

# Note sequence morphing algorithms for performance of electronic dance music

René Wooller and Andrew R. Brown  
Queensland University of Technology &  
Queensland Conservatorium, Griffith University  
renwooller@gmail.com  
andrew.r.brown@griffith.edu.au

## Abstract

This paper describes algorithms that can musically augment the realtime performance of electronic dance music by generating new musical material by morphing. Note sequence morphing involves the algorithmic generation of music that smoothly transitions between two existing musical segments. The potential of musical morphing in electronic dance music is outlined and previous research is summarised; including discussions of relevant music theoretic and algorithmic concepts. An outline and explanation is provided of a novel Markov morphing process that uses similarity measures to construct transition matrices. The paper reports on a “focus-concert” study used to evaluate this morphing algorithm and to compare its output with performances from a professional DJ. Discussions of this trial include reflections on some of the aesthetic characteristics of note sequence morphing. The research suggests that the proposed morphing technique could be effectively used in some electronic dance music contexts.

**Keywords:** music, morph, morphing, mixing, generative, metamorphosis, electronic dance music

## 1. Introduction

The musical morphing processes presented here utilise a mode of engagement that is similar to mixing but involves a more sophisticated integration of musical materials; at the level of note-sequences, rather than audio waveforms. The goal is to provide a novel and effective extension to the live musical capabilities of electronic dance music (EDM) producers who currently use note sequences in studio production, but are presently required to render these to audio recordings for mixing and manipulating during live performance.

### 1.1. Motivations

Despite developments in live electronic music performance software (Wooller 2003, Ableton 2004), the status quo for the presentation of EDM is two turn-tables/CDJs and a two-channel mixing desk. This simple interface has stood the test of time and the morphing software we have developed taps into a similar mode of engagement. That is, selecting source and target music that is pre-composed, and using a single parameter, the “morph-index,” in the primary control area traditionally defined as “crossfade”. However, despite the usability and reliability of the turn-tables and two channel mixer, it is difficult to alter the melodic, rhythmic and harmonic structure effectively. Motivated to overcome this problem, the morphing software we have developed utilises note sequences and is thus well suited to such manipulations.

### 1.2. Note sequence morphing in electronic dance music

Morphing is a pattern-merging process paying particular attention to continuity and coherent combinations of form. The topic of this paper is note sequence morphing of EDM, focusing

on musical constructs such as pitch and rhythm, rather than sonic elements. From here on in, this process will simply be referred to as morphing and the resulting musical material as a morph. Thus, morphing algorithms take source and target tracks, named S and T, and produce the morph named M. Realtime morphing algorithms also take the morph-index,  $\Omega$ , representing the progress of the morph (analogous to the position of a cross-fader), and the current time (in beats) of the music, t. Extending from Polansky's functional definition (Polansky 1992), we have  $M = f(S, T, \Omega, t)$ . The central problems of morphing emerge from the features that differentiate it from simple transitioning: how is continuity and coherence implemented? What ways exist to combine the form of S and T? How can we apply the aesthetics of new morphing techniques to existing EDM contexts, maintaining acceptable limits of divergence but without simply mimicking current "two channel mixer" practice? In this paper we hope to provide some answers to these questions.

## 2. Previous research

While the application of note sequence morphing specifically to EDM is new, there exist a small number of investigations into morphing of note sequence (typically MIDI based) music and associated music theoretic studies of modulation and transition. The interpolation of sonic parameters for timbral morphing have been investigated somewhat more, but those studies are less relevant to the current discussion.

### 2.1. Systems for note sequence morphing

There have been a small number note sequence morphing systems to date. While a review is available (Wooller 2006), some key works will be summarised here, including *GRIN* (GRaphical INput), *HMSL* (Hierarchical Music Specification Language), *DMorph* and *The Musifier*.

Mathews and Rosler (Mathews and Rosler 1969) created the first note sequence morphing system, GRIN. GRIN used envelopes for parameters such as pitch and duration. The envelopes of source and target were combined, weighted on the morph index, and converted into a note sequence for playback. Polansky applied morphing algorithms within *HMSL* (Burke, Polansky et al. 2005) to some of his compositions, including *Distance Music* (Polansky 1987), *Bedhaya Guthrie/Bedhaya Sadra* (Polansky 1996), *51 Melodies* (Polansky 1991), *Two Children's Songs* (Polansky 1992) and *Road to Chimachum*. MIDI renderings are available online (Polansky 2006).

Oppenheim created *DMorph* (1995) as a tool within *DMix* (Oppenheim 1994), discussed in detail as a patent (Oppenheim 1997). *DMorph* was realtime, interactive and dealt with 'n-source' morphs; which extends the original definition of morphing to include any number of inputs. The process involved grouping source and target notes and interpolating the note parameters (pitch, duration, and dynamic), thus creating a new note from each grouping. The software is protected, however, Oppenheim has numerous demonstrations.

Jonas Edlund developed *The Musifier* (2004), which utilises note sequence morphing. The Musifier is an 'orchestra-pit' agent – accompanying computer game-play. It performs n-source morphing on human composed themes which are linked to game state parameters, for example, the player's location or the proximity of enemies. Each theme reflects the emotional significance of each game state parameter and Edlund's morphing algorithm hybridises them, in realtime, based on the value of the game-state parameters. Edlund's techniques are secret, however a web application is available allowing a user to control theme parameters (Edlund 2006). Morphing between parts of different timbres has been adequately handled in MIDI by cross-fading the volume of parts on two different channels and sending identical note events to both channels. Edlund has developed a commercial prototype (Edlund 2006a), with demos showing instances where it can be effective.

## 2.2. Parameter morphing

Morphing in computer music research tends to focus on audio rather than note sequences, for example, interpolating filter coefficients, FM modulation ratios, or other synthesis parameters. Synthesis off topic, however, the approaches to parametric interpolation are applicable to note sequence morphing. It was first discussed by Babbitt (Babbitt 1965) and reviewed more recently by Momeni and Wessel (Momeni and Wessel 2003) and Bencina (Bencina 2005) when presenting their musical interfaces.

The *Metasurface* (Bencina 2005) interpolates between any number of parameter states positioned in two-dimensions. Although the parameters are mostly sonic rather than musical, some of these are border-line. For example, low frequency modulation can have a musically rhythmic effect. Momeni and Wessel utilise a Gaussian kernel at each state that provides a useful non-linear influence of musical states on the morph. Of particular relevance is 'Beat Space' (Momeni and Wessel 2003) where the parameters control probabilities of beat generation.

## 2.3. Morphing in music theory

Despite observations that morphing is an established compositional strategy (Oppenheim 1995) there is no definitive musicological literature, moreover, problems are addressed tangentially from various perspectives. For example, the continuous interpolation of pitch and rhythm in avant-garde music has been investigated (Callender 2004). In the Western music tradition, 'bridges' between musical sections are often composed but are rarely discussed specifically in music theory and, when mentioned, the overriding emphasis is on harmonic techniques (Carpenter 1983, Kamien and Wagner 1997). Key modulation (the transition from one key to another, such as from C major to F major) is the most well-developed morphing technique in music theory. Approaches to key modulation include direct modulations, pivot notes and progressions of dominant tonalities. In contrast, temporal modulation (moving from one tempo or speed to another) relatively recently emerged from avant-garde orchestral music.

Direct modulation involves jumping directly to the new key without any obvious transition section. It is common to have a cadence or end-phrase just before the new key occurs, to signal a change. Drabkin defines a pivot as "a chord (or a note) having different harmonic (or melodic) functions in two different keys, this property being used to effect a smooth transition from one key to the other" (2005). Key distance can be defined as the minimum number of pivot notes between two keys.

Another technique is to change a chord's function so it becomes the dominant of the target key, cadencing to the new key. The change in function warns the listener that a new key is immanent and hints what it is going to be. In jazz theory this is extended, changing the function of a chord to the second degree of the new key, which leads to the new dominant and finally the new tonic (ii-V-I). Often, the second is changed to a secondary-dominant, meaning that it has the function of a dominant that leads into another dominant chord.

While key-modulation occupies a central role in Western music theory, the concept of modulating from one metre, tempo or rhythmic pattern to another is examined less often, and usually only in the avant-garde. Two distinct issues seem to occupy theorists: 'tempo' or 'metric' modulation; and 'beat-class' modulation or 'rhythmic interpolation.' The former concerns the techniques to effect a smooth transition from one tempo or rate of play to another, while the latter deals with the problems of interpolating particular rhythmic patterns. Tempo (or metric) modulation, is defined as a transition from one tempo to another through a common sub-pulse.

No comprehensive theory of tempo modulation exists, and, despite being suggested by Fétis as a possibility as far back as 1832 (Arlin 2000), it has only recently come under closer

scrutiny (Benadon 2004). Late 19th century compositions display basic tempo modulation through hemiola (Lewin 1981), however, clear examples of more flexible tempo modulation were composed much later by Elliot Carter in *Variations for Orchestra* (1954-1955). In the same way as key modulation, tempo modulation utilises 'pivot tempo' which provides a common sub-pulse for source and target. Benadon (2004) outlines techniques for calculating pivot tempi.

Another problem is modulating from one rhythm to another, regardless of tempo. This is explored in some minimal music, where polyrhythmic beats cycle in and out of phase and *Piano Phase* by Steve Reich, is iconic of this. Early analysts of Reich's music (Warburton 1988; Cohn 1992) employed the notion of beat classes to describe rhythmic patterns when maximally in or out of phase. Callender (2004, p. 39-41) identified a variable calculated from the beat class set and sampling interval (morph index quantization rate) that, he argues, relates to the ability of the listener to predict the rhythmic patterns of the next sample period. In analysing more recent and less metrically confined works of Reich's, Roeder (2003) extended the notion of beat classes to deal with accent and pitch.

#### **2.4. DJ mixing agents**

Software applications designed for DJ mixing are relevant to this study, despite their obvious focus on audio rather than note-sequences, because they utilise a source and target to generate a transition as a kind of "morph" and attempt to systematise the procedure in a computer program. Examples of these include, Andersen (2005) who developed a system to enhance the transformational aspects of the fade by driving audio digital signal processing (DSP) effects with pulse information. Cliff (2000) identified effective beat-matching and synchronisation of pulse as important technical aspects of a transition, but focused mainly on generating play lists according to qualitative specifications such as tempo contour. Other researchers have been concerned with finding the appropriate crossover point in the music (Fujio and Shiizuka 2003). Jehan (2005), as part of the Skeleton software, developed an automatic DJ with tempo and downbeat extraction, alignment, pitch-independent time stretching and cross-fading. He employed an algorithm to find rhythmically similar sections within the two pieces that would be the most suitable for a transition. Fraser (2002) created an automatic DJ, with algorithms for tempo extraction, beat alignment and cross-fading. The cross-fade volume envelopes used during the transition could be drawn by the user, in contrast to typical DJ cross-fades which are limited to 'equal-energy' (logarithmic) curves.

### **3. Computational approaches to musical morphing**

In this section we will examine two different realtime morphing techniques, each developed to overcome observed limitations with previous attempts. The techniques discussed are weighted selection and probabilistic generation, and they lead into a discussion of our Markov morphing algorithm.

#### **3.1. Weighted selection**

Weighted selection, first introduced by Oppenheim (1995), probabilistically chooses between time-aligned sub-sections of either the source or target material, depending on the value of the morph-index. Throughout a transition, the morph-index value will increase, starting at 0 guaranteeing selection from the source, to 1 guaranteeing selection of material from the target. In-between, the likelihood of the morph containing fragments from either source or target shifts according to the morph-index. A possible outcome is depicted in figure 1. A significant problem with this approach is that the individual musical structures are fragmented and material in consecutive time-slices may not form a coherent progression.

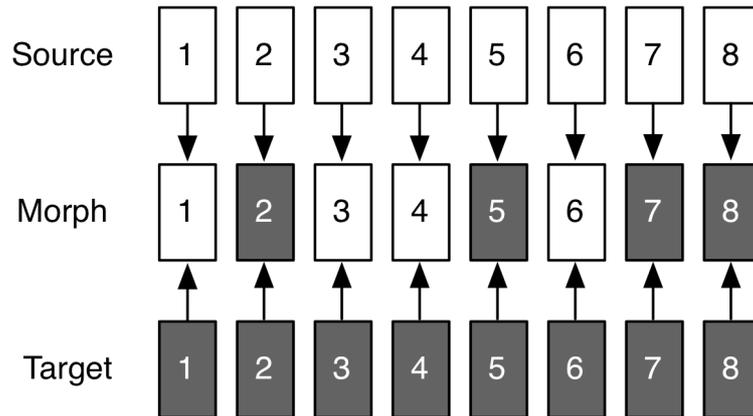


Figure 1. A diagrammatic depiction of the weighted selection morphing process.

One way to overcome this is to choose subsequent time-slices from either the source and target based on their similarity to the preceding time-slice in the morph. This strategy may increase the chances that the selected time-slices will follow coherently from each other. However, in practice this seems only to slightly increase the effectiveness of weighted selection for certain material that fit well together already. A logical extension of this approach is to concatenate sub-sections that appear to follow each other well regardless of which time-slice in the source or target they are from. For this extension, the time ordering for material selected from source and target becomes unfixed so the process can no longer be considered as weighted selection, rather it is a form of probabilistic generation.

### 3.2. Probabilistic generation

The probabilistic generation technique is a recombinatorial process that treats all sub-sections of both the source and target material as potentially available at every step during construction of the morph. A similarity assessment for each of the sub-sections is made against the most-recently added morph section and the similarity rating is used as a weighting for the probabilistic choice of the next section. This process treats the sub-sections more like a set than a sequence, as shown in figure 2, as a result of disassociating musical segments with their original temporal location. This approach enables morphs of any length to be constructed, rather than relating morph length to those of the source and target.

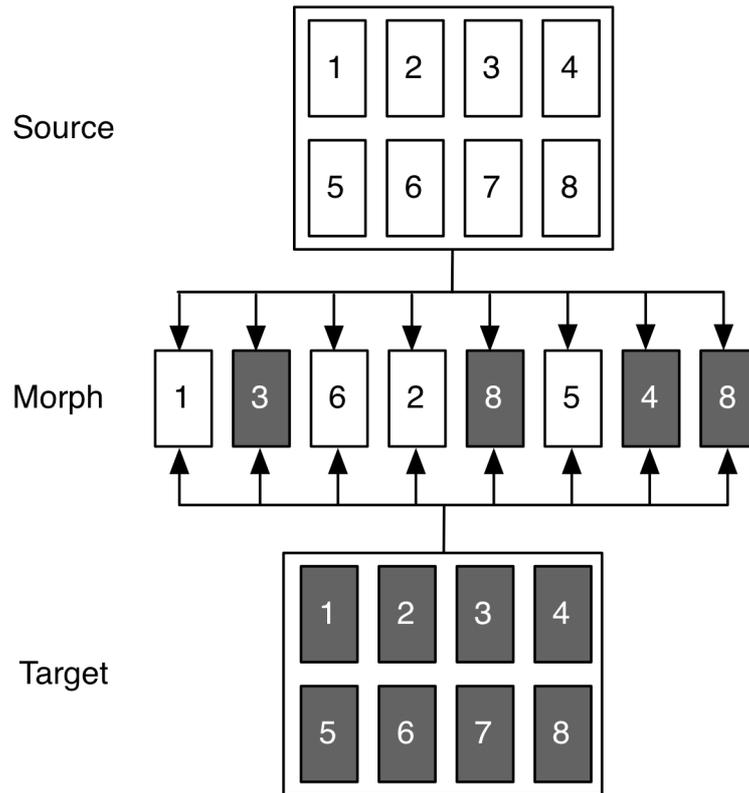


Figure 2. Sub-sections used as a set, disassociated from original temporal location.

Although probabilistic models have been employed to generate melodies through note-by-note prediction since almost the beginning of computer music history (Hiller and Isaacson 1958) they have not been applied to note sequence morphing. The standard process is to generate the entire sequence through a series of decisions about which note should follow the current one. When these decisions about succession are based on statistical likelihood the technique is often referred to as a Markov process. Typically in these processes the current or seed note influences a probability distribution in some way, from which a next note is drawn at random. The newly selected note is fed into the algorithm recursively. For example, Mozer (1994) formally tested the compositional output of Markov matrices and an Artificial Neural Network (ANN) as methods for this style of melody generation. Although his surveys found that ‘global coherence’ was lacking from the compositions, there is a possibility that, for short transitions, more typical of morphing passages, this attribute is less important and that adequate music can be generated. This supposition seems to be supported by research into the application of Markov processes to the problem of musical continuation or mimicking (Pachet 2002). The process of probabilistic generation we are describing has similarities to Markov approaches even though the probability settings are derived from similarity rather than proximity measures. It is therefore useful to explore in more detail the musical application of Markov processes.

A Markov model or Markov matrix describes the probabilities for generating a string of symbols that can be used to represent any sequential process. The probabilities can be automatically derived through statistical analysis of sequences; the more frequently one state follows another, the greater is the probability assigned to that transition in the model. Theoretically, states could be mapped to any musical element, for example, if they were mapped to every pitch class, this would mean each of the notes C, C#, D, etc. through to B are a state.

If we imagine there is a loop (just like the source or target material of a morph) playing these pitch-classes, C C C C# D, we could perform a simple statistical analysis to determine the probability of generating the three different pitches given the current pitch. There are five pitch transitions, C-C occurs twice, C-C# once, C#-D once and, given that this is a loop, D-C occurs once. Three of the transitions are from C to another pitch and thus if the current note pitch, referred to as the seed, is C, then the probability of generating a C again is 2/3, while C# is 1/3. If the seed is C#, the probability of generating a D is 1.0 (absolute) and with seed of D, probability of generating C is also 1.0. The probability of all possible state transitions is often represented in a matrix or grid, as shown in figure 3 for the simple example just presented.

Seed Pitch	C	2/3	1/3	0
	C#	0	0	1
	D	1	0	0
		C	C#	D
	Output Pitch			

Figure 3: Diagram of a simple Markov matrix for pitch transitions

If our seed data consists not only of the previous note, but the previous three notes, we say that the ‘order’ or ‘depth’ of the Markov model is three, while the example above was for a first order Markov process. The coherence of the generated sequences increases with depth, at the detriment of novelty.

### 3.3 Markov morphing

Our Markov morphing process is a probabilistic generation technique. Like most morphing processes, a Markov morph is assembled as a sequence of short musical segments. In this case segments are time slices of one musical part of, typically, half a beat in duration. The morph always begins with the first segment from the source material. A weighted selection process is used to decide whether to use material from the source or target to create a Markov matrix for generating the next segment. The most recent morph segment is used to build a one-dimensional Markov matrix that holds all possible extensions from the chosen material. A probabilistic choice is made from the matrix to select the next segment, and the process is repeated as many times as desired. This unconventional method of building a new one-dimensional matrix for each time slice was chosen for reasons of efficiency. It is theoretically equivalent to building two complete two-dimensional Markov matrices, one for the source and one for the target, and doing a weighted selection between these matrices.

In our implementation the user can specify the number of segments to use as a seed, thus determining the Markov depth of the search, in order to control the degree of stability (or similarity with the source and target materials) that can be achieved. Given the typically small size of the source and target files, it may appear that using this approach a match is unlikely or that the original material would simply be reconstructed in the morph. However, in

practice, by choosing a small segment size the resulting morph has an interesting mix of material that is sometimes even aesthetically pleasing.



Figure 4. An example Markov-morph over four bars (2 to 5), with a 68% null-prediction rate.

There is always a possibility that no match is found between the previous morph segment and segments from the selected material. This is more likely because we are dealing with short sections of music, not large databases as is normally the case with Markov models. On average, this problem of null prediction can occur up to 50% of the time, while the morph shown in figure 4 displays a 68% null prediction rate. We initially dealt with this by falling back on weighted selection for those occasions - which can be heard in the jamming together of fragmented structures in figure 4. However, a more subtle technique to avoid null prediction is to select 'similar' segments rather than exact matches when building the Markov matrix. This requires only a slight modification to the current process; instead of all occurrences of possible extensions having the same probability, their weighting is set relative to their 'similarity' to the seed. But how is similarity determined?

### 3.4. Similarity of notes

Segment similarity is based on the notes contained within the segment. The note representation that we employ consists of pitch, onset, duration and dynamic (loudness) attributes. The similarity measures for each of these are combined under user-controlled weightings that are explained more thoroughly elsewhere (Wooller 2006). While similarity for both duration and dynamic can be taken directly from linear differences in value, the same cannot be said for pitch and onset.

It is clear that distance measures are not effective for absolute pitch difference because they don't take account of harmonic relationships such as octave equivalence, and diatonic organisation. For example, by distance measures C4 bears more similarity to C4# and B3 (1 half-tone step) than to C5 (12 half-tone steps), despite the common perception of octave equivalence. To reflect music perception more accurately, one can transform pitch into a multi-dimensional space, such as Shepard's helix (Shepard 1982), and apply a Euclidian distance measure, as employed by Mozer (1994). Other models of similarity have also been developed (Hewlett and Selfridge-Field 1998; Deutsch 1999). We chose an algorithm that combines the difference in absolute pitch, pitch-class and distance in circle of fifths space (Shepard 1982) based on user defined weightings. This balances absolute, intervalic and harmonic distance considerations. Our software implementation allows the overall similarity threshold to be increased or reduced with a user-controlled parameter.

A similar technique is used as a similarity measure for note onset position. The absolute difference in onset is combined with the metric onset difference within cycles of two bars, one bar, half a bar, three beats, two beats and one beat. Each of these position measures can be weighted differently to allow for different time signatures and musical styles.

An interesting outcome of using similarity measures is that loosening similarity constraints reduces null prediction and increasing Markov depth increases null prediction, therefore these two parameters can be nicely balanced at some intermediate point to achieve a desired result.

The result of applying pitch and onset similarity to the same material as in figure 4 is shown in figure 5. The use of similarity rather than exact matching reduced the null prediction rate from 68% to 0, however a new problem called 'stream loss' is now introduced. If a prediction occurs, but the inter onset period between the selected note and the note prior to it does not match the time since the last note was played (inter onset to the current time), the note will not be used. If this happens enough times then there will reach a point where it is impossible to find a note with a long enough inter onset interval and this is 'stream loss'. In these cases, weighted selection is employed as a fall back. The aesthetic effect relative to weighted selection is one of sparser melodies but less fragmented and typically more coherent.



Figure 5. Markov morph using similarity.

A future improvement would be to reduce the scarcity by abandoning the strictly real time approach of making a new note prediction on every time slice, and instead generate the next note immediately after the current note is inserted, as with traditional note at a time prediction.

### 3.5. Implementation details and control features

So far we have discussed creating a morph within a single musical part. However, music is usually made up of several parts, tracks or channels. To accommodate this in our morphing software, each part (for example, drums, bass, lead) within source and target is allocated a single 'morph-channel.' Parts from the source and target that share a morph-channel will morph into each other during the transition. For example, depending on settings specified by the user, the bass part from one track could morph to either the bass, lead or drum part of the target, but not to all three at once. Source and target musical material is read in as MIDI files by our software and so, in this case, a part corresponds to a MIDI track. If parts from the source and target in the one morph-channel are on different MIDI channels, the notes generated by the algorithm are played on both MIDI channels, and the volume is cross-faded. This results in a simplistic timbral morph for that part. The gradient and cross-over point of the volume cross-fade can also be specified.

For each independent morph-channel the user can specify a morphing algorithm and set of parameter values. The morph-index, which is a simple linear function, can have a separate user-defined gradient and offset for each morph-channel. This allows the operator to decide how quickly and at what point during the morph the changes to musical structure take place. It makes sense to transition between parts with similar roles, for example, morphing drums into lead will usually produce random sounding percussive hits because the notes

specified by the lead part are being played on the MIDI channel of the drums. Morphing tonal parts that have widely different registers can produce unusual results. If parts being morphed have the same MIDI channel but do not share the same MIDI instrument, undesired results often occur due to timing problems involving program change messages. As with Oppenheim's (1995) system we leave responsibility for managing these situations to the user who can condition the MIDI data prior to use.

In our morphing software, playback of MIDI data can be affected by meta-musical parameters such as tempo, rhythmic quantize, swing, and loop-length. Values for these parameters from the source and target are interpolated, each with a similar gradient and offset function, over the duration of the morph.

#### **4. Trial concert**

In order to obtain feedback on the musical effectiveness of this approach to morphing we conducted a trial concert. The objective of the trial concert was to gauge the potential of note-sequence morphing in EDM contexts and receive feedback on aesthetic differences between morphing and current EDM practice.

##### **4.1. Method**

An audience of eleven volunteers who had at least moderate awareness of EDM were randomly selected from various musical backgrounds. They were played live morphing transitions between three EDM songs and completed a questionnaire about how applicable the morphs were to the EDM contexts they were familiar with. The first part of the questionnaire gauged their 'EDM credibility' by ascertaining the breadth of participant familiarity with EDM genres, how many times they had gone to dance music events over the last year, and when they last went. This information was used to group them into levels of EDM credibility. The second part of the questionnaire asked them to comment on the musical characteristics of what they heard. To establish a relative measure, similar questions were posed regarding a live DJ mix that used the same three tracks that were morphed. The first stage of testing was on 'short transitions' of less than 10 seconds where the aim was to smoothly cross over to the next track as simply as possible. The second stage was on 'long transitions' with durations up to several minutes, where the transition was seen as more of a creative work itself.

The three tracks were selected from a pool of thirty for each genre of house, drum'n'bass and trance (ninety all up). They were chosen according to how indicative the music is of the genre and how easily it might be replicated. As well as this, divergent pieces were chosen to test the ability of the morphing algorithms to cross stylistic boundaries. The songs selected were *Flatbeat* by Mr.Oizo for house, *Decoy* by Stakka and Skynet for drum'n'bass, and a fusion of *Wunderbaum* by Hypersonic Whomen and Computers and *Microprocessors* by Logic Bomb for trance. The main sections of the songs were replicated, with a little elaboration, using a custom-built MIDI sequencer driving Reason 2.5 synthesizer. To provide the DJ with music to work with, seven minute versions of the three songs were recorded from the MIDI reconstructions, with variations in the tracks limited to automating the muting and un-muting of layers.

An example of the type of morphing effect on EDM is shown in figure 6, and audio renderings of the examples used can be heard on the [www.limorph.com](http://www.limorph.com) web site. This example illustrates a short morph with two tracks, drums in the lower section of the piano roll display and bass above. The transition of is most easily visible in the bass line where the intervallic range difference between source and target is distinct.

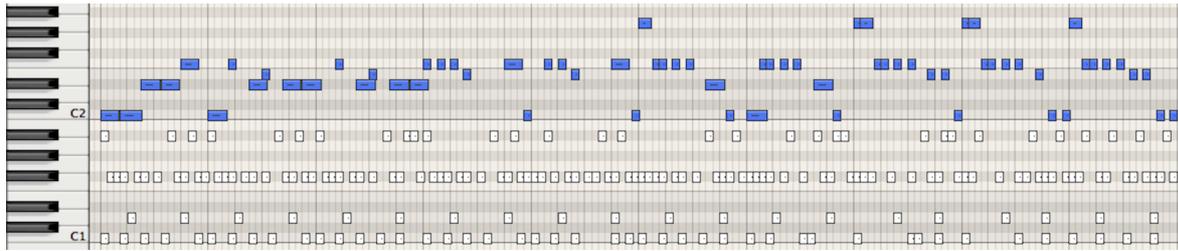


Figure 6. Piano-roll notation of a morph of drum and bass parts.

## 4.2. Results

The background EDM information given by each participant was used to calculate an ‘EDM credibility’ rating on a scale of 1 to 10. This was based primarily on frequency with which they attended EDM events and the range of genres they were familiar with. The average grade was 7, with a standard deviation of 2.14. The minimum rating was 3 and maximum 10. This value was used to weight responses according to their credibility. Interesting, applying this weighting did not have a significant effect. It therefore appears that opinions about the morphing outcomes amongst this group had little to do with their prior experience with EDM. When asked if they would expect to hear a DJ producing similar results to the benchmark DJ mixes that were played, participants unanimously said yes. When asked whether, at an EDM event, they might hear a short morph similar to those played, 4 said yes, 6 said no and 1 was undecided. When asked whether, at an EDM event, they might hear a long morph similar to the one played, 1 said yes, 7 said no and 3 were undecided. In open-ended comments about stylistic authenticity, a common observation was that DJ mixing always maintains tempo during the transition in order to perform beat matching, while the morphing software interpolates it. In the words of one participant:

“It seemed like the DJ actually beat-matched the two tracks, so that the tempo of the beat remained the same throughout - whereas it sounded like the morphing software sped up and one slowed down the other so that it ended faster than it started. This could be good sometimes but not always.”

Participants were also asked to compare the algorithmic morphs with live performed transitions from a DJ performing with the same material at the trial concert. For short transitions, no participant preferred morphing, 4 preferred DJ mixing and 7 were undecided. Most of those who were undecided liked both, but for different reasons. For example:

“I liked both - I like to watch skilled DJ's and find their mixing choices interesting on a personal level. The morphing had some cool rhythmic stuff happening - also, it goes beyond what I'm used to hearing in mixing and so was interesting in that respect.”

It should be noted that, during performance of the short morphs, the DJ spontaneously and unexpectedly increased the length of his mixes, meaning that, for the short morphs, the DJ mixing was perceptibly longer than the computer-generated morphs, which influenced the preferences. This induced some comments, for example:

“Perhaps the DJ mixing appealed more because the excerpts were longer. It was difficult with the very short morphing excerpts to compare it to the DJ.”

For long transitions, the preferences were significantly different to those for short transitions. 4 participants preferring the morphing, 3 preferring the DJ mixing and 4 were undecided. One comment from a participant who preferred the morphing was:

“Wow, very impressive. There is so much added musicality. Would keep people wondering where each sound is from.”

This comment relates to the fact that the synthesizer can trigger new sounds when rendering new MIDI notes, whereas no new sounds can be introduced into a DJ mix. For example, the pitch for 'cow bell' on the source drum kit could be the pitch for 'whistle' on the target, and if the target doesn't contain this pitch, a sound that exists in neither source or target (the whistle), will have been heard in the morph.

Participants were asked to score the morphs on their applicability to real electronic music contexts. Short morphs were rated only moderately applicable, with an average score of 64% with standard deviation of 22%, while the long morphs were judged highly applicable, with a rating of 78% and standard deviation of 17%. Weighting these results on EDM credibility did not seem to change them significantly. All participants agreed that the DJ mixes performed at the concert were similar to those of real EDM contexts, suggesting that any data gained regarding comparative measures between the DJ and the morphs are likely to apply to real world settings. Because the tracks used in the trial concert were selected according to how different they were, not how similar they were, the study purposefully ignored the ‘art of track selection’ that is part of the DJ’s skill. Future research could morph between songs that were selected by the DJ, or allow the DJ to use their own record collection to ‘construct’ a morph however similar tracks tend to be much easier for the morphing software to deal with, and thus less interesting in terms of morphing algorithm research.

#### **4.3 Discussion of results**

These results indicate that by and large the morphing processes were musically acceptable, but that morphing is aesthetically different to current methods of mixing employed in EDM. Participant comments highlight two primary factors in the morphs; a tendency for an erratic structure produced by the stochastic nature of the techniques, and the unconventional ramping of tempo due to interpolation. To overcome these factors, future algorithms might use a database of musical patterns against which the structure can be fitted to provide more choice of musical segments, as well as the option of tempo-matching the target to the source rather than interpolating between them.

The audience showed a preference towards long morphing transitions and gave high scores on general stylistic applicability ratings, indicating that morphing techniques could be a viable new approach to EDM, rather than simply an alternative to DJ crossfading. However, the results can not be construed to mean morphing generally sounds better than mixing over long periods due to the unexpected change in transition length that occurred during the DJs performance and to comments about some of the inappropriately idiosyncratic artefacts occasionally produced by the algorithm. People also commented that the short morphs occurred too quickly for them to be able to judge them effectively so this portion of the data may be limited in its reliability. As well as this, because the tracks were played through for one minute before the DJ started to mix them, people were unsure of when the ‘short’ DJ mix had started. Before the ‘long’ mix, the DJ verbally expressed his misgivings at attempting such a difficult mix which may have also biased people to prefer the morphing software. A useful suggestion was to conduct an online survey where participants could listen to a transition as many times as necessary to make a judgement. This has been done, along with a new morphing algorithm, and results will be published in the near future.

## 5. Conclusion

Note sequence morphing involves the algorithmic generation of a musical transition between note sequences. Morphing is a potential development in the evolution of EDM practice which is currently based on cross fading between audio tracks. We have presented an overview of some of the issues involved in designing musical morphing algorithms and analysed potential morphing strategies including weighted selection, probabilistic generation and described a new Markov morphing algorithm that uses similarity measures rather than exact matches when constructing Markov matrices. This algorithm has been implemented into a software morphing system for experimentation and trials.

We report on a trial concert conducted to gauge the applicability of the current morphing algorithms for EDM which suggested that morphing may indeed be useful, although it involves aesthetics that are somewhat unfamiliar when compared to existing mixing practices. In the case of longer morphs, a slight majority of the audience preferred morphing to traditional mixing, however, further tests that are more controlled or realistic, such as online surveys and in-situ performances, will be needed to build more reliable evidence. We look forward to further testing and to developing morphing algorithms that are even more convincing aesthetically, effective across a wider range of musical styles and that contribute further to understanding of musical structures. Full audio clips of current morph examples and a version of the morphing software program are available online at [www.limorph.com](http://www.limorph.com).

## 7. References

- Ableton. 2004. *Live Tour - Performance* Retrieved 15th of April, 2004, from <http://www.ableton.com/index.php?main=live>.
- Andersen, T. H., 2005. In the Mixxx: Novel Digital DJ Interfaces. In proceedings of *Conference on Human Factors in Computing Systems, CHI '05*, Portland, OR, USA, ACM Press.
- Arlin, M. I., 2000. Metric Mutation and Modulation: The Nineteenth-Century Speculations of F.-J. Fétis *Journal of Music Theory* **44**(2): 261-322.
- Babbitt, M., 1965. The use of computers in musicological research *Perspectives of New Music* **3**(2): 74-83.
- Benadon, F., 2004. Towards a theory of tempo modulation. *ICMPC8 Proceedings of the 8th International Conference on Music Perception and Cognition*, Evanston, IL, ICMPC.
- Bencina, R., 2005. The Metasurface - Applying Natural Neighbour Interpolation to Two-to-Many Mapping. In proceedings of *New Interfaces for Musical Expression*, Vancouver, BC, Canada.
- Burke, P., L., Polansky, et al. 2005. *HMSL* Retrieved 7th of March, 2006, from <http://www.softsynth.com/hmsl/>.
- Callender, C., 2004. Continuous Transformations. *Music Theory Online* **10**(3).
- Carpenter, P., 1983. Grundgestalt as tonal function. *Music Theory Spectrum* **5**: 15-38.
- Cliff, D., 2000. *Hang the DJ: Automatic Sequencing and Seamless Mixing of Dance Music*. Hewlett Packard.
- Cohn, R., 1992. Transpositional Combination of Beat-Class Sets in Steve Reich's Phase-Shifting Music. *Perspectives of New Music* **30**(2): 146-177.
- Deutsch, D., Ed. 1999. *The psychology of music*. New York, Academic Press.

- Drabkin, W., 2005. *Pivot*. Retrieved 6th of August, 2005, from <http://www.grovemusic.com>.
- Edlund, J., 2004. *The Virtues of the Musifier: A Matter of View*. Retrieved 16th of February, 2006, from <http://www.interamus.com/techTalk/musificationAndView.html>.
- Edlund, J., 2006. *Morphing Demo Server*. Retrieved 13th of April, 2006, from <http://www.interamus.com>.
- Edlund, J., 2006a. *Musifier Virtual Orchestra beta version*. Retrieved 7th of August, 2006, from <http://www.interamus.com/musifier/download.htm>.
- Fraser, M., 2002. *An Automatic Software DJ*. Undergraduate project dissertation, Software Engineering, Sheffield, University of Sheffield.
- Fujio, T. and Shiizuka, H., 2003. A system of mixing songs for automatic DJ performance using genetic programming. In proceedings of *6th Asian Design International Conference*, Tsukuba.
- Hewlett, W. B. and Selfridge-Field E., Eds. 1998. *Melodic Similarity: concepts, procedures, and applications*. Cambridge, Mass., MIT Press.
- Hiller, L. and Isaacson, L., 1958. Musical Composition with a High-Speed Digital Computer. In, Schwanauer, S. and Levitt, D. Eds. 1993. *Machine Models of Music*. Cambridge, MA: MIT Press.
- Jehan, T., 2005. *Creating Music by Listening*. Program in Media Arts and Sciences. Massachusetts, MIT.
- Kamien, R. and Wagner, N., 1997. Bridge Themes within a Chromaticized Voice Exchange in Mozart Expositions. *Music Theory Spectrum* **19**(1): 1-12.
- Lewin, D., 1981. On Harmony and Meter in Brahms's Op. 76, No.8 *19th-Century Music* **4**(3): 261-265.
- Mathews, M. V. and Rosler, L., 1969. Graphical Language for the Scores of Computer-generated Sounds. *Music by Computers*. H. V. Foerster and J. W. Beauchamp. New York, John Wiley and Sons, Inc.: 84-114.
- Momeni, A. and Wessel, D., 2003. Characterizing and Controlling Musical Material Intuitively with Geometric Models. In proceedings of the *Conference on New Interfaces for Musical Expression*, Montreal, Canada.
- Mozer, M. C. 1994. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing. *Connection Science* **6**: 247-280.
- Oppenheim, D. V. 1994. Slappability: A New metaphor for Human Computer Interaction. In M. Smith, A. Smaill and G. A. Wiggins (eds.) *Music Education: An Artificial Intelligence Approach*, Edinburgh 1993. London: Springer-Verlag. pp. 92-107.
- Oppenheim, D., 1995. Demonstrating MMorph: A System for Morphing Music in Real-time. *International Computer Music Conference*, Banff Canada, ICMA, pp. 479-480.
- Oppenheim, D., 1997. *Interactive system for compositional morphing of music in realtime*. USA, IBM.
- Pachet, F., 2002. The Continuator: Musical Interaction with Style. *International Computer Music Conference*, Göteborg, Sweden, pp. 211-218.
- Polansky, L., 1987. Distance Music I-VI for any number of programmer/performers and live, programmable computer music systems. *Perspectives of New Music* **25**(1/2): 537-544.

- Polansky, L., 1991. *51 Melodies*. Artifact Recordings/Frog Peak Music.
- Polansky, L., 1992. More on Morphological Mutations:Recent Techniques and Developments. *ICMC*, San Jose.
- Polansky, L., 1996. Bedhaya Guthrie/Bedhaya Sadra for Voices, Kemanak, Melody Instruments, and Accompanimental Javanese Gamelan. *Perspectives of New Music* **34**(1): 28-55.
- Polansky, L., 2006. *midifiles.demo.front* Retrieved 8th of March, 2006, from <http://eamusic.dartmouth.edu/~larry/midifiles.demo.front.html>.
- Roeder, J., 2003. Beat-Class Modulation in Steve Reich's Music. *Music Theory Spectrum* **25**(2): 275-304.
- Shepard, R. N., 1982. Geometrical approximations to the structure of musical pitch. *Psychological Review* **89**: 305-333.
- Warburton, D., 1988. A Working Terminology for Minimal Music. *Integral* **2**: 135-159.
- Wooller, R., 2003. *Researching and Developing Software for Live Electronic (Dance) Music*. Master of Music dissertation. Creative Industries. Queensland University of Technology. Brisbane, Australia.
- Wooller, R., 2005. *LEMu 2: Morpheus Tutorial*, from <http://www.lemu.com.au/>.
- Wooller, R. 2006. Review of Compositional Morphing:Works, Techniques, Applications and Possibilities. *Medi(t)ations: computers/music/intermedia the Australasian Computer Music Conference*, Adelaide, Australia: Australasian Computer Music Association, pp.172-177.

### Notes on contributors:

Rene Wooller completed his PhD in 2008, focusing on techniques for automated and interactive note sequence morphing in mainstream electronic music. For now he is a freelance software developer, casual research assistant at the Queensland University of Technology and itinerant computer musician.

Andrew R. Brown is a computational artist, researcher and educator. He is Professor of music and Sound at the Queensland University of Technology (QUT) in Australia, and was previously the Research Manager for the Australasian CRC for Interaction Design (ACID). Andrew is an active computational artist working in music and visual domains. His research interests include technology support for creativity and learning, computational aesthetics, and the philosophy of technology.