6 Probabilistic morphing algorithm

Having discovered only mediocre musical results from parametric morphing, another approach utilising probability, prediction and similarity measures was developed instead. The particular probabilistic morphing algorithm I developed is called the *Markov Morph* and is the topic of this chapter. The *Markov Morph* exhibits more musical coherence than the parametric morphing algorithm described in the previous chapter, which is not to say that parametric morphing is an approach that is flawed in general. The *Markov Morph* is also a more original development within the field of compositional morphing, as the parametric approach had been explored originally by Mathews and Rosler (1969). Musically, the *Markov Morph* is distinguished by a characteristic unpredictability in style. The *Markov Morph* was initially inspired by Markov probability techniques, but has since been modified, fairly drastically, to suit the particular conditions of realtime operation and small sample sizes.

The *Markov Morph* involves the weighted selection of source or target, creation of a probability distribution and the generation of notes. The technique is fairly efficient for the current requirements and thus possesses a high degree of realtime flexibility. It is also able to generate musical extensions and, being stochastic, the morphs are able to continually change even when the morph-index is fixed. Sophisticated note similarity measures allow elements of musical style to be controlled. While some previous research has been conducted into probabilistic techniques for morphing (Didkovsky 1997; Momeni and Wessel 2003; Oppenheim 1995; Polansky 2006), the *Markov Morph* contains developments that are novel to note sequence morphing, including conditional probability and note comparisons in a continuous space.

From a composer-agent standpoint (explained in 3.1.3), the *Markov Morph* models the **heuristic** approach to composition. The music is composed bottom-up, note-by-note, estimated from the recent output as it is generated and using probabilistic rules that are extracted from source or target. There is an intention for the music to be based on the existing styles but not necessarily a clear musical goal. There is no ‘perspiration’ that is indicative of the **trial** approach, nor is the focus on exploration of mappings, as with the **abstract** approach. The *Markov Morph* utilises **analytic**, **transformational** and **generative** algorithms, with a medium level of contextual breadth.

The first section, 6.1, will describe the techniques at the basis of the *Markov Morph* algorithm, including weighted selection, probability distributions and note similarity measurement. The process can be summarised thus: within each play cycle (usually at quarter beat intervals) either the source or target is selected through random selection weighted on the morph index. The recent history of note output is compared to each note in the selected source or target and a list of similarity-to-history ratings is created. The similarity ratings are used as a probability distribution to generate (or not generate) the next note from the selected source or target for that play cycle.

Musical outcomes from the *Markov Morph* were subjected to the scrutiny of a focus group and a focus concert, both of which are detailed in section 6.2. The former study was developmental, aiming to streamline formal evaluation processes. The latter was a functional test that established the competitiveness of morphing when compared to DJ cross-fading, particularly in situations where the source and target music are very different. Interesting stylistic characteristics of morphing became apparent through the comments of the participants and some ideas for further improvements to the algorithm were obtained. The focus concert itself appears to be a fairly original evaluation methodology for computer music, particularly for compositional morphing (Clarke and Cook 2004). While the data gathered in the focus concert was mostly valid, the experience was also useful for improving the evaluation techniques in future studies (see Chapter Seven).

Overall, the probabilistic morphing algorithm is an important part of the research into automated and interactive compositional morphing due to it being a novel approach with some advantages over traditional techniques such as DJ cross-fading. As with parametric morphing, there are a number of extensions to the current probabilistic algorithm that may be pursued to increase its musical efficacy and these are discussed in section 6.4.

6.1 Description

Firstly, it should be noted that for all of the note sequences described below, notes that occur with the same onset are grouped into the same event, rather than existing as separate elements within the list. Therefore, I will often refer to '**note-groups**' rather than notes. The **note-group** is a vertical grouping (like a chord) and should not be confused with **note sequence**, which is a list of notes. Instead of note sequences, I will also refer to **note-group sequences**, which are lists of **note-groups**.

During each play cycle of the *Markov Morph* algorithm, the morph index influences a random selection of either the source or target. This process, called **weighted selection**, is at the basis of the *Markov Morph* and is described in more detail below (6.1.1). In the simplest case, with the user specified **order** (as in, 'Markov order/depth') at 1, each note-group in this selected sequence is compared to the last note-group that was played and assigned a rating of similarity to it. The exact similarity measures are detailed in 6.1.3. With order above 1, each similarity rating will be conditioned on (multiplied by) the similarity rating of the notes immediately prior, so that short sequences of note-groups are compared, rather than just single note groups. The final similarity ratings are then used as a probability distribution to select the next note that will be played, which assumes conditional independence.

6.1.1 Weighted Selection

Weighted selection is a morphing technique that selects between the source and target during each play cycle. It was first introduced by Oppenheim (1995). Let Ω be the morph index that is normalised such that when $\Omega = 0$, the source is playing and when $\Omega = 1$, the target is playing. During each play cycle (usually distanced at quarter beat resolution), the probability of selecting the source is $1 - \Omega$ and the probability of selecting the target is Ω. Any notes in the selected source or target that start within the bounds spanned by the play cycle are then played.

The effect is that small segments of music are spliced together haphazardly. It produces quite interesting, although often bizarre, transitions between parts that have many notes and a strong sense of metre, such as drums. However, in sparser parts, important notes can easily be missed, greatly reducing the ability to perceive coherent phrases in the music. This problem is partially solved in the *Markov Morph* by using various statistical properties of notes as a selection measure rather than purely the occurrence of note onset.

6.1.2 *Markov Morph* overview

As with weighted selection, the *Markov Morph* starts by selecting either the source or target. However, rather than playing all of the notes in the selected pattern that are within the span of the current play cycle, the whole note-group sequence is compared with the recent output, using note-group similarity measures to generate a probability distribution to select the next note-group. The probability distribution is regenerated each play cycle because the source and target music, as well as a number of user-defined parameters, may change in realtime. The current implementation of the algorithm is able to function in realtime, given moderately sized source and target loops of under twenty note-groups in each part.

This approach deviates somewhat from the standard statistical approach of Markov Chains, where the similarity measure would return either only 0 or 1, depending on whether an exact match was found or not. The standard method only works when there are many items to form a comparison with, for example, a database of music with over 600 note-groups would probably be sufficient. In the context of morphing short loops of music in *LEMorpheus*, the sample space is clearly insufficient, often being less than 20 note-groups. With this small amount of note-groups, situations when the source and target do not contain any of the same notes are quite likely. In these situations and with a discrete 0 or 1 similarity measure, none of the notes would be given a probability of occurrence. In which case, the *Markov Morph* would fall back on weighted selection. In response to this, the more complex, continuous similarity measures that are currently used were developed.

In order to fully explain the *Markov Morph*, I will now describe in detail what constitutes the seed, how segments are compared to the seed and how notes can be generated using the similarity measurements.

Defining the seed

The seed is the list of note-groups of the most recent output, with length equal to the user-defined variable, 'depth' (Markov order). To define this formally, let the list of notes in the entire history of playback be H. Each play cycle, as note-groups are generated, they are appended to the end of H. Let the length of H at the current point in time be n and let the user-defined depth be d, which ranges from $1 \leq d \leq 12$. Let the list of notes that constitutes the seed be P. Provided that $d \leq n$, P consists of the d most recent elements of H, or $P = H_{n+1-d}, H_{\ldots}, H_n$. In this case, d also serves as the length of P.

If the morph starts playback before any notes have been played, H is initialised with the notes from the source, S, or target, T, depending on the direction of the morph: forward or backward respectively.

Comparing segments with the seed

Let the list of note-groups selected through weighted selection be W. Recalling that Ω is the morph index, the chance that $W = S$ is $1 - \Omega$ and the chance that $W = T$ is Ω. Each note-group in W, along with the note-groups preceding it, will be compared to the seed P (see above). Let X contain similarity measurements between P and all the note-group sequences in W that are of the same length as P, wrapping where needed. For example, if the depth is two, $d = 2$, and the length $|W| = 4$, then P, that is, P_1, P_2, would be compared to each of W_1, W_2, W_2, W_3, W_3, W_4 and W_4, W_1. Each of these comparisons would be a value in X.

Let m be the length of W and X, which are necessarily the same length. Let i be an index with the range $1 \leq i \leq m$ that points to the tail (last note-group) of a note-group sequence in W that is being compared to P. When $d > i$, the head and body of the segment wrap around through the end of W. Recall that P is the list of seed note-groups from the recent history of length d, the user specified depth/order. X_i will hold the similarity between P and the string of notes in W of length d that ends on W_i.

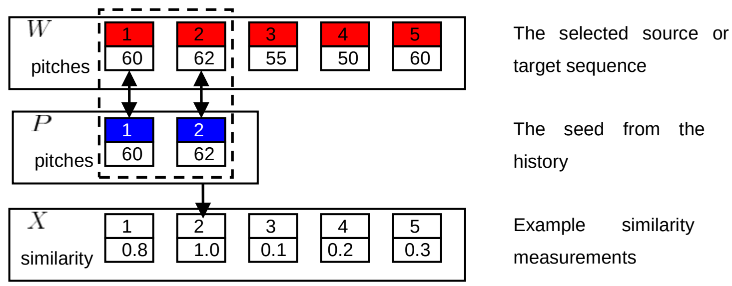


Figure 1 **An example of how similarity measurements between segments of the selected source or target sequence and the seed are created. In this simplified case, only monophonic pitches are being compared and the order (length of seed) is only two. It is clear that the first two note pitches are the same as the seed, 60 and 62, so the similarity measure relating to the tail of the segment (index 2) is 1.0. None of the other two-note segments in W match the seed and so the other similarity ratings are less than 1.0. Note that the sequence wraps around – the segment W_5 to W_1, 60 and 60, is fairly close to P, 60 and 62, which is apparent in the relatively high similarity rating for index 1 of 0.8.**

Recall that m is the length of W and X, d is the length of the seed and that i is an index pointing to the tail of the segment in W that is being compared. Allowing for wrap-around, the sequence $W_{((i-d) \bmod m)+1}, W_{\ldots}, W_i$ is being compared to the sequence $P_1, P_{\ldots}, P_d$ to generate the similarity measurement in X_i. If $d > m$, the sequence W would need to be looped until $m \geq d$, however, this has not been implemented as such a case is quite rare.

Factoring each element in the segments being compared

Let sim be a function that measures the similarity of two note-groups, returning a value between 0, least similar, and 1, most similar. The details of the similarity measure used will be provided later. Recall that W_i is the tail of the note group sequence being compared to the history. The similarity measurement of the most recent note-group in P and W_i is given by $sim(P_d, W_i)$. For cases where the user sets $d = 1$, the similarity matrix can be defined thus: $X_i = sim(P_d, W_i)$. For cases where $d > 1$, X_i will also be factored by the similarity of the note-groups preceding W_i to the note-groups preceding P_d. Let g be an index with the range $0 \leq g < d$. Recall that e is the index pointing to the head of the segment in W that is being compared to P. Let α be a normalisation constant which ensures that X sums to 1. Using the notation '$|\ |$' to indicate the cardinality (length) of the sequences, the similarity ratings are defined thus:

$$X_i = \alpha \prod_{g=0}^{g=d-1} sim(P_{g+1}, W_{((i-d+g) \bmod m)+1}) \qquad \text{for } 1 \leq i \leq m,\ d = |P| \text{ and } m = |W| = |X|$$

Equation 1 **Function to calculate the similarity matrix X. W_{i-d} is the head of the segment of length d in W that ends on W_i, which is to be compared to P. The $\bmod m$ is to allow for wrap around. α is a normalisation constant.**

Generating notes from the similarity matrix

Given that X_i holds the similarity between the seed, P, and the segment ending on W_i, how can this information be used to generate new notes? It is logical that the note-group which directly follows the segment in W that is the most similar to P should have a high likelihood of being played next – this way, if the music that is played would resemble W. Because of this, the similarity matrix can be treated as a complete (sum to 1) probability distribution matrix – the similarity value in X_i is the probability that $W_{(i \bmod m)+1}$ will be selected for playback. To select notes for playback, a note-group is chosen randomly according to the probability weights in X.

6.1.3 Markov Morph details

Combined similarity measure

The similarity measurement function that compares two different notes is a weighted combination of inverse distances between note pitch, duration and start-time. Dynamic, being less crucial, has not been included although it would be a beneficial addition for future work. The user controls the weights that affect the balance of each distance in the final value for similarity. Let sim be a similarity measure function that compares two notes, a and b, and returns the similarity rating. Let $pitchSim$, $durSim$ and $onsetSim$ (these are detailed below) be similarity measures functions for pitch, duration and onset respectively. Each of these takes two notes and returns a similarity rating between 0 and 1. Let w represent three user-defined weights for each of these similarity measure functions, each weight ranging from 0 to 1. Letting α be a factor such that the result of sim is normalised between 0 and 1, we have:

$$sim(a,b) = \alpha\,(w_0\,pitchSim(a,b) + w_1\,durSim(a,b) + w_2\,onsetSim(a,b))$$

Equation 2 The similarity between notes as the combined similarity of pitch, duration and onset, weighted by the user-defined weights.

The similarity measurement function is used to derive the similarity matrix, X, as explained in Equation 1.

Variable levels of similarity measurement contrast

I included a variable that controls the level of contrast in the similarity measurements, let it be k, which ranges from 0 to 1. It is used in a function, con, that transforms the similarity ratings in the similarity matrix X. When $k = 1$, the output of $con(X)$ will be effectively discretised to either 0 or 1. When $k = 0$, there is no difference. The musical effect of increasing the contrast is to reduce the level of randomness in the music being generated, by reducing the number of candidate note-groups. However, reducing the candidate note-groups also increases the likelihood of '**null predictions**', where $X = 0$.

The contrast function takes X to a power between 0 and 1000, defined by the user. The result is then normalised again between 0 and 1. With α as a normalisation constant:

$$con(X) = \alpha\,X^{1000k}$$

Equation 3 The contrast function. As k increases, the difference between low and high similarity ratings are compounded.

Values that are smaller than the arbitrary cut-off of 0.000001 are quantised to 0. While using an exponential function is currently sufficient, in future work, a sigmoid function would be more suited to accentuate the contrast, as it reduces low values and increases high values, instead of reducing all values (albeit, reducing the low values more than the high values).

Combined pitch similarity measure

The function that determines the similarity of note pitch, $pitchSim$, is a weighted combination of the similarity in linear pitch space (as per MIDI, from 0 to 127), the Circle of Fifths (CF) and the Circle of Chroma (CC). This pitch space is similar to that originally proposed by Shepard (1982) and therefore has some basis in music psychology. Let the linear pitch similarity function be $linSim$; the similarity in Circle of Fifth be $cofSim$; the similarity in CC be $chroSim$. Let v hold three user-defined weights and let a and b represent two MIDI pitches being compared. With α as a normalisation constant, we have:

$$pitchSim(a,b) = \alpha\ (v_0\, linSim(a,b) + v_1\, cofSim(a,b) + v_2\, chroSim(a,b))$$

Equation 4 **The similarity between two note pitches, a and b, as the combination of similarity in linear, CF and CC pitch spaces, each weighted by user-defined weights.**

Linear pitch similarity

The linear pitch similarity function, $linSim$ is the absolute difference between the two input pitches, normalised to the range from 0 to 1, taken to the 6^{th} power in order to magnify the resolution of smaller distances and finally subtracted from 1 so that 1 is totally similar and 0 is totally dissimilar. This simple similarity measure can be generalised. Let the range be r, which in the case of MIDI pitch equals 128, and let the magnification be m, which in this case equals 6. We have:

$$linSim(a,b) = \left(1 - \frac{|a-b|}{r}\right)^m$$

Equation 5 **Generalized similarity in linear space is the inverse of the difference, normalised by the range of possible values, r, and taken to the m^{th} power, in order to exaggerate the smaller differences. For MIDI pitch space, $r = 128$. I settled on the magnification $m = 6$ after experimentation.**

The 6^{th} power was chosen, after some experimentation with other powers, because it changed the curve of distances to an order of magnitude that seemed to roughly reflect my own intuitive musical understanding of the similarity measure more accurately, as well as making it more easily comparable to other similarity measures. For example, an octave difference, without the magnification, would rate a similarity of 0.9. With the 6^{th} power, this drops down to less than 0.6. The difference of a semitone, the most similar interval in linear pitch space, would be 0.99 without the power and 0.95 with it – this appeared to be suitable. Using a power that deviated from $m = 6$ by a small amount would be unlikely to have a significant effect on the algorithm.

CC and CF similarity measure

The similarity in the CC is the distance in pitch class, normalised between 0 and 1, and subtracted from 1. Let s represent the number of steps per octave. The default is $s = 12$, due to there usually being 12 steps per octave. In implementation, *LEMorpheus* does not use any other value than this default, however, it is described as a variable here so as to theoretically allow for other possibilities. Let a and b be the two input MIDI pitches (range 0 to 127) and let $chroSim$ be a function that compares them in the CC and returns a similarity rating. α is a normalisation constant that ensures the result is between 0 and 1. The CC similarity measure can be defined thus:

$$chroSim(a,b) = 1 - \alpha \min((a-b) \bmod s, (b-a) \bmod s) \qquad \text{where } \alpha = \frac{2}{s}$$

Equation 6 Similarity measure as shortest distance between pitches in the CC[1].

Because this equation can be a similarity measure in any circle space, the formula for the CF is the same, except that a and b need to be transformed into the CF before they are compared. Let the interval of the fifth be f. Recalling that s is the number of steps per octave, the fifth is calculated thus: $f = \frac{2}{3}s - 1$. The -1 is needed because the 0^{th} interval is the tonic, rather than the 1^{st}. Usually, $s = 12$ which means: $f = \frac{2}{3}12 - 1 = 7$.

Transforming a pitch into the CF involves multiplying by the fifth interval, f, and modulating by the number of intervals per octave, s. The CF similarity measure, $cofSim$, is thus:

$$cofSim(a,b) = chroSim((f\,a) \bmod s, (f\,b) \bmod s)$$

Equation 7 Similarity in CF space. The $chroSim$ function measures the shortest distance between the inputs in chromatic circle space and is explained in the previous equation.

Note duration similarity measure

The function that determines the similarity of note duration is a combination of two functions: relative distance and factorial difference. The relative distance function was developed in response to the fact that duration does not have a limit – rather than normalising the difference

[1] Note that modulo (mod) operates like a clock face, always returning a positive number ≥ 0, as opposed to similar operators in some programming language, for example the remainder operator (%) in Java which returns negatives.

between 0 and an arbitrary limit, the relative distance function is based on a ratio between the inputs.

Let a and b be the durations of the two notes being compared and let rd be the relative distance function. The variable c indicates the value of the difference, $|a-b|$, at which point the similarity measure is half. That is, when $|a-b|=c$, $rd(a,b)=0.5$. The mid-point variable, c, thus controls at which level the measure will be most sensitive. In *LEMorpheus*, $c=1$. The relative distance function for duration is thus:

$$rd(a,b) = \frac{\frac{1}{c}\,|a-b|}{\frac{1}{c}\,|a-b|+1}$$

Equation 8 **Function that finds the relative distance between two durations, without any upper limit imposed. It is centralized on c.**

Another component of the duration similarity measure is the ‘common factor’ difference function. The purpose of this function is to determine how closely note durations are related by a common factor. For example, the duration of short notes that are swung may be factors of long notes that are swung, for example, 0.33, 0.66, and 1.33. In this case, a duration of 1.25 should bear less similarity to 0.66 than 1.33, even though it is closer. To measure the common factor distance, the remainder from dividing the higher number by the lower number can be taken, centred on 0, made absolute and then normalised to be between 0 and 1:

$$fd(a,b) = 2\left|\frac{\max(a,b) \bmod \min(a,b)}{\max(a,b)} - \frac{1}{2}\right|$$

Equation 9 **A measure of the common factor difference between two durations.**

The duration similarity measure, let it be $durSim$, is a combination of the relative distance function, rd, and the factorial difference function, fd. As the distance between the two durations has much greater impact on their being perceived as similar compared to the factorial difference, the latter is weighted with less influence in the combination. It currently only contributes $\frac{1}{5}$ to the duration similarity function. This weighting that was determined though informal trial and error analysis of test-cases:

$$durSim(a,b) = 1 - \left(\frac{4}{5}\,rd(a,b) + \frac{1}{5}\,fd(a,b)\right)$$

Equation 10 Duration similarity is the inverse combination two separate distance measures: relative distance and factor distance (described in previous two equations). The latter is weighted less due to it being a less important measure of similarity.

Onset similarity measure

The onset similarity function is a weighted combination of various inverse circle distances, each representing a different metre of loop length. Let cd be a circle distance function, (similar to Equation 6), that takes three arguments: the two onset values to be compared, a and b, and the length of the circle space within which they are to be compared, s. The onset values are decimals with a range from 0 to the length of the loop. We have:

$$cd(a,b,s) = \frac{2}{s}\min((a-b) \bmod s, (b-a) \bmod s)$$

Equation 11 Circle distance function. a and b are two values being compared, and s is the length of the circle space they are being compared within.

This circle distance function is then applied to find the distance from a combination of circle spaces within the bar. Let y contain five different user-defined weights, one for each circle space length of 8, 4, 3, 2 and 1 (in beats). With the normalisation constant α to keep the result within the range of 0 to 1, the onset similarity function, $onsetSim$, is thus:

$$onsetSim(a,b) = 1 - \alpha\left(y_0\,cd(a,b,8) + y_1\,cd(a,b,4) + y_2\,cd(a,b,3) + y_3\,cd(a,b,2) + y_4\,cd(a,b,1)\right)$$

Equation 12 The onset similarity function is the inverse of a weighted combination of distance in five loop spaces of different lengths.

This technique allows stylistic notions of rhythmic similarity to be specified by the user. For example, within a straight 4/4 musical style of morph, it might make more sense to increase y_0 and y_1 above the others, as they relate to lengths of 8 and 4 respectively. y_4 should be weighted heavier when similarity on the micro-level needs to be exaggerated, for example, between 0.33 and 0.25. For the 3 over 4 rhythmic influence, y_2, the weight favouring the loop-space of 3 beats can be used. For future work it would make sense to also include a weight for a 6 beat circle space.

Polyphony

When the similarity measurement function is given note-groups with more than a single note, each note within one group is compared to each note within the other and the comparison with maximal (the highest) similarity is the one that is used. Let A and B be two note-groups (sets of notes with the same onset) being compared, with the number of items in each set being larger than 1. Let the function $s()$ return the similarity between two note-groups and let $s_n()$ return the similarity between two individual notes, as per the 'combined similarity measure' discussed above (Equation 4). Polyphony is dealt with thus:

$$s(A,B) = \max_{i,j} s_n(A_i, B_j) \qquad \text{where } 0 \leq i < |A| \text{ and } 0 \leq j < |B|$$

Equation 13 The similarity measure $s()$ for two polyphonic note-groups A and B. $s_n()$ is the similarity measure for individual notes within the note groups.

This is not an optimal measure – if two note-groups each have a note that is maximally similar to the other, then removing or adding other notes will have no effect on the similarity rating between the note-groups. Some ideas for improvements to the approach that would be trivial to implement are mentioned in section 6.4.

Realtime implementation

The note selection process outlined above occurs every play cycle, so as to allow the system to be responsive to changes in realtime. If the inter-onset between the most recently played note-group and the time at the current play cycle is not equal to the inter-onset between selected note-group and the note-group previous to it in sequence then nothing is played. If nothing is played and the gap between the current time and the recent note onset is larger than the largest inter-onset interval in the selected source or target, W, then it will be impossible to select a note for playback. This is called stream loss, whereby the default fallback of weighted selection is used.

In order to control the occurrence of stream loss, I have included a user-defined weight called 'anticipation', which ranges from 0.25 to 1.5. When checking for stream loss, the anticipation variable is multiplied against the value of the largest inter-onset interval in W to make it smaller or larger than it really is. If anticipation is smaller than 1, stream loss may be 'forced' when it otherwise need not occur. Another alternative would be to predict the next note only once – after a note has been created, rather than each frame, and keep this selection until the appropriate point to play it is reached. This would ensure that weighted selection is never used as a fall back, however it would be less responsive in realtime, particularly when very long notes are used.

The time complexity of this algorithm is $O(n)$ for monophonic music. It is also $O(n)$ in polyphonic music if the number of voices is limited. This enables realtime adaptability to changes in source and target music. The complexity was determined thus: if n is the number of note-groups in the selected sequence, p is the number of layers of polyphony in each note-group and d is the Markov depth, in each play cycle there is: $O(n\,p^2\,d)$. Since d is limited to below 12 and p is most often no larger than 10, they can be considered constants, thus leaving n.

6.2 Informal analysis

6.2.1 Weighted selection

The informal analysis of weighted selection was conducted by creating numerous musical examples and reflecting on the impact that the weighted selection algorithm had on how they sounded. To recap, weighted selection operates by selecting between either source or target, weighted on the morph index, at each play cycle. Two examples of weighted selection morphing between drum patterns are included within this section.

Unlike parametric interpolation, the weighted selection deals particularly well with rhythmic and polyphonic parts, rather than monophonic and melodic parts. Like all morphing algorithms it works better with source and target patterns that are already similar, however similar note onsets in particular are useful for providing consistency. In the current implementation, because the notes are being copied as a whole group of notes within the 0.25 beat span, rather than being generated as a single note, the possible onset resolution is much smaller than the play cycle resolution. That is, many notes within one 0.25 span can be sent to the scheduler and thus more complex rhythmic possibilities emerge.

A demonstration of the ability of weighted selection to deal with rhythmic and polyphonic parts can be shown by morphing between two drum patterns, as in the following example (~6.1), transcribed below (Figure 3).

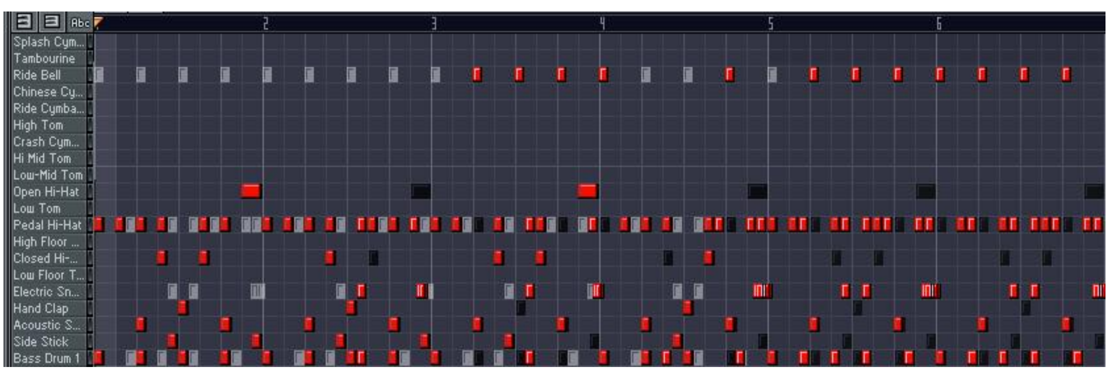


Figure 3 **Morphing, through weighted selection, between two different drum rhythms of the same length. The audible notes are in red, while the semi-transparent black and white notes underneath are provided as a reference to the source and target respectively. The first bar is the source and the last bar is the target.**

In all the morphing algorithms, swing is handled easily because it is treated as a meta-parameter that can be linearly interpolated. A technical problem of the current implementation is that for swing to occur, quantise must also occur which means that the target of the previous example would not have been able to represent the snare roll on beat 4.75 (counting from 1). The following example (Figure 4) is similar to the previous, except the source is hard swung and the target is not (~6.2). In terms of swing, this transition appears to be handled adequately.

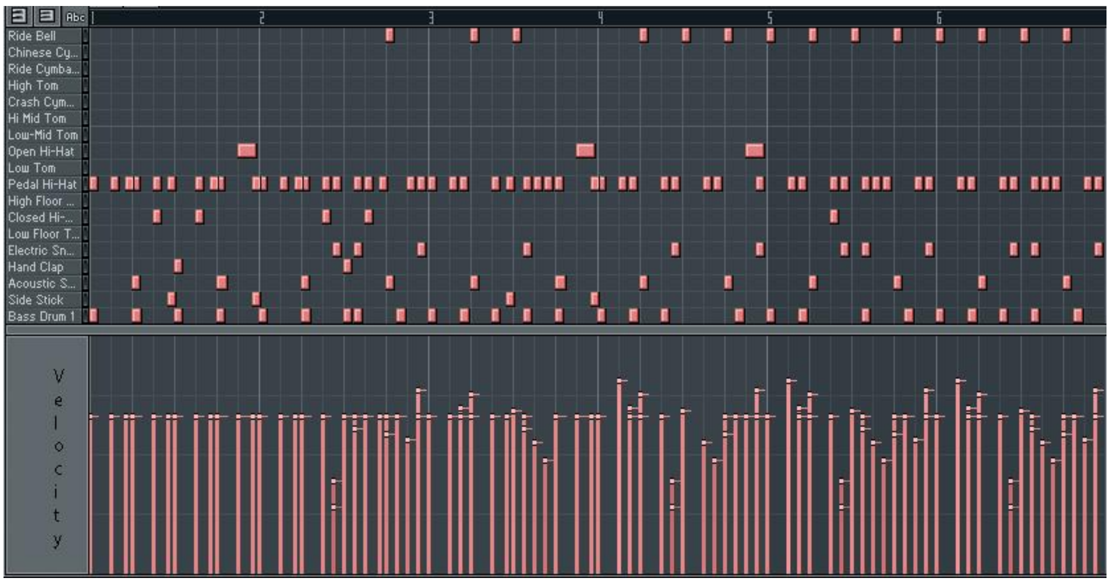


Figure 4 **Swing can be dealt with easily, but different dynamic levels make the random nature of the weighted selection more obvious.**

A major problem with weighted selection is that any similar sections within the source and target that are not within proximity are overlooked as pivot points. That is, even when sections that might be spliced together well exist, if they are not in the appropriate location within the bar, they will never be spliced together. Another major problem with weighted selection is that the resulting output tends to be randomly spliced at the frame resolution, rather than the length of phrases as would be more musically appropriate. This becomes especially obvious when the source and target have substantial differences in their pitch, dynamic and duration content. For example, the rendering in Figure 6 shows how the dynamics can jump unexpectedly from high to low or visa-versa. These two problems, pivot points and phrase boundaries, were addressed by the *Markov Morph* and *TraSe* algorithms respectively.

6.2.2 Demonstrating Markov Morph through variation

For this section, the effect of the different parameters described above will be demonstrated through a number of variations (as in, '*variation* on a theme'). While not the focus of the research, variation is an effective technique for examining the properties of the algorithm because it is essentially the 'easiest' kind of morphing – from a source to itself.

To recap on the *Markov Morph* process, the note-groups are probabilistically selected from within a note-group sequence (source or target through weighted selection), favouring note-groups where the preceding note-group (in the sequence) is similar to the most recent note-group output. This approach enables interesting connections to be drawn between sections of the source and target that are similar. The weighting of the similarity measure enables different styles of generation to occur, including: rhythmic, melodic, fifths-oriented melody or chromatically-oriented melody. Despite these gains, the current implementation has a range of limitations and problems, including illogical handling of polyphony, the disregard of pitch interval data and discontinuity when inter-onsets in source and target are very different. These problems are overcome in the *TraSe (Transform Select)* morphing algorithm that is discussed in the next chapter.

Controlling the level of variation

The *Markov Morph* algorithm includes a user-defined parameter, 'contrast', which can exaggerate the level of contrast within the probability distribution. The musical effect of this is to control the amount of overlap and randomness that is possible. When it is high, many different notes could be mistaken for the seed (like a composer with blurry vision) and thus used to generate the subsequent note, whereas, when it is low, only the notes that match the seed very specifically will be recognised. For musical variations, it essentially controls the extent of deviation the variation has from the original. In the following example (Figure 5), the contrast is decreased during bars 3 to 8 and increased over bars 23 to 28 (~6.5).

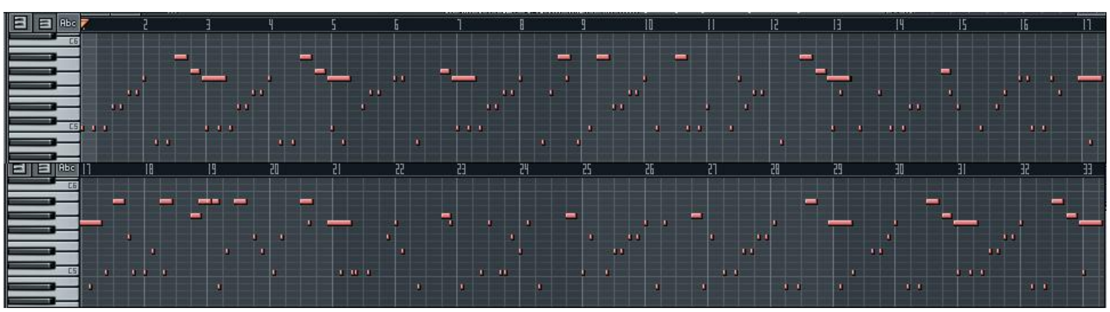


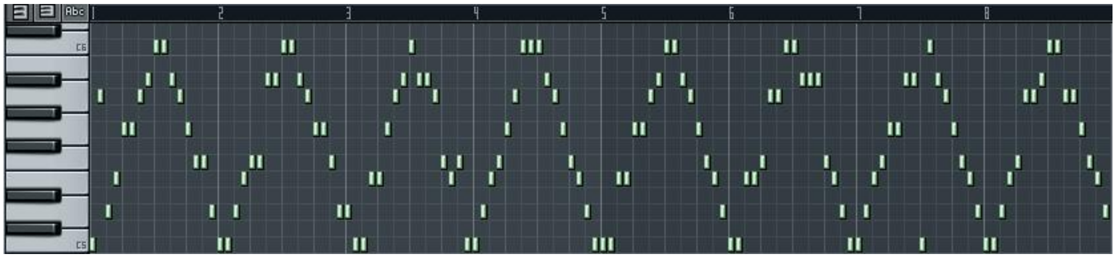
Figure 5 **Example of a variation of a two bar bassline of cheesy nineties dance (first two bars). The top piano roll contains bars one through sixteen and the bottom piano roll continues with bars seventeen to thirty-two. The contrast is decreased from bars three to seven and it remains low until it is increased again over bars twenty-four to twenty-eight.**

This example also shows that notes are only allowed to be played if the inter-onset with the note preceding it matches the time since the last note was played. By bar 27, we can see that the original pattern had emerged but was behind the beat by 0.25. This has occurred because the contrast has been increased far enough to strongly favour notes that fit the original pattern, but the onset of the note is relative to the previous note, which in this case happened to be 0.25 behind the beat. At this point the contrast has not been increased as far as to force weighted selection through **stream loss** and the onset intervals at that point are fairly short, further reducing the opportunity for **stream loss**, which would have shifted the pattern back onto the beat.

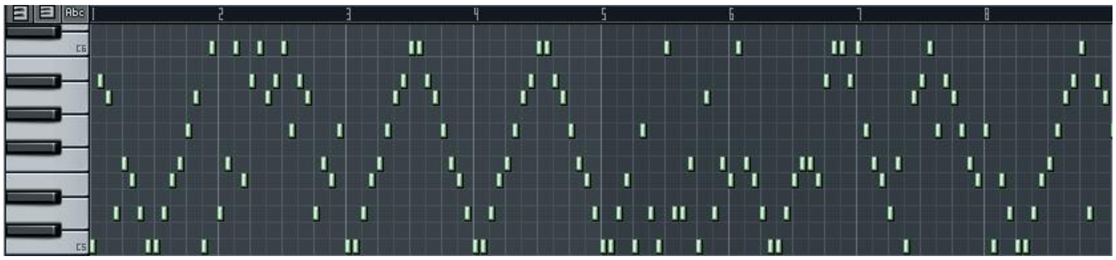
Comparing the Circle of Fifths and the Circle of Chroma

The differences between using the CF rather than the CC for similarity of pitch becomes much clearer when applied to simple scales rather than any particular melody. With a diatonic scale, the CF space will, on average, yield a greater variety of note choices than CC, indicated by a higher average value for the sum of the similarity matrix at each frame (compare Appendix C Figures C-5 and C-6). With a short chromatic run that does not have the entire set of chromatic pitch classes (for example, only seven pitch classes, as with the diatonic scale), the opposite is true. With a chromatic scale, containing all of the pitch-classes, this value will be fairly similar for both CF and CC.

The following two examples (Figure 6) are variations on the major scale with pitch weighted maximally, applying the CC (~6.6) and the CF (~6.7) respectively. The contrast for the CF example is increased so that the two examples have a similar overall rate of variation, making the harmonic differences more audible. The moving average shown in the print-out in Appendix C Figures C-7 and C-8 verifies that there is only an average difference in overall similarity of 0.07, confirming that the two have a comparable rate of variation.



A.



B.

Figure 6 **Transcripts of two different variations on a simple major scale run, up and down. “A” uses only the CC for the similarity measure and “B” uses only the CF. The depth for both is 5. For “B”, the contrast was increased so that the two could have a comparable rate of variation, as the pitches of diatonic scales are more closely related in the CF than the pitches of chromatic scales.**

Multiple perfect matches through onset similarity

On some settings, the similarity matrix contains multiple ‘perfect’ matches, that is, maximum similarity with seed. In this case, even the maximum setting of contrast will not be able to provide a completely accurate reproduction of the source because these values cannot be ‘squashed’ any further and multiple options for note generation will remain. This can be seen when using any similarity measure that judges two different notes to be equally similar.

For example, if onset similarity is weighted maximally; the modulus 8 beat space component of onset similarity is weighted maximally; the length of the loop is 16 beats; and some of the notes in the first two bars are on the same onset (relative to the 8 beat space) as the notes in the second two bars, then these notes will have equal weighting and thus both be candidates. If the contrast is maximal, this would be equivalent to weighted selection between the first two bars and last two bars, as demonstrated in the following example. The musical example of *Take On Me* by *a-ha* was chosen because the onset pattern in the second two bars is exactly the same as the onset pattern in the first two. The Markov order has no effect because the note onsets in the modulus 8 space are exactly the same for both sections. The first four bars contain the original pattern. After this point, only the modulus 8 beat space similarity measure was used and it is clear that the pitches from the first two bars of the original loop are mixed with the pitches of the last two bars (~6.8).

Figure 7 **Variation of a four bar loop from *Take On Me* by *a-ha*, from bar five and onwards. Due to using only modulus eight beat space as the similarity measure, the notes from bars three and four of the original loop are measured as a perfect match to the notes in bars one and two. The notes highlighted in green are the actual notes played. For reference, the notes in black are the notes of the second two bars and the notes in white are the notes of the first two bars of the loop.**

This is equivalent to a weighted selection between the first and second halves of the sequence. For confirmation, see Figure C-1 in Appendix C which is a snippet of the print out that shows the similarity matrix created at each quarter beat.

Modulus three and four beat onset space

A similar effect is can be created, with a notably different musical result, by using modulus 4 beat space and modulus 3 beat space together. In this case, with a Markov depth of 1, some of the time there will only be one single match found (the original note will always provide a perfect match, no matter what space it is in) and at other times, two onsets coexist within the macroperiod of modulus 3 and 4 space. The example used here is the *a-ha* sequence of the previous example, with a length of 16 beats, however, if the length of the original loop was longer than 24, there would be more than two such sections and thus more than two matches possible. The contrast remains high in this example (~6.9).

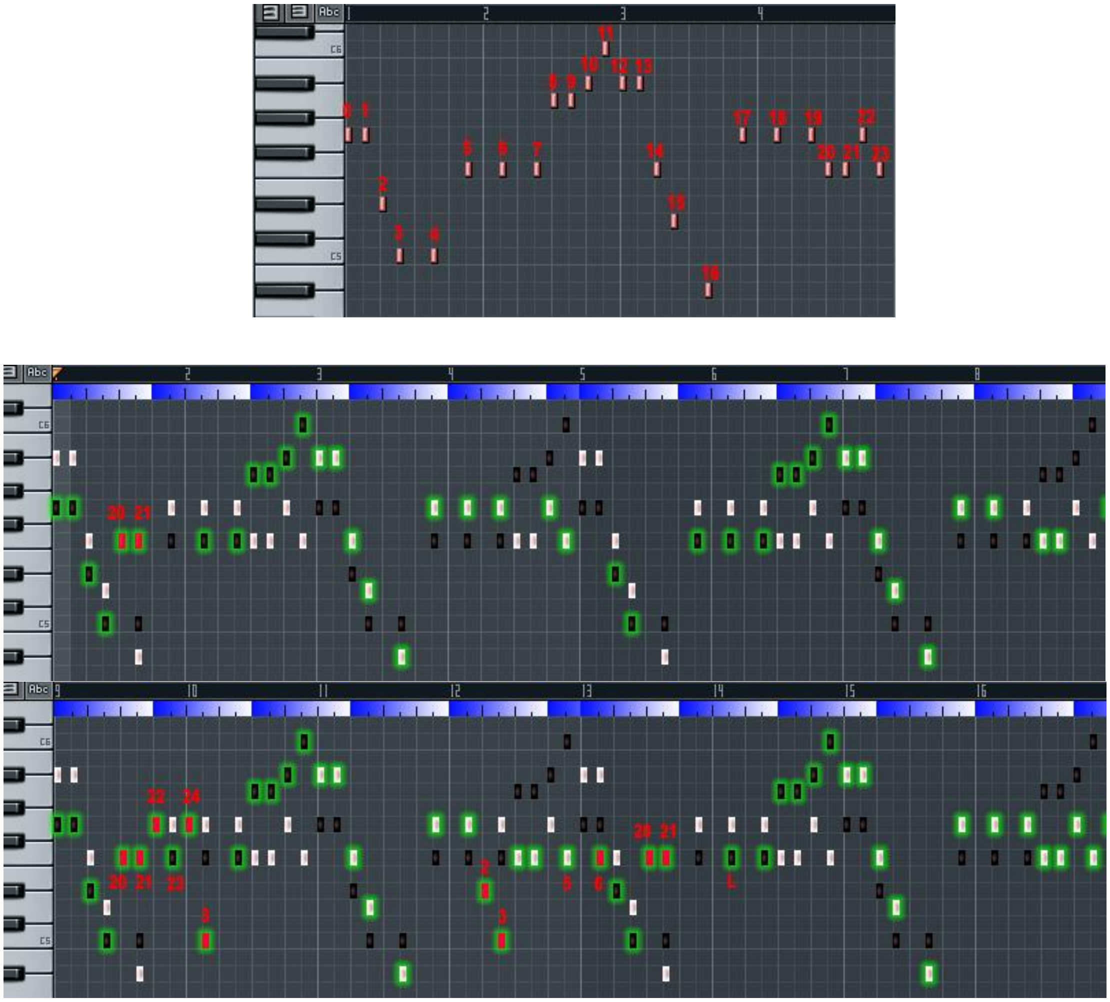


Figure 8 **Top is a numbered transcript of the *Take On Me* loop that can be used as a reference for the occurrence of the original notes in the generated material. In the two panels beneath this is a variation generated from the *Take On Me* loop of the previous example, using modulus three and four beat space for onset similarity. The actual notes played are highlighted in green. For reference, the notes from the first two bars of the original are overlaid in black while the notes from the second two bars of the original are overlaid in white. New notes that do not appear in either are red. Numbers are written in above or below notes refer to the position in the note-list (counting from zero) of the original sequence. "L" indicates a note generated through stream loss. The blue band at the top can be used as a reference to modulus three beat space.**

In the diagram, it is possible to see in bar 1 and 9 that many of the notes are generated through combined similarity in modulus 3 and 4 beat space (they are annotated with numbers). Effectively, notes that occur where the 3 beat space is synchronous with 4 beat space (bar 1 and

4) are randomly substituted. See Figure C-2 in Appendix C for an example of the print out of similarity matrices generated for each quarter beat.

Many multiple perfect matches through pitch similarity

In the previous two examples, the onset similarity is weighted maximally and this tends to favour rhythmic coherence or, on some weightings of modulus 1, 2, 3, 4, or 8 beat space, rhythmic complexity, within the music generated. For more of an 'improvised melody' effect in the music, the pitch similarity can be weighted maximally as is the case in the following example (~6.10 and Figure 9). Using the same *a-ha* tune, a Markov order of 1 , maximal contrast and weighting linear pitch space highly, the musical difference is immediately noticeable. The patterns that emerge deviate much more from the original, because there are many more choices available at each quarter beat.

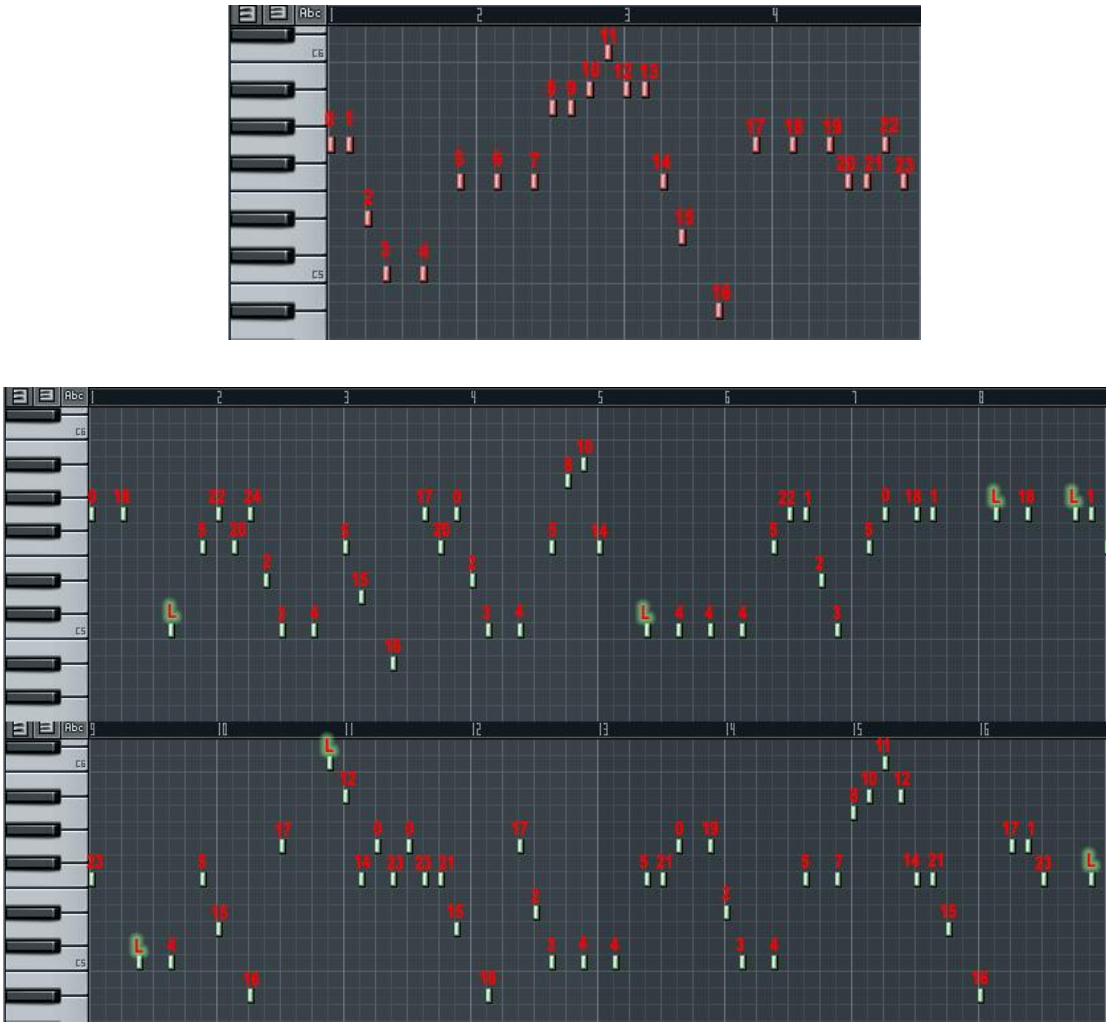


Figure 9 **Example of variation of *Take On Me* by *a-ha* where linear pitch similarity is weighted maximally, Markov order is one and contrast is maximum. To assist analysis, the number of the original note, from which the generated note is copied, is notated in red above each note. If a note is generated through weighted selection due to stream loss, an "L" is used instead. For reference, the numbers of each original note is included above.**

In this example, because the contrast was decreased to a moderate level, the generative capacity of the algorithm is significantly higher, in that there are many options at any particular time to choose from. This is evident from the accompanying print out in the Appendix C, Figure C-3.

This is essentially an example of the well known and commonly used technique of generating note pitches using Markov chains, except that it is in realtime and the current implementation allows **stream loss** to occur, rather than making a single decision at each step of note generation. While some of the phrases that are generated sound believable, there is not much coherence in their placement in relation to each other or within a larger structure. For example, bar 2 sounds like it could be a worthwhile phrase by itself, but would be more suitable at the end of bar 4 and the beginning of bar 1 where most of the notes originated.

Stylistic constraint

Constraints that favour placement of phrases in some way more similar to the original can be introduced by increasing the weighting of onset similarity, for example, to 0.6, while keeping pitch similarity high at 1.0. In the following example (~6.11 and Figure 10), the modulus 1, 2 and 3 beat spaces were weighted high in order to reduce the similarity of notes that are close to each other, while still favouring notes with onsets that fit the original pattern. The CF similarity was weighted maximally, order was 1, and contrast was moderate.

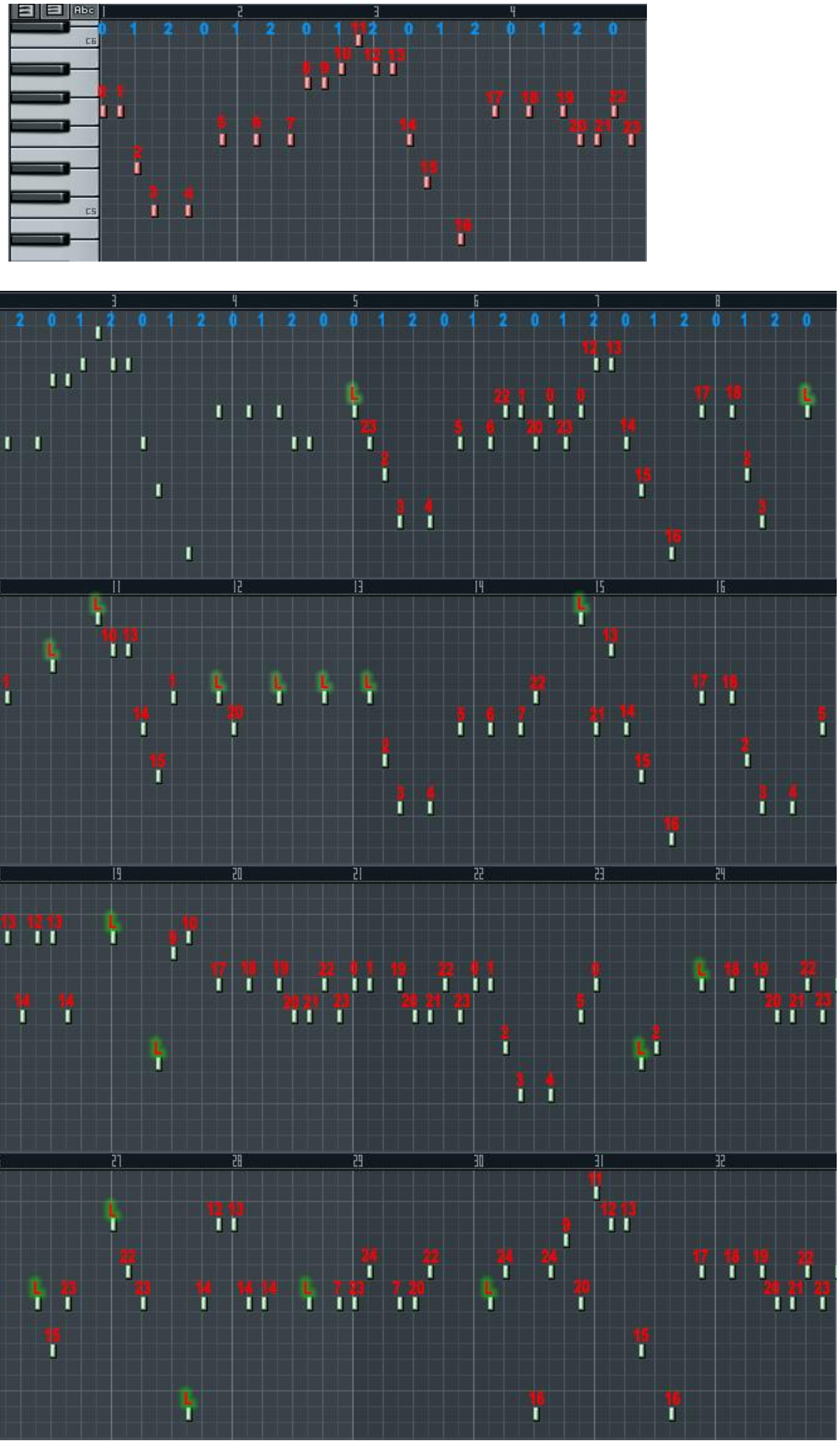


Figure 10 **The *Take On Me* variation using pitch similarity at 1.0; onset similarity at 0.66; Markov order 1; moderate contrast; inter-onset space of modulus 1, 2, 3 and 4; and CF pitch space. The cycle of 3 is included in blue numbering along the top for**

reference, as is the original pattern (red notes, top). The numbers indicate the original note. The L indicates stream loss.

Although, the structure remains random, it is clear that the timing of the original pattern tends to occur more strongly. Some of the generated patterns, even ones that are only vaguely similar to the original, tend to work well harmonically, for example, bar 5, 9, 18, 19, 30 and 31. This can be attributed to the transformational effect of the CF space used as a similarity measure because, in these examples, while the note onset similarity is not particularly high, the pitch similarity in the CF is. Repetition of short phrases does not always work but can sometimes be reminiscent of the free structures of improvisation, for example, 16, 20, 21 and 29. This is can be attributed to the transformational effect of the combination of modulus 1, 2, 3 and 4 beat space for onset similarity, as is evident in the print-out in Appendix C, Figure C-4.

The similarity weights are user controlled parameters that influence the compositional processes within the algorithm, and their moderate success in the *Markov Morph* partially inspired the development of the *TraSe (Transform-Select)* algorithm (Chapter Seven). In the *TraSe* algorithm, such parameters play a more central role, with a great many compositional techniques being controlled through weights.

A problem with decreased contrast and more options for note selection is that **stream loss** tends to occur more often. This is because a variety of values for inter-onset to the previous note will exist and only one of these inter-onsets will fit the current time interval since the last note was generated. This is not a problem with the technique, rather the implementation, which has not been refined due to the focus of development having shifted to the different approach of the *TraSe* algorithm (Chapter Seven).

6.2.3 When Johnny comes morphing home

Although application of the *Markov Morph* to variation simplified the context and highlighted interesting properties of the algorithm, it was also necessary to test it according to how well it morphed (this being the ultimate aim). Morphing between the *British Grenadiers* and *When Johnny Comes Marching Home* (Mathews and Rosler 1969), the performance remained fairly poor, but arguably better than weighted selection or linear interpolation. For this example (~6.12 and Figure 11) Pitch is weighted 1.0; onset weighted 0.49; Markov order 2; contrast is moderately high; anticipation is at 0.28; modulus 8, 4, 3, 2 and 1 onset spaces all weighted 1.0; linear pitch weighted 0.08; CF pitch-space weighted 1.0.

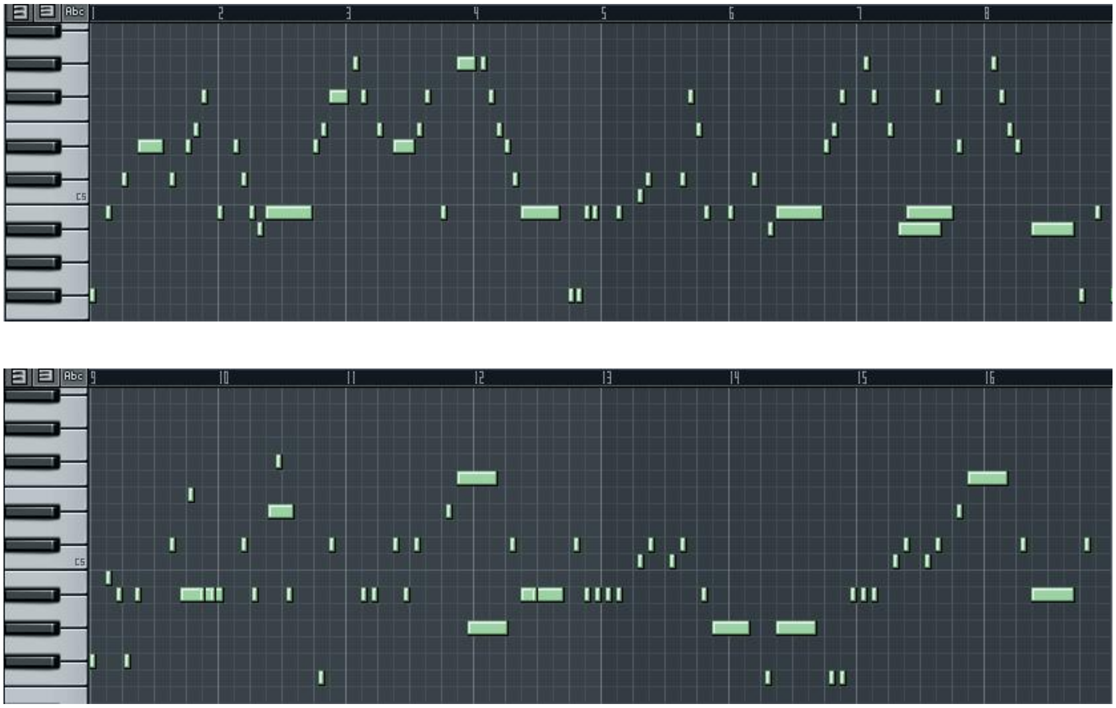


Figure 11 **Transcript of a *Markov Morph* between *The British Grenadiers* and *When Johnny Comes Marching Home*. The source and target occupy the first four and the last four bars respectively.**

The primary problems facing the *Markov Morph* are **stream loss** and small sample space. Stream loss has been only a minor problem in most of the examples shown above, because the source and target are the same – weighted selection can only serve to bring the music back more closely to the original. As well as this, most of the music has had a small set of possible note inter-onsets. However, when morphing between a source and target with a very different set of inter-onsets (Figure 11) it is clear that the problem becomes much more frequent and interferes substantially with the musical outcome. In this example, the margin for stream loss occurrence was reduced through the 'anticipation' variable (see Realtime implementation in section 6.1.3, above), thus inducing a more frequent occurrence of stream loss in order to maintain continuity within the music. If this was not done, there would be many more discontinuous breaks in the melody which sound awkward. From the print out, the average stream loss rate over the entire morph was extremely high, at 0.416. This meant that many of the notes chosen resulted from weighted selection rather than *Markov Morph*ing. This is an implementation problem which could be easily overcome in future work by making a single selection for the next note whenever the current note has just been generated, instead of making the selection every frame.

In statistical terms, the sample space of two short note sequences is too small to be useful. This has been partially overcome by incorporating similarity measures which allow fairly high depths while still retaining a diverse probability distribution. However, there appears to be a fundamental limit to the amount of coherent material that can be generated without incorporating some kind of external source of musical stylistic information. Some preliminary trials using long pieces of classical music with *Markov Morph* were attempted, but the time and memory limitations with creating a huge note probability distribution on each play-cycle made this intractable with the current algorithm. Another approach would be to use a database and pre-render the probability distributions, however, this direction would shift the focus of research further from morphing as a compositional technique and put the emphasis too heavily on machine learning. Rather than improving the *Markov Morph* algorithm, the *TraSe* approach was developed instead as it held the promise of being able to more easily and explicitly incorporate music composition knowledge into the system and seemed like a more realistic model of compositional processes related to morphing.

6.3 Formal evaluation

Two formal evaluations of the *Markov Morph* took place: a focus group and a ‘focus concert’. The focus group included a prototypical questionnaire and was designed to explore how the morphing systems might best be evaluated, rather than to generate valid data. The focus concert was aimed at assessing the *Markov Morph* in the realistic context of a concert while applying a focus-group style questionnaire. Live morphing using the *Markov Morph* algorithm was compared to the live mixing of a professional DJ, who used the same source and target material. The emphasis on live delivery was to provide a degree of realism to the focus concert.

6.3.1 Focus group

The aim of the focus group was to obtain methodological criticism regarding the focus group process itself and the questionnaire used and to a lesser extent, musicological feedback on the transitions generated by morphing algorithms. **Smoothness**, **coherence** and ‘**danceability**’ were posed as quantitative attributes of the morph, while **mood** required qualitative feedback. While a range of useful improvements to the methods of designing and executing the questionnaire were obtained, one of the sections was invalidated due to technical errors and as for the others, only some fairly obvious musicological evidence was generated.

Method

I will now explain the method of the focus group evaluation which included: the algorithms and algorithm settings that were tested, the source and target music, the content of the questionnaire and the procedure that was enacted to obtain information from the participants.

Four different algorithms and parameter configurations were tested:

1. Cross-fade
2. Weighted selection
3. *Markov Morph* favouring pitch and with Markov order of two
4. *Markov Morph* favouring pitch and onset evenly with depth of four and low contrast

For the pitch similarity measure, the CC was used because the CF had not been implemented at that stage. The cross-fade was included as a benchmark to represent the 'industry-standard' approach currently being applied in the context of computer games or DJ mixing. From the informal tests it was clear that weighted selection would perform poorly. Despite this, it was included so that comparisons could be made between it and the *Markov Morph*, which, due to **stream loss**, would fall back on weighted selection some of the time.

The original plan was to also include *Markov Morph* examples with:

1. Duration weighted maximally and Markov order of two
2. Pitch, duration and onset weighted evenly and a Markov order of two
3. Pitch, duration and onset weighted equally, a low contrast and Markov order of four.

Unfortunately, technical problems occurred while running the test so that duration could not be weighted without incurring **null predictions**. As a result, the settings were modified after the problem had been discovered, so as to avoid duration. The primary reason for using the different Markov settings was to confirm my own subjective opinion that using a high order and low contrast was more musically appealing than high contrast and low order.

Three different sets of source and target music were created, each within a different 'style': electro, psychedelic trance and drum'n'bass. The music was based on the tracks *Emerge* by Fisherspooner (Dave Clark remix) and *Flatbeat* by Mr Oizo for electro; *Planet Dust* by Bad Company and *Decoy* by Stakka and Skynet for drum'n'bass. It was more difficult to find trance tracks in which note-level composition rather than spectromorphology was the dominant feature. The final examples created for trance were loosely based on *Wunderbaum* by Highpersonic Whomen, *Computers & Microprocessors* by Logic Bomb and *Solar* by Miraculix. All of these tracks were selected from a pool of thirty for each genre of electro, drum'n'bass and trance (ninety in total) and were chosen by scoring each according to how indicative the music was of my own subjective notion of the genre, and how easily it might be replicated. The main sections of the songs were replicated, with a little elaboration, using the *LEMorpheus* MIDI sequencer, driving the *Reason*™ synthesiser.

Throughout the focus group each morph occurred from two different loops, both belonging to the same musical style. Each algorithm and/or parameter configuration was used to create a different morph in each of the three styles. This meant that because there were four different algorithms/settings and three different styles, twelve examples were played overall.

The questionnaire sheet was designed with a page for each morphing algorithm/configuration, with three sections per page for each of the three styles. Each section had the same format of four questions regarding: smoothness, coherence, danceability and mood. Smoothness was defined as moment to moment continuity. Coherence was defined as the degree to which the music sounded as though it was effectively communicating what the composer intended. Danceability was defined as how easy and appealing the music would be to dance to. Mood was defined as the subjective aesthetic experience induced in the listener by the music. The smoothness, coherence and danceability were considered as quantitative criteria which required a rating from 1 to 10, but additional space was left for comments. The mood was considered as qualitative and a space for descriptive terms was provided for each source, morph and target. At the bottom of each page was space for overall comments on the algorithm.

It was envisaged that, as a practice study, the quantitative criteria could have statistical techniques applied to discover trends from example to example. The qualitative question was included to confirm that the source and target were musically realistic in that they were able to express some kind of mood, and also to examine if the mood projected during the morph was a hybrid of the source and target or if it were something else more common to all the examples regardless of source and target mood and style.

There were six participants in attendance which is not a statistically significant sample size. After signing the consent form, the goal and structure of the focus group was introduced and the ground rules were made clear. The meaning of the criteria was discussed and refreshments were made available. Notes were taken throughout and the event which was recorded onto video-tape. The structure as follows was repeated for each of the algorithms/parameter configurations:

1. Three examples with different source and target material were played.

2. The group listened to each example.

3. Each participant answered the questions relating to the morph.

4. After all examples were played, the group discussed the musical properties of the algorithm overall.

Feedback on the questions and criteria were noted during the discussions, and more extensive feedback on this was had at the end of the session.

The ground rules of the focus group (McNamara 2007) were to keep focused, sustain momentum and obtain closure for each question.

Results

The most useful and reliable results from this study was feedback and ideas for improving the definitions, quality of data and methods. As far as the actual data is concerned, the most striking feature was the amount of variation from person to person, from algorithm to algorithm and from style to style. The results were inconclusive.

Overall, the smoothest, most coherent and most danceable style was perceived to be drum'n'bass, and the most popular morphing algorithm was the *Markov Morph* on pitch and onset with high-order and low distinction. However, this slight trend may be entirely due to a methodological problem that is described further below. The criteria seemed to be closely correlated in that for each example marked, the three scores were rarely different by more than two points (on a scale of one to ten). In general, electro was less liked than trance which was less liked than drum'n'bass, although, there were often exceptions. Most people gave the cross-fade and the *Markov Morph* with pitch only and depth of two a low rating on all criteria when applied to the electro example, while the trance and drum'n'bass fared moderately well.

From the discussions with the participants at the end of each algorithm/parameter configuration, it appears that different people would listen to different aspects of the music and this would define to a large extent how they perceived the transition, thus partially accounting for the large amount of variation. For example, if the melody happened to be performing quite well while the drums were erratic, some listeners would rate the example very high, without noticing or giving much weight to the drums, which were perceived by them to be peripheral. Other listeners would pay attention to the drums only and mark the example very low accordingly. This highlights the importance of participant instructions.

Numerous feedback items regarding the methods employed within the focus group study were obtained:

- Either the algorithms need to be more deterministic and reliable, or a greater number of examples for each need to be played. This controls the chance that unlikely morphs overly influence the results.

- The examples should be recorded prior to the test to ensure that no software problems occur during the test and threaten the integrity of the data. If playing live, the software needs to be tested sufficiently prior.

- Either more specific listening instructions or additional questions regarding what the participant is listening to within the music is needed. This would act as a control for participants tuning in to different parts such as the bass, melody or drums.

- Questions that confirm whether or not the source, target and morph are musically ‘realistic’ need to be clarified. Realism in this case means how likely it would be for the music to occur in a real world context such as a club or a computer game.

- The range for the quantitative criteria should be odd rather than even, so that there is a number that represents the middle.

- Other methods of judging one example against another, such as ranking, need to be investigated.

- The size of the group must be much larger if statistically significant findings are to emerge from the quantitative data.

- The order of examples should be structured randomly so the participants have no expectations that might be projected from a logically structured questionnaire. For example, the questionnaire used followed a progression from cross-fade, weighted selection, Markov depth two and Markov depth four with low distinction. The styles followed a progression from electro to trance to drum’n’bass. This ordering might explain the trend to prefer drum’n’bass.

- The problem of biased expectation could also have been reduced by leaving each example untitled.

- There was a significant drop in volume during the middle of each morph that hindered analysis. A better approach to cross-fade would be through the equal power (logarithmic) curve as used on DJ mixing desks.

- The length of the morphs was too short to judge them confidently. The test should have morphs of different lengths or be conducted online so as to allow the participants to replay the morph.

- It was suggested that a way of obtaining feedback could be to predict the effect that the algorithm is likely to have on unheard musical examples or discuss which aspects did and didn't work.

Summary of formal focus group evaluation

The primary objective of the focus group – to obtain thorough methodological feedback regarding the gathering of musicological morphing data – was clearly achieved. The secondary objective – to gather useful musicological data – was only partially achieved, due to the many problems highlighted above. Despite this, the findings did suggest that morphing was at least competitive with the industry standard of cross-fading and also that the technique of *Markov Morph*ing was promising and at least marginally more successful than the technique of weighted selection. Some of the feedback gained was directly useful in highlighting aspects of the software in need of technical improvements. Overall, the results helped clarify the directions for further development and for more robust, conclusive studies to occur, as shown below.

6.3.2 Focus concert

The objective of the focus concert was to gauge the potential of morphing divergent styles in a live Electronic Dance Music (EDM) context and to receive musicological feedback on differences between morphing and current EDM practice that could feed into future developments. The term 'focus concert' was used because it was conducted with much of the same context and structure of a live concert but with the intention and apparatus of a focus group. Participants were asked to provide written responses to live music, some of it performed by me using *LEMorpheus*, and some of it performed by a DJ using standard mixing and sampling equipment.

The results indicated that, for divergent styles and particularly for long durations, morphing is competitive and, for some individuals, preferable to DJ mixing. Particularly noticeable aesthetic features that were contrasted against standard DJ mixing were the erratic structures of probabilistic morphing and tempo interpolation.

Method

The approach of the focus concert was to play live morphs and elicit written responses from the audience regarding various aspects of the music that was heard. Both long and short morphs through both *LEMorpheus* morphing and DJ-style mixing occurred. The morphing was performed by myself and the DJ mixing was performed by an experienced professional DJ who was paid for the gig. The morphs were between three quite different styles of EDM. Questions regarding musicological aspects were broad and qualitative while other questions that were directed at gauging the morphing algorithm were quantitative. Comments and feedback regarding methodological aspects of the focus concert approach and the questionnaire content were gathered at the end, both in written form and verbally.

Three divergent tracks were selected and recreated, in an approach that was very similar to that used in the focus group described above. From a pool of thirty for each style of electro, drum'n'bass and trance (ninety in total), the tracks were chosen according to how indicative the music was of the genre and how easily it might be replicated. As well as this, particularly contrasting pieces were chosen so as to test the ability of the morphing algorithms to cross stylistic boundaries. The songs were *Flatbeat* by Mr.Oizo for electro, *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 the *LEMorpheus* sequencer driving the *Reason*[TM] synthesiser. The examples are (~6.13) for electro, (~6.14) for trance and (~6.15) for drum'n'bass.

The DJ was provided with music with which to practise and familiarise himself, two weeks before the concert date. These were seven-minute versions of the tracks that were recorded through automating the muting and un-muting of various layers within the main section. These are (~6.16) for electro, (~6.17) for trance and (~6.18) for drum'n'bass. My own form of preparation was to adjust the various parameters of the *Markov Morph* algorithm. The DJ was questioned informally, confirming that the material was of a realistic standard, albeit an odd selection of music to play, back to back.

On the day, an audience of sixteen (only eleven completed the response form) volunteers from various musical backgrounds were situated in a concert room as if to watch an event, but with pens, table-chairs and the questionnaire response form. The background of the project, the goals of the focus concert, the structure and ground rules were all explained before participant consent forms were signed. The planned structure was as follows:

1. 7:00 - 03 Get pens, read information pack, sign consent form
2. 7:03 - 05 Answer question 1: Electronic music background
3. 7:05 - 06 Read through the structure and the questions that will be asked, so as to have an idea of what to listen for during the music.
4. 7:06 -11 **Section 1**: Short transitions
    a. DJ mixes from track A to track B to track C (two short mixes)
    b. Researcher morphs from track A to track B to track C (two short morphs)
5. 7:11 - 16 Answer the questions for section 1: short transitions
6. 7:16 - 26 **Section 2**: Extended transitions
    a. DJ mixes from track A to track B to track C (two long mixes)
    b. Researcher morphs from track A to track B to track C (two long morphs)
7. 7:26 - 31 Answer the questions for section 2: extended transitions
8. 7:31 - Finish

Note that the music was labelled anonymously. In reality, the focus concert went overtime by almost half an hour. As with the focus group, the ground rules reminded the participants to stay focused, sustain momentum and obtain closure for each question.

Before listening to music, the participants filled out a number of questions pertaining to their background experience in EDM and the breadth of the participant's familiarity with EDM genres. This involved selecting which genres of music were played at the dance music events they attended, including: house, hardcore, techno, drum'n'bass, trance, disco, old school, electro, breaks, industrial, glitch-tech and underground. Room was provided to name genres that were not on this list. The location of the events was similarly gauged, including: pub, night club, friend's house and outdoor party. The participants were then asked approximately how many EDM events they attended over the last year, as well as how long it had been since they last attended such an event.

Each performance consisted of two morphs/mixes played back to back, starting with the trance track (A), proceeding to drum'n'bass (B) and then electro (C). During the first section of the focus concert, the morphs/mixes that were analysed were short, the DJ mixing within 30 seconds while

I morphed in around 10 seconds. Originally, it was planned for the DJ mixing to also be 10 seconds; however, it was subjected to the spontaneity of live performance. The *Markov Morph*ing algorithm with the CF similarity measure was used for the tonal parts, while manually specified note layering was used for the drums. After two short performances the audience participants were asked various questions. Firstly, they were asked whether or not each transition played would ever be heard at an EDM event (separate questions for morphing and mixing) and if not, why not. Following this, it was asked which technique for short transitions would be preferred to listen to at the kinds of EDM events that they attended. A vote for 'undecided' could also be given. The participant was then asked to justify the preference. They were then asked to give a rating from one to seven on how applicable the short morphs would be in a live electronic music context. Following this, they were asked to comment on the musical qualities of the DJ mixing and then the morphing software. At the end of the section there was room for any additional comments. The second section of the focus concerts was identical to the first section except that the morphs and mixes were longer and the questions referred to long transitions rather than short transitions. The transitions were 90 to 180 seconds for the morph and, again due to the unpredictable nature of the live performance, 45 seconds for the DJ mix. Some *Markov Morph* examples which were recorded on a separate occasion are included and are indicative of those that were played at the focus concert (~6.19, ~6.20, ~6.21, ~6.22, ~6.23).

Results

The focus concert has been recorded on video (~6.24). I analysed the background EDM information given by each participant and gave each one an 'EDM credibility' rating from 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 2.144. The minimum was 3 and maximum 10. This was used to weight results, as discussed further below, but without substantial effect.

When asked if they would expect to hear DJ mixing similar to that which was heard during the study at an EDM event, participants unanimously said yes.

When asked whether, at an EDM event, they might hear a short morph similar to the one played, 4 said yes, 6 said no and 1 was undecided after the short morphs.

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. When asked to comment, 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. For example:

"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."

For short transitions, no one 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 increased the length of his mixes, meaning that, for the short morphs, the DJ mixing was perceptibly longer than the *LEMorpheus* morphs, which is likely to have 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 different to those for short transitions. 4 preferring the morphing, 3 preferring the DJ mixing and 4 were undecided. When these preferences were weighted on the 'EDM credibility' of the individual participant, they did not change significantly. 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."

The comment about the sounds relates to the fact that the synthesiser creates new sounds when rendering new MIDI notes, whereas this is unable to occur through the DJ-mixing.

Short morphs were rated only moderately on applicability to real electronic music contexts, scoring a 64% average with standard deviation of 22%, while the long morphs were judged highly applicable, with a 78% rating and standard deviation of 17%. Weighting these results on EDM credibility did not change them significantly.

Discussion

The fact that all participants agreed the DJ mixes were similar to those in real EDM contexts implies that the benchmark comparison was well-founded and likely to apply to real world settings. However, because the tracks were selected according to how different they were, not how similar they were, the study purposefully ignored the ‘art of track selection’ which is part of the skill of being a DJ. From conversations afterwards it appear that the question could have been reworded in more direct language, such as “do you think the DJ demonstrates technical ability similar to the professional DJs you have heard at EDM events”. Future focus concerts could morph between songs that were selected by the DJ, however, tracks that are similar tend to be much less challenging for the morphing software to deal with and so it is unlikely that such a test would provide useful new musicological insights that improve the software. Another approach could be to allow the DJ to use their own record collection to ‘construct’ a lengthy morph, or to raise the benchmark further by employing a producer rather than a DJ.

The fact that people generally did not hear morphing-style transitions in the EDM events that they attended indicates that morphing is a practice that is alien to current methods. From comments, two factors are primarily responsible: the erratic structure produced by the unpredictable probabilistic techniques and the ramping of tempo due to interpolation. Future algorithms might use a database of musical patterns against which the structure can be fitted, or any number of improvements (see 6.4) while the option of tempo matching rather than interpolating will need to be added.

The preference the audience showed towards morphing in long transitions, as well as the applicability ratings, indicated that morphing is a viable new approach. It is particularly interesting to consider the high preference ratings even though morphing was clearly judged as being atypical to the EDM contexts the participants were familiar with, implying that there is a musical niche that may be filled. The preferences directed towards the DJ mixes for the short morphs should be tempered by fact that the DJ mixes were longer and comments that cited length as a positive factor. However, before the ‘long’ mix, the DJ verbally expressed his misgivings at attempting a difficult mix which may have suggested a kind of dominance of the morphing software in the minds of the participants.

Conclusion

In conclusion, it is clear from the results that morphing of EDM is applicable and competitive with status quo EDM mixing techniques when applied to divergent styles of music. Despite this, it results in an unfamiliar aesthetic that could undermine people’s expectations and subsequent judgements, mainly due to the interpolated tempo during transition and the aleatoric nature of the *Markov Morph*. While adding an edge of realism, the focus concert approach was difficult to control. Ideas for improvements included:

- A more controlled environment, either with better rehearsal or in a context that is not live.
- More explicit and detailed wording of instructions and explanations that seeks qualitative rather than quantitative responses.
- More time for the participants to properly word their responses.
- Putting the section on participant background at the end of the survey so they do not feel compelled to live up to their own descriptions.
- Raising the benchmark by comparing the morphs with human composed and produced transitions or other performance software such as *Abelton live!* rather than DJ mixes.
- A larger body of participants to add variety and validity to the results.

6.4 Improvements to probabilistic morphing

There are a number of possible improvements to the current algorithm, including: phrase detection, inter-onset similarity measure, note loudness/dynamic (MIDI velocity) similarity measure, multiple note similarity comparisons, metric constraints, better handling of **stream loss** and sample size.

Musical phrases that are generated during the morph are currently spliced together from play cycle length chunks of music from the source or target. Musically, there is no reason for phrases to be composed of these segments and the chance of a short musical phrase ending prematurely is higher than the chance of the phrase ending at a natural end point. This problem might be improved using phrase detection to find the start and end points of phrases and purposefully increase the chances of selecting notes from the phrase – the value of such a technique would depend on the number of phrases used in the source and target music.

Currently, the similarity measure from which the probability distribution is constructed uses a combination of pitch, onset and duration similarity. Perhaps more suitable notes might be favoured by the probability distribution if some additional measures were combined within the current similarity function, such as inter-onset and dynamic/velocity. While the current onset measure, that assesses the onset similarity relative to the various loop lengths, is useful for enforcing a kind of metre, an *inter*-onset measure would increase the continuity of phrases, regardless of their position within the loop. The inter-onset similarity function would be similar to the duration similarity function described in Equation 10. Including a similarity measure for dynamic would provide an additional dimension of detail. The appropriate function for dynamic would be a version of the linear similarity shown in Equation 5. The range would be 0 to 127, and the magnification would likely be smaller than 6, as large distances generally seem more common in velocity than in pitch.

Currently, some detail is lost when comparing polyphonic note-groups. The approach takes only the most similar notes within the note-groups and the similarity rating of these is used to represent the similarity rating between the two note-groups as a whole. This measure was used because it ensures that when two different note-groups have exactly the same notes, they will be considered to be the same. This is not the case using some other measures, for example, taking the average note similarity between all possible note comparisons. Since moving on from the *Markov Morph*, I have developed a Nearest-Neighbour distance (detailed in the following chapter) which also has the ability to more accurately judge two polyphonic note-groups, while providing an additional level of accuracy. A trivial future development that would assist the ability of *Markov Morph* to deal appropriately with polyphonic music could be to apply this Nearest Neighbour distance to note-group comparisons.

Currently, the user increases or decreases the 'contrast' variable at will, effectively to control how similar the notes need to be to the seed to be candidates for playback. Instead of the user controlling the contrast, with it remaining fixed for much of the time as a result, it should automatically increase if the selected source or target is the pattern that generated the seed and decrease if it did not generate the seed. This way, connections based on similarities between very different source and target patterns would be accentuated.

Musically, perhaps the biggest problem with the *Markov Morph* is the sense of randomness in the interjection of notes. Therefore if some kind of grid was applied that constrained notes to start times that are coherent within a specified metre, this problem might be alleviated somewhat.

As mentioned above, the case of **stream loss** occurs when the largest inter-onset interval in the selected source or target is smaller than the interval between the previously generated note and the current frame. The current strategy for **stream loss** is to revert to weighted selection. Perhaps a better strategy would be to predict a note only once each time – after a note has been created, rather than doing it again each frame. This would ensure that a note which has been selected according to the probability distribution is used; however, it would be less responsive to realtime changes. The realtime responsiveness could be maintained if combined with a 'listener' that polls for significant change, triggering a re-prediction when they occur.

A quite severe problem is that the amount of musical information provided to the algorithm is usually not enough to build a working probabilistic model with a depth/order greater than two, depending on the amount of repetition in the material. Statistical techniques such as Markov chains are more commonly performed on large databases. This problem has been reduced by the use of a continuous similarity measure, however, it remains significant. To overcome it more effectively, a database of music could be incorporated from which a more comprehensive model is built. *The Continuator* provides a good example of how this could be implemented efficiently (Pachet 2004). The reliance on music of different styles could be weighted differently by the user.

Fundamentally, the Markov technique, as currently implemented, reflects only the musical surface and is fundamentally different to my own compositional practice, those of other composer/producers I know, and to compositional techniques described in music textbooks. Because of this, rather than developing the increasingly complex layers mentioned above, it seemed more fruitful to first attempt a more convincing morph that was able to directly incorporate compositional techniques without the restrictions of realtime operation. It was also noted that no evolutionary approaches to morphing had been attempted before. If this processor-intensive technique proved successful, realtime adaptation could be explored later. This new approach was the *TraSe* morphing algorithm, which is detailed in the following chapter.

6.5 Summary of the probabilistic morphing algorithm

To summarise, a probabilistic morphing algorithm has been developed that operates through the weighted selection of source and target which is used to generate a probability distribution for note generation conditioned on previous output as a seed. Each play cycle, the probability distribution is used to generate a note and if the inter-onset distance between the most recent output and the currently generated note is correct, the note is played. The length of the seed and Markov depth is determined by the user. The probability distribution is generated by comparing the similarity of the seed with each segment of the same length in the selected source or target note-group sequence.

The *Markov Morph* uses the similarity measurements to obtain continuous probabilities, thus compensating for the statistically insignificant number of note-groups in the source and target. The similarity between note-groups is derived from a weighted combination of similarities of pitch, duration and onset. The pitch similarity is a weighted combination of distance in linear pitch space, the CC and the CF.

Informal evaluation of the *Markov Morph* demonstrated how the level of variation and various stylistic aspect of the music can be controlled by the user. The first formal focus group evaluation provided a number of ideas and refinements for future tests. The second formal 'focus concert' evaluation benchmarked the software against a professional DJ in a live performance context. The *Markov Morph* was competitive, particularly with long durations between source and target, possessing a somewhat erratic and unusual style.

Future research in probabilistic morphing could incorporate a number of refinements to the current *Markov Morph*. This includes a more musical approach to phrase lengths, rather than using the play cycle length. More similarity measures such as inter-onset distance and dynamic are needed. Polyphonic note-groups could be compared more accurately using a Nearest Neighbour similarity measure. Automatic adjustment of the contrast of similarity measurements is needed to obtain a consistent level of variation in the probability distribution. Additional constraints of music representation, particularly for rhythm, could help to reduce the erratic nature of the music. **Stream loss** could be avoided if the notes were predicted only one at a time, rather than each play cycle.

Rather than implementing and testing these potential improvements, however, an even more novel approach was envisioned, involving evolutionary processes. After some initial scoping, this approach was found to be practical and the remaining research effort was directed towards it, as explained in the following chapter.