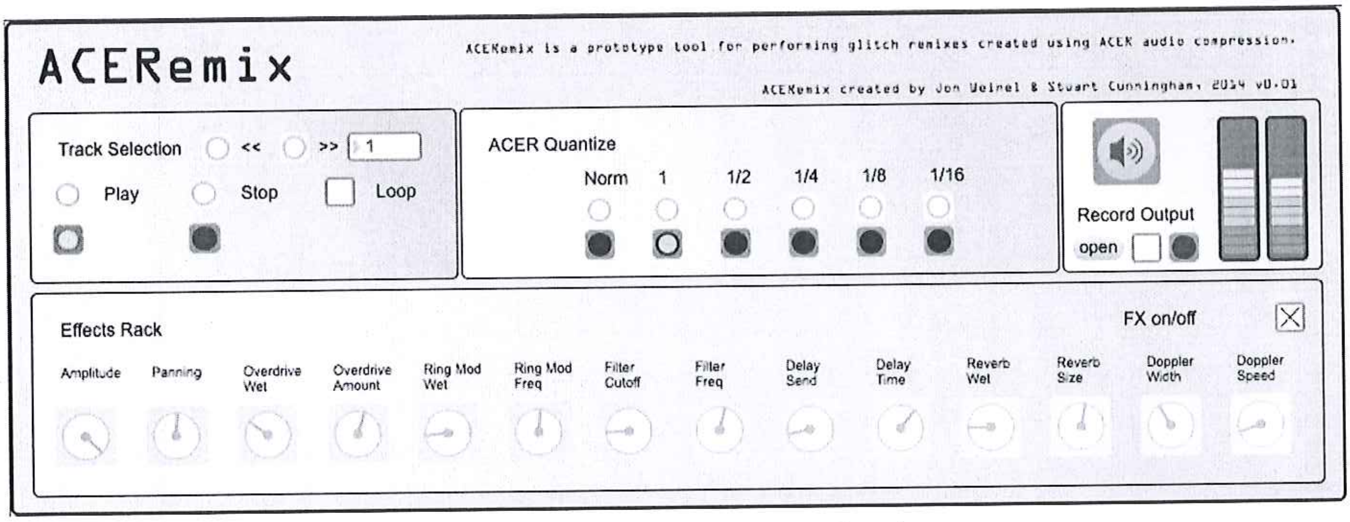

Figure 9: ACERemix Performance Tool Interface