From Motion to Musical Gesture: Experiments with Machine Learning in Computer-Aided Composition

Jean Bresson[1], **Paul Best**[1], **Diemo Schwarz**[1], **Alireza Farhang**[2]
[1]Ircam, CNRS, Sorbonne Université: Science et Technologies de la Musique et du Son (STMS), Paris, France
[2]Royal Conservatoire Antwerp, Belgium

Abstract

This paper presents preliminary works exploring the use of machine learning in computer-aided composition processes. We propose a work direction using motion recognition and audio descriptors to learn abstract musical gestures.

Introduction

Contemporary music creation has long taken advantage of technology and computing systems to increase sonic and compositional possibilities, enhancing at the same time the expressivity and language of musicians, and the experience of music listeners. Artificial intelligence inspired the very beginnings of computer music (Hiller and Isaacson 1959), as well as the work of several contemporary composers, such as David Cope's *Experiments in Musical Intetlligence* (Cope 1996), Shlomo Dubnov's *Memex* compositions (Dubnov 2008), or Daniele Ghisi's recent project *La Fabrique des Monstres*, which demonstrated the generative potential of machine learning using raw sound signals (Ghisi 2018).

Machine learning techniques were also intensively applied for data mining and classification in the fields of Music Information Retrieveal (Pearce, Müllensiefen, and Wiggins 2008; Illescas, Rizo, and M. 2008), computational musicology (Camilleri 1993; Meredith 2015), human-machine co-improvisation—where computer agents learn from musical sequences in order to produce new sequences imitating a style or a "mixture" of styles (Assayag, Dubnov, and Delerue 1999), research of instrumental combinations for the synthesis of orchestral timbres (Esling, Carpentier, and Agon 2010; Crestel and Esling 2017), or gesture following in real-time musical interaction (Françoise, Schnell, and Bevilacqua 2013). They were recently brought to the forefront of music technology research (Dubnov and Surges 2014), with publicised projects such as *Flow Machines* (Ghedini, Pachet, and Roy 2015), numerous mainstream products and online services, as well as a whole new field of research applying deep-learning techniques to varied aspects of music processing (Briot, Hadjeres, and Pachet 2018).

Despite a great variety of possible applications, and the few examples cited previously, however, to our knowledge machine learning and AI remain seldom used by professional composers. Composition in musical metacreation research is usually addressed as a means to substitute humans in realizing tasks requiring some kind of creativity (Pasquier et al. 2016), hence somehow challenging the very role and status of composers within their own field of expertise.

The field of computer-aided composition—as defined for instance in (Assayag 1998)—displays a somehow opposite approach, focusing on users' creative input and subjectivity in the design of musical processes. This standpoint led to the development and adoption of a more "constructivist" approach (in the sense that musical objects are made and structured explicitely via generative or transformational programs), and to other emerging aspects of information technology, such as end-user programming (Burnett and Scaffidi 2014) and visual programming languages (Assayag 1995). Our current work lies within this field of research: in the OpenMusic (OM) visual programming language, composers can freely develop, formalize, and implement ideas under the form of programs associated to varied musical representations (Bresson, Agon, and Assayag 2011).

Programming in a system like OM provides extended expressive power to trained composers, allowing them to make the computer do exactly what they want, in accordance with formalized objectives and work procedures. In this context, machine learning and AI must therefore be employed and approached following a slightly different perspective. An objective of this work will be to enhance the environment with an easy and controlled access to these techniques in computer-aided composition practice.

We report here on preliminary experiments applying existing machine learning technology, initially dedicated to motion data, to audio descriptors for the classification of "gestures" within musical extracts. An ongoing compositional project is leveraged as a context and source for experimental data, for which we have put together a work environment embedding computer-aided composition, machine learning procedures, and visual programming. Through this particular example is also addressed the challenge of grasping the concept of "musical gesture" using physical motion recognition tools.

Learning Musical Gestures

Musical structures carry abstract features and characteristics that can be straightforward to identify by composers and/or listeners, but difficult or impossible to formally describe using the elements of standard score representations (e.g. identifying harmonic/melodic patterns etc.) The concept of "gesture" is frequently found in compositional discourse and studies to characterize such structures, yet in a variety of different and more or less abstract meanings (Godøy and Leman 2010; Hervé and Voisin 2006; Farhang 2016). In (Maxwell, Eigenfeldt, and Pasquier 2012), the authors use the term of *object* to similarly describe elements of the score's surface which can be "captured" and have a specific meaning for the composer.

In the field of Movement & Computing research (MOCO),[1] machine learning technology today permits to deal with gestural data input, mapping, and processing using powerful and operational tools (Wanderley 2002; Bevilacqua et al. 2011; Caramiaux et al. 2014). One of such technologies is the model developed by Jules Françoise in the XMM[2] library. Based on hybrid techniques combining Gaussian Mixture Models and Hidden Markov Models, XMM dynamically analyses time-series and streams of gesture-description signals in order to classify movements: at each time of a real-time data input can be estimated the highest-probability gesture being performed, as well as the position of this estimation within a global model of the gesture (Françoise, Schnell, and Bevilacqua 2013).

The experiment described in this paper suggests that such motion learning and recognition technique (usually applied to physical gestures) might be used as well to recognize and process abstract musical gestures.

Recent computer-aided composition works and projects outlined a connection between motion/gesture descriptions and more abstract musical conceptions. In Jérémie Garcia's *pOM* project with composer Philippe Leroux, for instance, hand-drawn pen gestures were converted to symbolic compositional structures (Garcia, Leroux, and Bresson 2014). Marlon Schumacher's *OM-Geste*[3] library for OpenMusic (Schumacher and Wanderley 2017) encodes multidimensional gesture-description signals and maps them to musical objects at different scales and time-resolutions, in order to process these gestures in compositional workflows leading to the generation of symbolic musical structures (scores), or to the control of sound synthesis. Machine learning could provide new insights on musical gestures in such computerized compositional processes.

A Machine Learning Framework in OM

We developed an operational framework for gesture learning and recognition, wrapping the XMM library within the OpenMusic computer-aided composition environment.

The prototype under development (OM-XMM[4]) considers simple pairs {*data*, *label*} for building, training and running a model. The *data* is a vector of n time-series corresponding, in the standard case, to a set of temporal descriptors (e.g. x, y, z positions, acceleration, orientation, etc. for a motion description, but actually any other set of discrete signals as well).

Building efficient machine learning models requires careful parameterization, training, data selection and weighting, observation and analysis of intermediate results. The OM-XMM library provides a number of facilities and test procedures for model validation (testing it over ground-truth data) and hyperparameters optimization (for instance using a genetic algorithm adjusting parameter values). In addition, the environment provides a whole set of general-purpose visual programming facilities to implement such experimental procedures (Bresson and Agon 2010), embedding them in iterative processes and storing/displaying results easily.

XMM models have the advantage of not requiring large training-sets: a few examples can be enough to recognize simple shapes performed, drawn or input in the system. This characteristic fits well with most composers' workflow, where we can assume that generally training sets will be of a relatively small dimension. In Figure 1, an OM patch (visual program) trains an XMM model with a few labeled hand-drawn shapes (point coordinates and derivatives for speed). The model is then used to recognize and output the name of additional input shapes.

Application

The OM-XMM library was used in composer Alireza Farhang's 2018 musical research residency project at Ircam.[5] In this project, the composer's intention was to produce a "data-flow score" from audio performances, in order to control simultaneous performances in various media involved in a multidisciplinary work. Part of this data-flow score elaboration consists in identifying abstract "gestures" in existing or incoming audio material, and using these gestures as a common ground for sequencing and controlling actions for all the players and components of the work. Audio streams must therefore be segmented and processed to output the description of a discrete sequence of gesture labels, according to a given training set of audio-recorded material.

To build the training set, the composer manually annotated a score with labels corresponding to different classes of musical gestures (A, B, C...). Each class is actually matched as well to a more personal, graphical symbol. The subjective aspect in this process is important: this classification does not necessarily rely on quantifiable information, and

[1] www.movementcomputing.org/

[2] ircam-rnd.github.io/xmm/

[3] github.com/marleynoe/OM-Geste

[4] github.com/openmusic-project/om-xmm – The prototype and figures presented in this paper run in the "o7" implementation of the visual language (Bresson et al. 2017). The library is also compatible with OM 6.13 on macOS systems. The presented examples and OM patches are available and distributed along with the OM-XMM library sources.

[5] www.alirezafarhang.com/alirezafarhang/texts_traces_of_expressivity_en

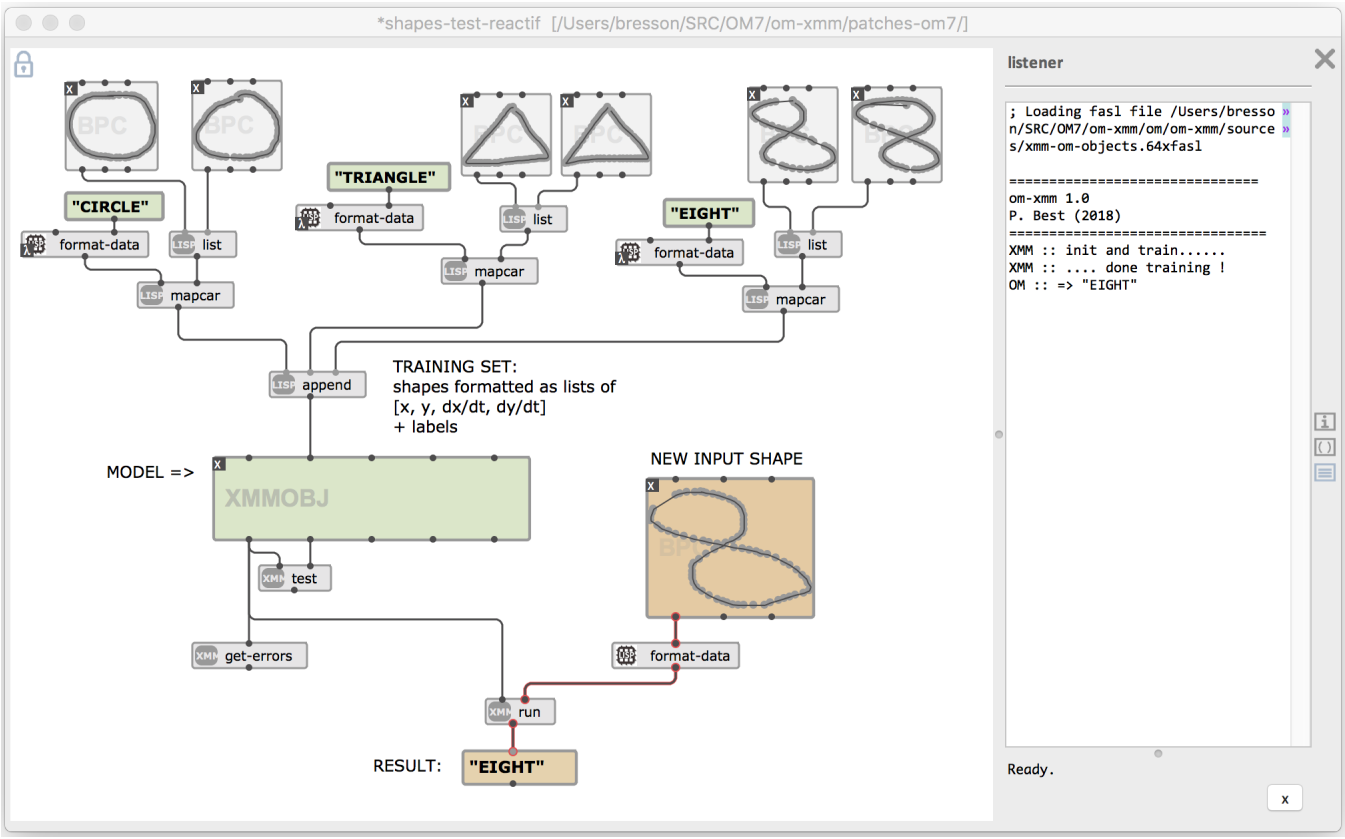

Figure 1: Basic shape recognition model using XMM in the OpenMusic visual programming environment. Using a small training set of [*point-coordinates*, *class-label*], the model can estimate the class of a new input (hand-drawn shape at the right).

can be informed by any consideration from the composer. Corresponding segments were then extracted from an audio recording of the score performance, labeled with class-name tags, analyzed through several audio descriptors, and used to train a model to estimate the class of further incoming audio segment (or comparable vector of audio-descriptor signals).

This initial approach is pretty basic, and the quality of the results will highly depend on the comparative nature of the training vs. incoming audio signals, and most importantly, on the choice of sound descriptors and parameters used for building and running the model. These tasks cannot be fully automatized and all the settings must be fine-tuned and tested according to the composer's specific use, material at hand, and subjective goals. The main challenge at the core of this experiment is therefore to help with the appropriation and integration of the "machine learning workflow" within the composer's work and practice. Pre-processing, categorization, labeling of a training data-set, calibration and fine-tuning of the system, must all become part of the composer's work and therefore require adequate tools, taking into account the specificity of his/her profile, expertise, and artistic approach.

Figure 2 shows a composer's workspace in OpenMusic, including an XMM model trained and tested over a series of sound extracts for gesture classification. The upper part of the visual program creates a dataset from a list of sound files, each named after its assigned label (e.g. "002Q.aiff" means this is segment #2, of class *Q*). Note that for this experiment, we consider that a same segment can correspond to, or be a "mix" of several gesture classes, hence we find for instance both "002Q.aiff" and "002Z.aiff" in the training dataset. On the left is an indication for the training-set building procedure to use MFCCs and a given subset of audio descriptors (*descr*) to create the *data* corresponding to each sample (here, descriptors #1, 2, 3, 4, 14... are selected among the whole set of available audio descriptors).[6] Once the model is built and trained, the *run* function at the bottom allows to test a new extract (or set of descriptor signals) for classification: the output of *run* will be one or more class labels, returned along with respective likelihood scores.

Preliminary Results and Conclusions

Results are currently being collected and analyzed: overall, they are not yet satisfactory to the point of being reliable for accurate gesture classification. A satisfactory point, from our perspective, was the composer's ability to get into the

[6]The sub-patch *make-dataset* uses the PiPo stream processing framework embedded in the IAE audio library to perform all audio-descriptor analyses (Schnell et al. 2017).

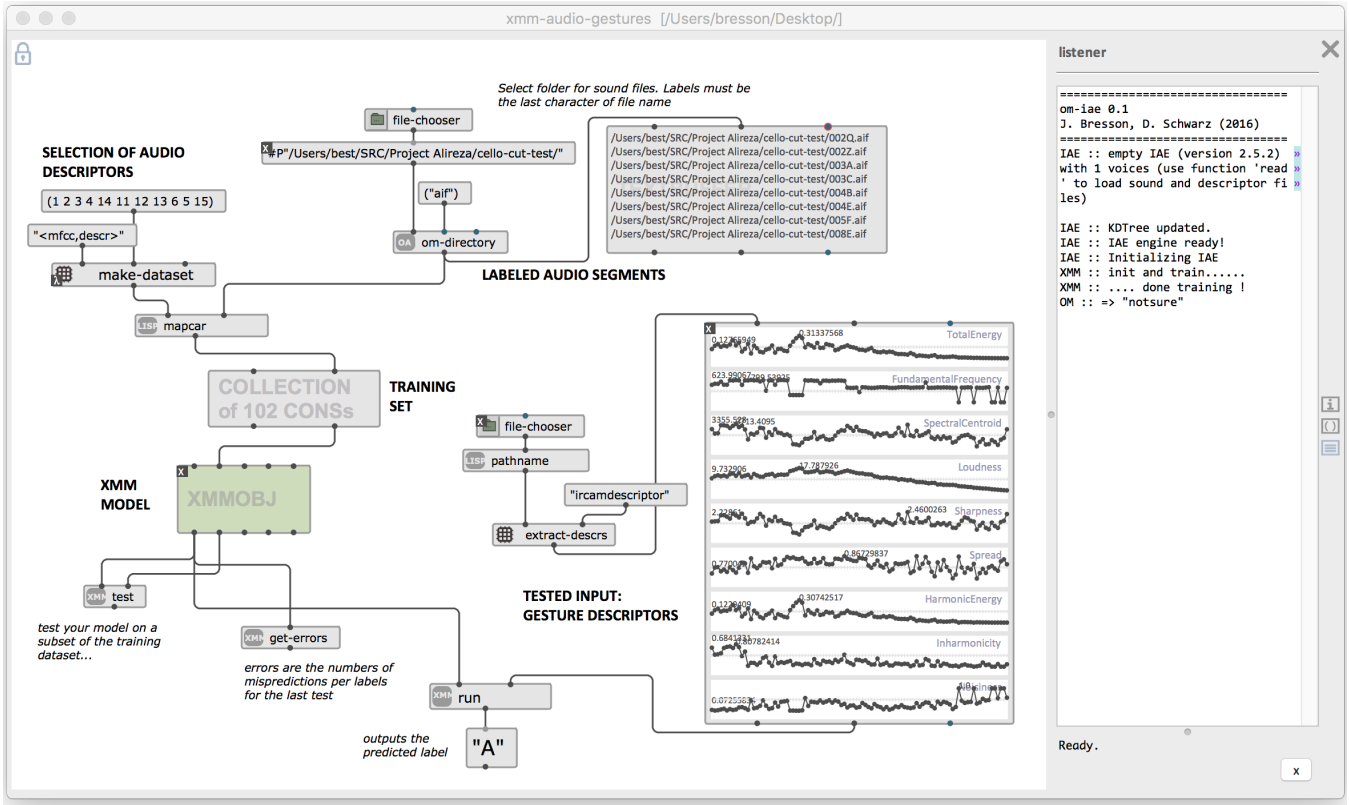

Figure 2: Classification of musical "gestures" from audio signal descriptors: an OpenMusic composer's workspace.

experimental process of training, testing, running machine learning models by himself within his usual composition environment and software. Hyperparameter optimization also helped singling out a subset of more relevant audio descriptors to use, and other settings of interests in the model.

Besides this specific experiment and application with gesture-recognition and analysis, we hope this will be a starting point for more integration of machine learning and AI-related tools in the computer-aided composition environment. A great amount of tools and technology are currently available, which could be used to help, accelerate, or improve other compositional tasks. Techniques such as neural networks, data clustering, or Bayesian networks could well be adapted and applied to symbolic (sequential, hierarchical...) musical structures, for instance to classify and process chords, melodies, patterns, diagrams, or spatial information. Prospective applications might then include varied tasks and stages of compositional processes: analysis and transcription, complex problem solving and operational research, composition by re-composition or concatenation of patterns, etc. In this range of applications, machine learning provides an opportunity to better understand, control, or generate musical structures emanating from composers' formalized thinking, musicianship, and creativity.

Acknowledgments

This work is carried out within the PEPS I3A exploratory projects support framework of the French National Center for Scientific Research (CNRS).

References

Assayag, G.; Dubnov, S.; and Delerue, O. 1999. Guessing the Composer's Mind: Applying Universal Prediction to Musical Style. In *Proceedings of the International Computer Music Conference (ICMC'99)*.

Assayag, G. 1995. Visual Programming in Music. In *Proceedings of the International Computer Music Conference (ICMC'95)*.

Assayag, G. 1998. Computer Assisted Composition Today. In *1st symposium on music and computers*.

Bevilacqua, F.; Schnell, N.; Rasamimanana, N.; Zamborlin, B.; and Guédy, F. 2011. Online Gesture Analysis and Control of Audio Processing. In Solis, J., and Ng, K., eds., *Musical Robots and Interactive Multimodal Systems*. Springer.

Bresson, J.; Agon, C.; and Assayag, G. 2011. OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research. In *ACM MultiMedia (MM'11) OpenSource Software Competition*.

Bresson, J., and Agon, C. 2010. Processing Sound and Music Description Data Using OpenMusic. In *Proceedings of the International Computer Music Conference (ICMC'10)*.

Bresson, J.; Bouche, D.; Carpentier, T.; Schwarz, D.; and Garcia, J. 2017. Next-generation Computer-aided Composition Environment: A New Implementation of OpenMusic. In *Proceedings of the International Computer Music Conference (ICMC'17)*.

Briot, J.-P.; Hadjeres, G.; and Pachet, F. 2018. *Deep Learning Techniques for Music Generation*. Computational Synthesis and Creative Systems. Springer.

Burnett, M., and Scaffidi, C. 2014. End-User Development. In Soegaard, M., and Dam, R. F., eds., *The Encyclopedia of Human-Computer Interaction*. Interaction Design Foundation.

Camilleri, L. 1993. Computational Musicology. A Survey on Methodologies and Applications. *Revue Informatique et Statistique dans les Sciences Humaines* 29.

Caramiaux, B.; Montecchio, N.; Tanaka, A.; and Bevilacqua, F. 2014. Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems* 4(4).

Cope, D. 1996. *Experiments in Musical Intelligence*. A-R Editions.

Crestel, L., and Esling, P. 2017. Live Orchestral Piano, a system for real-time orchestral music generation. In *Proceedings of the Sound and Music Computing Conference (SMC'17)*.

Dubnov, S., and Surges, G. 2014. Delegating Creativity: Use of Musical Algorithms in Machine Listening and Composition. In Lee, N., ed., *Digital Da Vinci: Computers in Music*. Springer.

Dubnov, S. 2008. Memex and Composer Duets: Computer-Aided Composition Using Style Modeling and Mixing. In Bresson, J.; Agon, C.; and Assayag, G., eds., *The OM Composer's Book. 2*. Editions Delatour / Ircam.

Esling, P.; Carpentier, G.; and Agon, C. 2010. Dynamic Musical Orchestration using Genetic Algorithms and Spectro-Temporal Description of Musical Instruments. In *Applications of Evolutionary Computation: EvoApplications 2010*, LNCS 6024. Springer.

Farhang, A. 2016. Modelling a gesture: *Tak-Sīm* for string quartet and live electronics. In Bresson, J.; Agon, C.; and Assayag, G., eds., *The OM Composer's Book. 3*. Editions Delatour / Ircam-Centre Pompidou.

Françoise, J.; Schnell, N.; and Bevilacqua, F. 2013. A Multimodal Probabilistic Model for Gesture-based Control of Sound Synthesis. In *ACM MultiMedia (MM'13)*.

Garcia, J.; Leroux, P.; and Bresson, J. 2014. pOM: Linking Pen Gestures to Computer-Aided Composition Processes. In *Joint International Computer Music / Sound and Music Computing Conferences (ICMC–SMC'14)*.

Ghedini, F.; Pachet, F.; and Roy, P. 2015. Creating Music and Texts with Flow Machines. In Corazza, G. E., and Agnoli, S., eds., *Multidisciplinary Contributions to the Science of Creative Thinking (Creativity in the Twenty First Century)*. Springer.

Ghisi, D. 2018. La Fabrique des Monstres. [Online:] http://www.danieleghisi.com/works/la-fabrique-des-monstres/.

Godøy, R. I., and Leman, M., eds. 2010. *Musical Gestures: Sound, Movement, and Meaning*. Routledge.

Hervé, J.-L., and Voisin, F. 2006. Composing the Qualitative, on *Encore* Composition. In Agon, C.; Assayag, G.; and Bresson, J., eds., *The OM Composer's Book. 1*. Editions Delatour / Ircam-Centre Pompidou.

Hiller, L. A., and Isaacson, L. M. 1959. *Experimental Music: Composition With an Electronic Computer*. McGraw-Hill.

Illescas, P. R.; Rizo, D.; and M., I. J. 2008. Learning to Analyse Tonal Music. In *Proceedings of the International Workshop on Machine Learning and Music*.

Maxwell, J.; Eigenfeldt, A.; and Pasquier, P. 2012. ManuScore: Music Notation-based Computer Assisted Composition. In *Proceedings of the International Computer Music Conference (ICMC'12)*.

Meredith, D., ed. 2015. *Computational Music Analysis*. Springer.

Pasquier, P.; Eigenfeldt, A.; Bown, O.; and Dubnov, S. 2016. An Introduction to Musical Metacreation. *Computers in Entertainment* 14(2).

Pearce, M. T.; Müllensiefen, D.; and Wiggins, G. A. 2008. A Comparison of Statistical and Rule-based Models of Melodic Segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR'08)*.

Schnell, N.; Schwarz, D.; Larralde, J.; and Borghesi, R. 2017. PiPo, A Plugin Interface for Afferent Data Stream Processing Modules. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'17)*.

Schumacher, M., and Wanderley, M. 2017. Integrating gesture data in computer-aided composition: A framework for representation, processing and mapping. *Journal of New Music Research* 46(1).

Wanderley, M., ed. 2002. *Mapping Strategies in Real-time Computer Music*. Special Issue of *Organised Sound* 7(2). Cambridge University Press.