8 Conclusion

In conclusion, the primary research objectives have been fulfilled. Interactive compositional morphing has been investigated as a new approach to enhance adaptivity. New aesthetic possibilities enabled by compositional morphing have been explored. I will end with a summary of the research that has been completed (8.1), some demonstrations of potential applications of the work (8.2), directions for future research (8.3) and some final remarks (8.4).

8.1 Summary

The **first chapter** introduced the concept of compositional morphing and explained the motivating factors behind its use. The goals and methods were clarified and the key knowledge contributions were summarised, followed by a description of thesis structure.

The **second chapter** situated the thesis within the musical context of Mainstream Electronic Music (MEM) and reviewed musical practice for instances of morphing, both within MEM and within other musical contexts. MEM is the genre of choice for this research, that is, the techniques I have developed are designed to produce MEM.

In **chapter three** I introduced a new framework for discussing algorithmic music systems and reviewed various fields of algorithmic music in order to gain a sense of what would constitute novelty and also to inspire the research. The framework distinguishes between algorithmic music systems and simple music algorithms. Algorithmic music systems were conceived as composer agents which could take one of three approaches: **trial**, which involved trial and error; **heuristic** which involved rule estimation; and **abstract**, which involved mapping. Simple musical algorithms were described according to their musical **function** and the contextual **breadth**. The function continuum ranges from **analytic**, **transformational** through to **generative**. The contextual breadth is the amount of surrounding information that influences the computation of the algorithm.

The broad review of algorithmic composition covered composer agents, Computer Assisted Algorithmic Composition (CAAC), sonifications and DJ agents. In a review of interactive music, I examined meta-instruments, jamming agents, adaptive music and interactive installations. Following this, I reviewed note level morphing itself, covering four major works in detail and providing a brief overview of a number of other smaller projects. The potential for growth in the

area of note level morphing was highlighted, particularly as only one of the four major works was currently in active development. Most of the major examples of morphing were exploratory in nature and based on the **abstract** compositional philosophy. This demonstrated some additional room for development of morphing algorithms that embodied the **heuristic** and **trial** approaches to composition. In terms of research method, none of the previous morphing research projects utilised formal empirical testing procedures.

Chapter four described the system architecture of *LEMorpheus*, the software application that emerged from the thesis. The system allows the user to specify note-sequences and meta-parameters for each layer. Various morphing algorithms can be selected and parameters that influence the morph can be adjusted. The morph index is a high level parameter which controls how far the morph has progressed. Other parameters can be used to control structural aspects for each individual layer of the morph. Pitches are represented by a combination of scale, key, scale degree and passing note combined with octave. Aside from note onset, rhythm is dealt with via transformations – quantise, shuffle and loop. The software infrastructure is designed on the principle of extensibility and the system can be adapted for a range of applications. MIDI output is achieved through periodic polling of the morphing algorithm to generate the next segment of the note sequence, typically at quarter beat intervals. In a single cycle, the current beat is updated, meta-parameters are interpolated and morphed note sequence data for that cycle period is generated. Transformations to the generated note sequence are applied and the tonal representation is converted to MIDI pitch before the sequence is sent out to the synthesiser.

Chapter five explained the parametric morphing algorithm, which was the first note level morphing algorithm that was developed. It is parametric in the sense that abstract parameter envelopes are the primary representation for musical data. The morph is created by weighting the envelopes of the source and target according to the morph-index and combining them. The parametric morphing algorithm is only really effective when presented with examples of the source and target that are fairly similar.

Because the output of the parametric morphing algorithm was found to be so clearly lacking, formal testing did not seem necessary. There were a number of ideas for improvements to the parametric morphing algorithm, including re-implementation of Mathews and Rosler's self-synchronising function (1969), higher-level musical representations and phase offset detection. However, particularly as parametric morphing had been examined in the past, it was considered better to pursue more novel approaches.

Chapter six detailed a probabilistic approach to morphing, called the *Markov Morph*. Each play cycle, the *Markov Morph* randomly selects either the source or target, weighted on the morph index, so that there is more probability of selecting the source during the start of the morph and the target towards the end. A segment of the recent note output history is then compared to each segment of the same size in the selected source or target to create a similarity matrix. The similarity matrix is used as a probability distribution to predict the next note. A number of parameters can be changed to influence which notes are more likely to be selected, for example, increasing the weight of Circle of Fifth space in similarity judgements means that pitches separated by a fifth will be more similar. I demonstrated the influence of various parameters through some informal tests of the algorithm.

The *Markov Morph* was examined with a formal focus group, primarily to gain feedback on the questionnaire and research method but also to examine how people responded to different morphs, such as weighted selection and an early version of the *Markov Morph*. In these trials, morphing was benchmarked against simple cross-fading. The results were fairly unreliable, however, many ideas for improvements and feedback were collated and incorporated into subsequent formal tests.

Part of the evaluation was a 'focus concert' where the morphing algorithms were tested in a controlled (but at least semi-realistic) context. The *Markov Morph* algorithm was benchmarked against a human DJ who mixed the specified source and target material. The results indicated that the *Markov Morph* was at least competitive and, in some cases, the preferred choice, particularly with extended morphs. Obviously, a limitation of note level morphing is that it would require access to the MIDI sequences and synthesisers that were used to create the original tracks. The data generated from the focus concert was, on the whole, methodologically sound, however, some useful ideas for improvements were given by participants, for example, benchmarking against a human composer/producer rather than a DJ.

Chapter seven detailed a third and final morphing algorithm called *TraSe* which used evolutionary processes to generate the morph. Before playback, *TraSe* iteratively transforms the source until it becomes the target and the result of each iteration is a single frame in a whole series from source to target. During playback, the value of the morph index determines which of these frames are being played. This system design allows for a number of different compositional transformations to be explicitly incorporated into the algorithm and given more or less weighting by the user, thus allowing a substantial degree of musical control.

There are currently eight transformations through which, at each iteration, the note sequence transformed by the previous iteration is passed through as a chain: *divide/merge, rate, phase, harmonise, scale pitch, inversion, octave and add/remove*. Each of these have a number of different parameter configurations that produce different transformed results. The results are all compared to the target and ranked according to their measured dissimilarity with the target. A target dissimilarity is established through a user-defined parameter which controls how fast the morph will converge and the transformed result that matches the target level of dissimilarity is selected as the final output for that transformation.

The same process is executed on the key and scale to determine a logical progression of key modulations or chord sequences. Because the whole key/scale space is covered, the *TraSe* process for the *key/scale morph* is more controllable than the *TraSe* process for note sequences.

Informal testing of *TraSe* highlighted the emergent properties of the algorithm by generating morphs that differed quite markedly, despite only small changes to the morph parameters and no changes to the source and target. Playing the frames that were generated by *TraSe* in reverse order was informally evaluated. Forward morphing appeared to exhibit a more natural musical structure. Playing the morph in reverse order appeared less natural, as the dramatic change occurred at the end of the morph, a time which should mark some return to stability. The range of control for *key/scale morphing* was demonstrated through a number of examples that explored various flavours of morphing between C Major and the distant key of F# Major. The ability for *key/scale morphing* to generate Jazz-like chord progressions was also demonstrated through modulation from C Major to A# Major, with a harmonic function change and ii-V-I 'turnaround' in the new key.

The $O(n^3)$ time complexity for *TraSe* was confirmed empirically by testing source and targets samples with increasing numbers of notes. A comparison between *TraSe* morphing with the *add/remove* transformation only and *TraSe* morphing with all eight transformations on default settings was executed with fifty samples at different numbers of notes. It was found that all eight transformations can sometimes lead to convergence in less frames than is possible with *add/remove* alone. Despite this, the typical result of applying all eight transformations on default settings is many more frames than *add/remove* alone. The results may be improved with tuning of parameters and the use of 'musical' rather than 'random' source and target material.

Formal evaluation of *TraSe* was conducted through a qualitative online questionnaire, where participants could play through musical examples and answer a range of questions pertaining to

them. All participants were musically active, except for one who was an avid listener. The morphs were benchmarked against a human composer/producer who was asked to create a hybrid transition between the same sets of source and targets that were used by *TraSe* to generate the morph.

Participants were asked to verbalise a subjective response to the source and target, and list ideas regarding how they would approach the task of morphing between them. For each morph example, including those composed by the human, the participants were asked to provide a subjective response and discuss the smoothness and coherence of the morph. They also commented on important changes that occurred, noting the exact time and describing the effect that the change had on their listening experience. Lastly, the participants were asked whether or not the morph they heard would be acceptable to most people if it were applied to real-world contexts such as computer games or electronic dance music.

The morphs generated by *TraSe* were considered competent and sometimes perceived as innovative, however they were more often criticised for deficiencies in structural clarity and coherence. In contrast, the human composed morphs were more acceptable overall, although were never viewed as particularly innovative and sometimes were considered to be too plain.

Potential extensions to the *TraSe* morph algorithm include: note thinning, note clustering, new transformations, new combinations of the transformations, automatic adjustment of parameters, layering and higher level structural control of the music and optimization of current process. New combinations of transformations would provide many new possibilities, rather than the single linear chain of transformations that is currently employed. Automatic adjustment of parameters would allow all the morphs that converge in a practical number of frames to be found with little effort. An algorithm which can determine when and how to perform dramatic changes, such as breakdowns, would add some realism and a clearer sense of purpose or coherence to the music generated. Optimisation of *TraSe* would afford greater realtime interactivity, as the morph would not need to be precomputed.

8.2 Demonstrations of potential applications

The morphing software has been trialled in three different contexts: concerts, installations and a computer game. These demonstrate how the morphing techniques that were developed can be applied within real-world contexts. The concerts show how the morphing software can be used as a meta-instrument, with the high-level control of the morph index facilitating a comfortable performance. The interactive installations show how the morphing algorithms can be the basis of an engaging social musical interface. The computer game application demonstrates how morphing can automatically adapt a musical score to suit changes in the game and generate new material.

8.2.1 Concerts

During stage three, some morphs were created for a lunch time concert at the Queensland University of Technology and were also used later for the onsite concert at the Australasian Computer Music Conference (ACMC) 2005. This is included here (~8.1, ~8.2).

At this stage in the development of *TraSe*, key/scale was not a separate representation that could be morphed independently of the note data. Instead, two compositional transformations, *transpose* and *mode-lock* were included within the chain of compositional transformations so as to enable changes in key and scale. This is noticeable in some of the more drastic changes that occur within the music.

At *Earpoke*, the offsite event for ACMC 2006, a more improvised use of the morphing software occurred using some of the material developed for the online questionnaire. For this event, the separate *key/scale morphing* was used. The software was used alongside live saxophone, boxo, and clapping.

8.2.2 Interactive Table Installation

A tabletop interface has been developed that demonstrates the morphing algorithms being applied to the context of interactive installation/toy/instrument. It has been showcased at four events: Sound Polaroids at The Brisbane Powerhouse, Bowerbird at Substation Number Four, as part of Richie's Electronic Playground at the Peats Ridge Festival and the opening of the Computational Arts Research Group (CARG) at QUT. The gigs at the substation and the Peats Ridge Festival were almost completely free of technical problems and were accordingly the best

demonstrations of the technology. Overall, the morph table performs well, particularly in terms of levels of engagement, musical acceptability, meaningful interaction and social participation.

Figure 1 Pictures from the Morphing table installation

The table is a flat transparent surface upon which four different glowing cubes can be placed. Each of the cubes relates to one of four different parts: drums, bass, lead, pads. The cubes are tracked by a webcam and the locations of the cubes are sent to *LEMorpheus*. When a cube is on the total left-hand side of the table, the morph index for that part is zero and moving the cube across to the right will increase the morph index for that part continuously until it reaches one at the right-hand edge. Moving the cube away (up, on the screen) will increase the sound effects that are applied to that part in the synthesiser (*Reason*^{TM}) – for example reverb, delay or distortion depending on the patch – and moving it closer will reduce the sound effects.

Four sides to each cube each have a different fiducial that represents a different morph for that part that has been loaded up before the start of the event. The two other faces of the cube house the part labels for the cube, for example “Bass”. Each time the cube is flipped, fiducial down, the music for that part is switched to the morph represented by that fiducial. Each of the four morphs has a totally different set of instruments and the source and target patterns for each morph are also on different instruments, which means thirty-two channels, each with a different patch and sound effects are running simultaneously in the synthesiser. To achieve this, *LEMorpheus* sends out on two MIDI ports.

Each cube has a carefully designed cardboard 'inner-cube', within which are embedded thirty-six bright Light Emitting Diodes (LEDs), nine for each of the four sides of the cube. The inner-cube has supporting flukes on each edge to hold it in the middle. All of the LEDs run off two AAA batteries, which maintains sufficient power levels for at least two hours.

The table utilises the reacTIVision toolkit developed at the University of Pompeau Fabra (Bencina, et al. 2006) to track the fiducials and send changes to *LEMorpheus* via Open Sound Control (OSC). A Mac Mini running boot-camp and *reacTIVision* utilises a wide angle (seventy-two degrees) webcam underneath the table to track the cubes. The locations of the fiducials are sent to through Ethernet cable to an Acer TravelMate 8000 running Windows, *LEMorpheus* and *Reason*^{TM}. The locations are picked up by *LEMorpheus*, translated into morph indexes for various parts and the resulting MIDI output is sent to *Reason*^{TM} via two Midishare (Grame 2004) ports and MIDI Yoke (O'Connell 2003). The sounds are the rendered by *Reason*^{TM} and outputted to the speakers which are under the Morph Table.

The table itself was co-designed by Brendan Wright and myself and constructed by Brendan Wright. The light emitting cubes were co-designed and built by Kate Thomas and myself, with electronic engineering guidance from Matt Petoe. Andrew Brown supervised the morph table project.

A short documentary about the Morph Table was made by Conan Fitzpatrick (~8.3).

8.2.3 Computer Game

As a preliminary investigation into the application of morphing for a computer game context, a morph was adapted for the open-source adventure game Beige. In this game, a character walks between various scenes on a map, killing creatures and collecting gems. The classic adventure game genre was chosen due to its topological affordances and small Central Processing Unit (CPU) load. Beige in particular was selected from many other games due to its being open source, fully developed, and able to be easily modified. The mapping that was chosen was the x-axis position of the character's avatar in the world map. The world consisted of a 6x6 grid of scenes and the morph index was quantised to remain constant across each scene. That is, when the character moves into a new scene to the right, the morph index decreases by $\frac{1}{6}$, and when the character moves into a new scene to the left, the morph index increases by $\frac{1}{6}$. Moving up or down has no effect at all. Video documentation of this development, which includes a view of the morph index above the game screen, is included (~8.4).

From this demonstration, it is clear that morphing can be used to automatically compose interesting transitional material that relates to an aspect of the computer game. The fact that the music changes subtly in each scene is a new aesthetic that is quite uncommon in most computer-games. The standard is for the same music to be repeated constantly within one whole region (multiple scenes) and then cross-fade to a pre-composed transition when entering the next region – for a composer to create the huge amount of material that would be needed otherwise would be very uneconomic. In the example (~8.4), only two pieces of music were composed (source and target) while the four intervening pieces were generate by the morphing algorithm. Whether or not this is a significant or welcome change and whether or not the particular music generated is suitable is another question that can only be answered in further formal tests or market research.

While this simple study is sufficient for a preliminary investigation, there are a number of deficiencies that would need to be addressed for application to a real game:

- Musical structure vs game-play: when the character shifts into a new scene, the changes need to occur on the beat, not exactly when the shift happens.
- Utilising the Y-Axis: currently the morph is only between a source and a target, not multiple sources.
- Temporary cues such as the killing of a creature or the collection of a gem are overlooked in the music.
- Timing glitches: Java **garbage collection** occasionally causes a timing glitch.
- The high CPU load of synthesis: during this demo, the synthesiser was hogging between 20-40% of the CPU which is a totally unacceptable 'CPU budget' for audio in the commercial computer game industry (Brown 2006; Edlund 2006; Sanger 2003).

8.3 Future research

For future research into automatic and interactive note level morphing of mainstream electronic music, a number of ideas have been conceived that would be likely to contribute to substantial improvements. I have catalogued these ideas into four primary directions of research: algorithmic composition techniques, production techniques, evaluation methods and applications.

Potential algorithmic compositional techniques include:

- Note thinning. For the *TraSe* morph this could involve a combination of reducing the number of notes if it is greater than the average in source and target and removing notes that overlap, are very close together or outside of the expected metre or tonality.

- Note clustering/phrase detection. *TraSe* morph could use a pre-analysis of the material and determine which phrases from the source and target to morph together. In the *Markov Morph*, phrases could be played through to completion, rather than inter-splicing them randomly.

- New transformations. *TraSe* morph depends on hard-coded transformations for the range of the compositional decisions that generate the result and therefore new transformations would add greater musical possibilities. For the parametric morphing algorithm, the continuous parameter envelopes which represent the music can each be thought of as a transformation of a fundamental pattern, for example, the inter-onset envelope controls the value for a 'rate' transformation, the pitch envelope controls the value for a 'transpose' transformation and the phase envelope controls the value for a 'phase shift' transformation. Given this, any other transformation that is governed by a continuous parameter could be adapted to the parametric morphing algorithm and potentially increase the appearance of more human-like compositional thought in the morph.

- New or extended note similarity measures. *TraSe* morph has a number of similarity measures, each of which complements a transformation. The similarity measures could be adjusted to better suit the transformation to which they apply. The similarity measures used in the *Markov Morph* could deal better with polyphony by using the

Nearest Neighbour measure. The *Markov Morph* should also incorporate other musical dimensions such as dynamic and inter-onset into the similarity measurements.

- Combinations of transformations. In the *TraSe* morph, new methods of combining the transformations could be developed that are more effective. For example, using all possible combinations of the transformations would create a greater diversity in the potential output. This may be computationally quite intensive, so perhaps it could be tested and only a few different combinations which are the most useful are used.

- Layering and structural changes. All morphing algorithms developed thus far could benefit with a more natural approach to layering and musical structure. Rather than always attempting a smooth morph on each layer simultaneously, the algorithm should be able to, for circumstances where the source and target material are particularly 'different', cut some layers out and add them back, in a way adheres to a musical structure.

- Automatic adjustment of parameters. For the *TraSe* morph, all of the weightings for each of the transformations could be searched automatically, so as to find the various different solutions to the morph that have an acceptable number of frames. For the *Markov Morph* the 'contrast' parameter in particular should be automatically adjusted so that the probability distribution always has a target level of variance. For example, it would be decreased automatically by a certain amount if the note sequence in the seed was foreign to the selected source or target.

- Constraints. Within the *Markov Morph* and the parametric morphing algorithm, rhythmic constraints would add a degree of stability. For example, with parametric morphing, the inter-onset intervals and the onsets could be quantised to those that are present in the source and target. With the *Markov Morph*, a metric template with weightings for each possible onset could be extracted from the source and target, and the probability of the notes playing on the onsets could be factored by the meter. Another constraint that would be useful within the *Markov Morph* is to restrict the note generation process to each time a note is played, rather than every frame, so as to avoid stream-loss.

- Data-mining. The *Markov Morph* could incorporate a database of musical material to inform the probability matrices. For the *TraSe* morph, there is a possibility that compositional transformations could be automatically extracted from a database.

- Higher-level musical constructs. As an extreme example, parametric morphing could interpolate between the high level psychological dimensions of intensity and valence (Wundt 1896), which would drive a hierarchy of other musical dimensions such as rate of tonal change, tonal stability, number of independent streams, level of polyrhythm, additive rhythm, syncopation and many others discussed in Chapter Two.

- Automatic mixing via MIDI volumes and control parameters to suit acceptable production values, for example, balancing the parts and reducing muddiness. This would be applied to the system infrastructure rather than a specific morphing algorithm.

Potential new data gathering methods include:

- An online forum style questionnaire, where the aesthetics of the morphing examples are debated continuously and in more detail. This would also better suit the rapid iterative development methodology that is applied to the software. For example, if a particular deficiency emerges from the debate, it could be quickly addressed, until the various aesthetic protagonists are satisfied as much as is practical.

- A music competition where the morphs composed via *LEMorpheus* are covertly entered alongside a range of human composers' morphs and judged by an unknowing expert listening panel. All of the pieces would be ranked and data could be gathered from the assessment criteria and comments filled out by the panel.

- Playing the *LEMorpheus* morphs alongside human composed morphs on a radio station and inviting listener feedback. This style of evaluation has also been employed by Pachet (2005). The advantage is that it covers a wide range of audiences in a realistic context, although, as it is a single event, there is less control and the responses are less able to be directed.

Potential developments relating to applications of morphing include:

- Optimisation. Increasing the time efficiency of the *TraSe* morph would enable changes to morph parameters and the source and target sequences to become the subjects of interactivity, rather than only the morph index.

- N-source morphing. Morphing between many sources would better suit applications such as computer games where the non-linear narrative to which the music adapts

contains multiple themes that should be integrated. This may also better suit Cartesian interfaces into the music, such as the Morph Table.

8.4 Concluding remarks

At a fundamental level, the research utilised known musical processes and concepts from standard music theory and algorithmic music, which was necessary to enable some degree of musical stylistic coherence. These were applied in a novel way to produce successful musical morphing between two distinct note sequences. However, considering the lack of theoretical texts on note level morphing, substantial effort was devoted to the exploration of new aesthetic possibilities. This included new processes that would be near intractable without the aid of a computer, for example, the evolutionary *TraSe* algorithm.

Perhaps the most interesting findings to emerge from the research were the differences in approach between the human composer/producer and the automated morphing algorithms I created. When confronted with a source and target that seemed incompatible, the human composer used dramatic changes but integrated them in a way that appeared natural. This highlighted the fact that smoothness is not always perceived in terms of moment to moment continuity of the musical surface, but sometimes in terms of how closely the moment to moment expectations are fulfilled – if dramatic changes are expected, they will be perceived as 'smooth' changes. This finding is interesting because it implies an effective new form of morphing at the level of musical structure, which has, until now, been largely overlooked within previous research, which tended to focus on continuity of the musical surface.

Another interesting outcome was the polarised response to particular automatically generated morphs – some people perceived them to be novel, innovative and skilful, while others expressed immense dislike. This is extremely rewarding as much algorithmic music so often achieves the exact opposite – a middle-ground that is inoffensive but not exceptional. The fact that at least some people had such an intensely positive response to some of the examples is also rewarding and shows promise that the techniques could be refined to be very successful in the future.

It is also pleasing that the current note level morphing techniques are applicable to real world contexts, as evidenced by the overwhelming agreement from the questionnaire respondents. The occasional remarks to the contrary were balanced by similar remarks pertaining to the human composed transitions. To further demonstrate this, the software was employed in real world contexts at multiple events throughout the duration of the study, including live

performances and interactive installations and was well received. Despite this, it should be noted that many technical hurdles must be overcome before more widely accessible commercial applications can be realised.