SOUND SEARCH BY CONTENT-BASED NAVIGATION IN LARGE DATABASES

Diemo Schwarz, Norbert Schnell
Ircam – CNRS STMS
Paris, France

ABSTRACT

We propose to apply the principle of interactive real-time corpus-based concatenative synthesis to search in effects or instrument sound databases, which becomes content-based navigation in a space of descriptors and categories. This surpasses existing approaches of presenting the sound database first in a hierarchy given by metadata, and then letting the user listen to the remaining list of responses. It is based on three scalable algorithms and novel concepts for efficient visualisation and interaction: Fast similarity-based search by a *k*D-tree in the high-dimensional descriptor space, a mass–spring model for layout, efficient dimensionality reduction for visualisation by hybrid multi-dimensional scaling, and novel modes for interaction in a 2D representation of the descriptor space such as filtering, tiling, and fluent navigation by zoom and pan, supported by an efficient 3-tier visualisation architecture. The algorithms are implemented and tested as C-libraries and Max/MSP externals within a prototype sound exploration application.

1 INTRODUCTION

For sound design, film and multi-media production, and musical creation, databases with instrumental or environmental sounds and sound effects are a vital resource. The number and size of commercially available sound effects databases, such as *Hollywood Edge*, *Sound Ideas*, loops collections, or community-driven on-line collections like *freesound* are growing steadily, [1] with rising network bandwith, growing harddisk capacity, and falling prices for storage and distribution media accelerating this growth even further.

From a certain scale onwards, the practical problem in the exploitation of these databases is no longer the question if a specific sound exists in the database, but how to find it. In a user survey conducted within the *SampleOrchestrator* project, professional film sound designers reported about their practice of collecting the soundtracks of the *rushes*, i.e. the raw, unedited footage shot during the making of a film, to augment their collection of ambiences, reaching the mark of 1 TB of sound data, and their difficulty of finding one special event in many long recordings with tools not adapted to such large sizes. They get by with the disciplined use of manually edited metadata, and orient themselves by the audio waveform displayed in a sound browsing application.

Our contribution to alleviating the problems of finding the right sound in a mass of unstructured recordings is interactive navigation with immediate audio feedback in a space of sound descriptors populated by sound segments. This approach greatly speeds up the usual workflow of hierarchical menu or search mask, result list, and play/stop buttons that put many mouseclicks between the user's idea of the sound and listening to appropriate contents of the database.

We propose to replace the menu- and list-driven interface with a 2D representation of a sound and category space. While navigating through the space, the sound segments close to the current position are immediately played. Playing is layered if movement is fast, so that large parts of the sound space can be explored rapidly. The strong interactivity enables the user to quickly understand the dimensions and areas of the presented space by probing sound snippets that are played as they are passed by.

This principle of navigation poses tougher requirements on the efficiency of the underlying algorithms, and on their scalability to very large databases. An accompanying article [18] concentrates on fast similarity-based search and the efficiency of low-dimensional embedding of the high-dimensional descriptor and category space with special attention to scalability. There, we chose and improved three algorithms that are also briefly described in section 3: the *k*D-tree search algorithm, the simulation of a mass–spring–damper (MSD) system, and the hybrid multi-dimensional scaling algorithm for dimensionality reduction.

We will then treat the aspects of efficient visualisation and novel methods of interaction with large sound databases for sound search by navigation, based on these improvements. To allow an efficient exploration of the space of sounds defined by the sound descriptors, we developed an architecture of the graphic model based on three coordinate spaces (model, world, device coordinate spaces) and on affine transformations or non-linear mapping between spaces (section 4.1). This architecture allows an easy integration of functionalities like zoom & pan and a novel mode *tile* formed by the subdivision of display according to categorial descriptors (section 4.2).

The algorithms and visualisation components are implemented and tested as C-libraries and Max/MSP patches de-

1 . Since its start in 2005, http://freesound.org almost doubled every year to 62701 sounds, 681 hours, 252 GB in Februray 2009.

scribed in section 5. The 229 sound descriptors used in the prototype application are analysed by an external program [9, 10] and loaded from SDIF files [2]. A limited subset of 24 descriptors can also be analysed in the prototype system itself, as described in [17].

2 RELATED WORK

The navigational approach to sound search has been inspired by interactive real-time *corpus-based concatenative synthesis* for musical creation [15] [13, 15, 14] as implemented in the CataRT system [16, 17]. This recent method permits to create music by selecting snippets of a large database of pre-recorded sound by navigating through a two- or higher-dimensional space where each snippet takes up a place according to its sonic character, such as pitch, loudness, brilliance. This allows to use a corpus of sounds as an instrument, or to compose the path through the corpus, or to resynthesise an audio file or live input with the source sounds, and thus to create novel harmonic, melodic and timbral structures. The selected units are concatenated and played, after possibly some transformations. The method can be seen as a content-based extension to granular synthesis providing direct access to specific sound characteristics.

Evidently, the high interactivity of the corpus as an instrument to create music could be immediately applied to the exploration of the sounds in the corpus with the aim of searching sounds. The existing 2D interface had to be extended to allow zoom&pan, and categories and class hierarchies had to be represented.

Related work in Music Information Retrieval start to apply spatial interfaces to content-based audio searches [3, 21], inspired by our work [16, 17], or independently [5, 6], or are concerned with the efficiency of nearest neighbour search [11] or the recent method of *locality-sensitive hashing* (LSH) [20].

3 SCALABLE ALGORITHMS

In order to optimize the exploration of large effects or instrument sound databases, the three algorithms briefly presented here solve recurrent problems in retrieval and visualisation in an efficient and scalable manner. Their functioning, improvements over the state of the art, and evaluation are described in detail in [18].

3.1 Efficient Nearest Neighbour Search with *k*D-Trees

While navigating the database, the problem of finding the sound segment closest to a target point x^t in the multi-dimensional descriptor space is solved efficiently by a *branch and bound* search algorithm based on the tree-structured index provided by the *k*D-tree.

The *k*D-tree represents a hierarchical decomposition of the descriptor space, and during search, whole subtrees are discarded from the search, by application of an elimination rule based on the farthest neighbour found so far. This removes a large amount of the distance calculations between vectors, resulting in a sublinear time complexity. Several variants of the algorithm are compared in [4], and it is argued that the best decomposition is along the hyperplanes orthogonal to the principal components, since it maximises the distance among the points in different subtrees and thus the probability that a subtree can be pruned.

The elimination of nodes is achieved by calculating the *split plane* of a node n defined by an orthogonal vector s_n and going through a point μ_n that is the mean of the node's elements. This plane is used in the vector-to-node distance function based on the dot product:

$$dist_{V2N}(x,n) = (x-\mu_n)/\sigma \cdot s_n \qquad (1)$$

Ideally, s_n is the principal component vector of the node, but choosing it orthogonal to the axis of the dimension with the greatest variability results in an almost equally efficient search with less overhead for the decomposition.

The search algorithm uses a stack of nodes to be visited and the distance d of the target point x^t to the node's split plane in order for the elimination rule to prune child nodes when no vector closer than the current nearest neighbours can be found. An additional radius parameter r limits the returned nearest neighbours to lie within distance r from x^t. If $r = \infty$ all k nearest neighbours are returned. Note that both distance functions dist_{V2N} and the vector-to-vector distance

$$\text{dist}_{V2V}(x,y) = (x-y)/\sigma \qquad (2)$$

can include per-descriptor-weights in σ that balance the influence of each dimension in the search, even after the tree index is built.

The performance measurements in [18] show the logarithmic time complexity of search, linear complexity for building the tree, and the exponential influence of the number of dimensions. However, the highest single search time is just 2.2 ms for a database size of 10^6 10-dimensional points, and an initial overhead over linear search is quickly passed by with data sizes over 100. What's more, using PCA-based decomposition would reduce the dimensionality to the intrinsic number of dimensions, i.e. linearly dependent dimensions would not contribute to the complexity of the search.

3.2 Mass–Spring–Damper–Repulsion Model

Turning to the visualisation of sounds as points in a graphical interface, a useful model is the simulation of a system of masses connected by springs (or, more generally, by links). This model is the heart of the dimensionality reduction algorithm explained in section 3.3, but it can already be applied

to an existing projection on two or three dimensions of a sound database for two purposes: First, it allows to interactively move displayed sounds with neighbouring sounds following, in order to organise the sound space. Second, the repulsion force that has been added in our implementation avoids overlapping points by pushing them apart.

The model is following MSD [7], implemented for MAX/MSP and PUREDATA. We chose an inert model for faster convergence and to avoid self-oscillating systems of masses. For the differential equations of the physics model and their implicit discretisation, see [7].

The available parameters are the nominal length of the links L, the stiffness parameter K, friction damping η, and viscosity damping μ. Repulsion takes place when the mass distance is lower than a threshold L_R and rises linearly up to R. Its effect can be seen in figure 1, where a cluster of overlapping points is distributed in space to reveal all sounds.

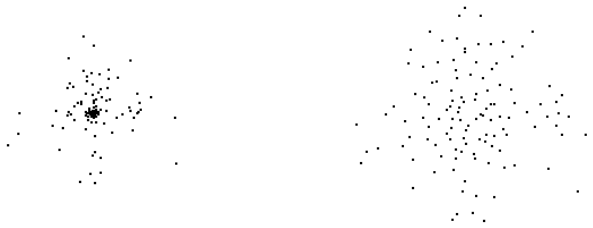

Figure 1. Effect of repulsion (right) on a cluster (left).

3.3 Hybrid Multi-Dimensional Scaling

For the low-dimensional visualisation of a high-dimensional space, the Chalmers algorithm [8] uses a mass–spring model, where the nominal spring lengths are given by the distance in the high-dimensional data space. The basic assumption is that the final minimal stress configuration, that the model will converge to, corresponds to a good layout, where points that are close in data space are also close in layout space. An additional advantage is that the algorithm is iterative such that the current configuration can be displayed to the user while the system converges.

Our improved hybrid algorithm first lays out a random sample of $n_s = \sqrt{N}$ points with a fully-connected mass–spring model to provide a good starting layout for faster convergence, then places the remaining points around their nearest neighbour from data space, taking advantage of the kD-tree, and finally lays all points out by running a mass–spring model with constant numbers of links to the nearest neighbours, and random links that change at each iteration.

The choice of n_s means that each iteration in the initialisation phase is linear, since a fully connected system takes $O(n_s^2) = O(N)$, while the placement is of complexity $O(N \log N)$ and the final iterations constant. Only few iterations are necessary until the total stress reaches a minimum.

4 INTERFACE

The search and musical synthesis of sounds from a large database of sounds and descriptors is similar to the exploration of data manipulating a graphical representation. This task is well described and much research on it has been done in the field of information visualization. Shneiderman and Plaisant [19] define the *mantra of information visualization* as:

Overview, Zoom and Filter, Details on Demand.

Interactive navigation in multi-dimensional data spaces requires either an exploration of individual descriptor dimensions, or a reduction of dimensionality to two or three to allow them being displayed: Methods of dimensionality reduction such as multi-dimensional scaling (MDS), principal component analysis (PCA) with the interactive integration of weights per dimension, linear mapping-by-example, and, if class labels are available, linear discriminant analysis (LDA), can all help to make high dimensional spaces available for interactive navigation.

We will concentrate in the following on new strategies for visualisation of sound databases for efficient search, including the integration of MDS and PCA, zoom & pan, a new display mode *tile*, and filtering options for categories. All these improvements rely on an efficient visualisation architecture that will be described first:

4.1 Visualisation Architecture

The graphics model implements a 3-tier architecture using three separate coordinate spaces (model, world, and device coordinates) and mappings and transformations between them as explained in the following:

4.1.1 Tier 1: Model Coordinates

The model coordinate space is the N-dimensional space of the descriptor data in raw descriptor coordinates (units like Hz, dB, linear amplitude, etc), populated by M units. The choice of D descriptors to use for display takes place in this space, as well as a preselection or inclusion/exclusion of U units to draw.

The model is thus represented as a matrix $model(U,D)$ which is a subset of the original unit data matrix $ud(M,N)$.

4.1.2 Tier 2: World Coordinates

The world coordinate space consists of the descriptor data projected to 2D (or 3D) and a colour scale index, normalised to $[0,1]$. It is represented as the matrix $world(U,D)$ and lists of labels for symbolic descriptors.

The selection of the unit to play being closest to the mouse pointer position takes place in world coordinates, and not in the full descriptor space of *model*.

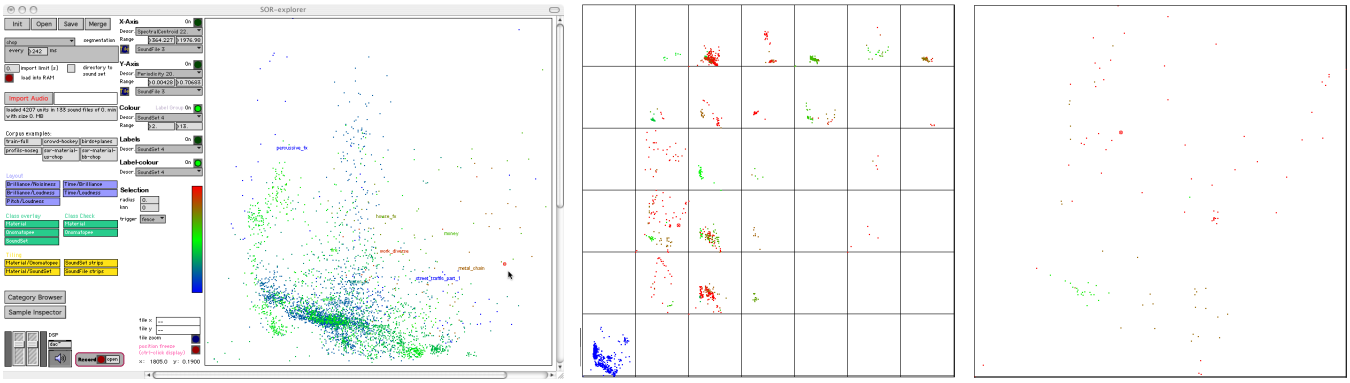

Figure 2. Screenshot of the sound navigator prototype in full view (left), the display canvas in *tile mode* (middle), and *tile zoom* (right), where the tile under the mouse is zoomed in fully.

4.1.3 Tier 3: Device Coordinates

This space is constituted of the raw coordinates of the output device in $device(U,D)$, i.e. screen canvas pixels and RGB colour values, or OpenGL screen coordinates, or other 3D space coordinates.

4.1.4 Mapping A: Model → World

Mapping A is only applied when the descriptor choice or the data changes. Usually, this mapping is a simple projection to the two selected descriptors. More advanced methods of dimensionality reduction, like PCA or MDS can be applied here. The inverse mapping A^{-1} converts a world coordinate $p(p_x,p_y)$ back into model coordinates, i.e. descriptor values.

The different modes of mapping A listed in the following, allow the transformation of the model space to the world coordinate space, or the inclusion of additional dimensions in the visualisation.

Standard Projection
Projection to two descriptor axes d_x,d_y of *model* and a colour scale d_c.

Transformed Projection
Dimensionality reduction by MDS or PCA from a mix of descriptors, d_x = principal component, d_y = secondary component, d_c = tertiary. However, any axis can also be a descriptor.

Tiled View
Can be used to display a view tiled into columns, rows, or cells by a categorical descriptor d^c such as class ID, sound set. Within each cell, a displacement descriptor d^d, scaled to $[0,1]$ is added to the category coordinate.

Pivot View *(not yet implemented)*
Pivot view uses one marked unit as the pivot p, which adds the distance between p and ud as an additional descriptor to choose from. The distance can be derived as a linear combination of descriptor distances, or be given by a distance matrix defined for a categorical descriptor. A further possibility is to place the pivot in the center of the plot, and display the remaining units in polar coordinates, with the distance mapped to the radius, and a selectable descriptor mapped to angle.

4.1.5 Mapping B: World → Device

The mapping from world to device coordinates produces the matrices in device coordinates (pixels and RGB colour values), label positions and colours.

The world-to-device mapping B keeps track of the current *view*: the rectangle $v(ll_x,ll_y,ur_x,ur_y)$ in world coordinates that is displayed (default: $v=(0,0,1,1)$), which is converted to a transformation matrix B including a scale factor s and a displacement vector $t(t_x,t_y)$, such that $device = world \cdot B^T$, or:

$$\begin{pmatrix} x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots \\ x'_n & y'_n & 1 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ t_x & t_y & 1 \end{pmatrix} \quad (3)$$

This results, however, in 7 unused multiplications and 4 unused additions per point $p(x,y)$, and the additional memory for the expansion of the coordinate matrix by one column, so that the direct form is more efficient.

Mapping B provides the following functions that modify the 2D layout of the data points to ease browsing and inspection :

Zoom & Pan: Scaling and moving in the world coordinate space by changing the view v.

Magnifying glass using a sigmoid function mapping which is steepest around the current position.

The inverse mapping B^{-1} maps the input from controllers from device coordinates (pixel position of the mouse pointer) back to world coordinates (taking into account the current view), in order to perform selection.

4.2 Presentation and Interaction

The interface supporting sound search by navigation is shown in figure 2 (left). It allows to choose the descriptors used for the X- and Y-axes of the 2D space, which descriptor is displayed as a label for each sound or group of sounds, and the colour scale of the points representing a sound segment and the labels. Presets exist for the most useful settings of these choices. When moving the mouse pointer through the space, the sound segment close to the current position is immediately played. Playing is layered if movement is fast, so that large parts of the sound space can be explored rapidly. Because the input sounds would typically be segmented in rather short snippets (200–500 ms), the played sounds stop quickly, and thus also long recordings can be inspected by navigation. Other trigger modes exist that continue playing the segments of a given sound when the mouse does not move, in order to hear a recording entirely. Note that the time of a segment in the recording is also part of the selectable descriptors, allowing time-based browsing.

This strong interactivity enables the user to quickly understand the dimensions and areas of the presented space by an initial traversal, probing sound snippets that are played as they are passed by. Zooming in and out of the space and moving the view, together with the layout improvements of the MSDR model (section 3.2), allows to inspect the sound space of continuous descriptors in detail.

In order to explore the category descriptors and classes resulting from automatic classification, or groups of sounds defined by the user, two methods have been implemented: *tile* mode and the category browser.

Figure 2 (middle) shows the display tiled by two categories along the axes. Each tile contains a scaled version of the original descriptor space, but only with the units that are members of both of the two categories. The user can jump between the overall view and the view of a tile zoomed in fully, as illustrated in figure 2 (right).

The *category browser* (figure 3) allows to choose which units are active. To keep the overview of the whole sound distribution, inactive units are still displayed but are greyed out and not selectable. Combinations of categories can either be muted or soloed.

5 IMPLEMENTATION

The algorithms and interfaces described here are implemented as C-libraries and as externals within the FTM&CO extension library [12] at http://ftm.ircam.fr for MAX/MSP and PUREDATA, taking advantage of FTM&CO's advanced

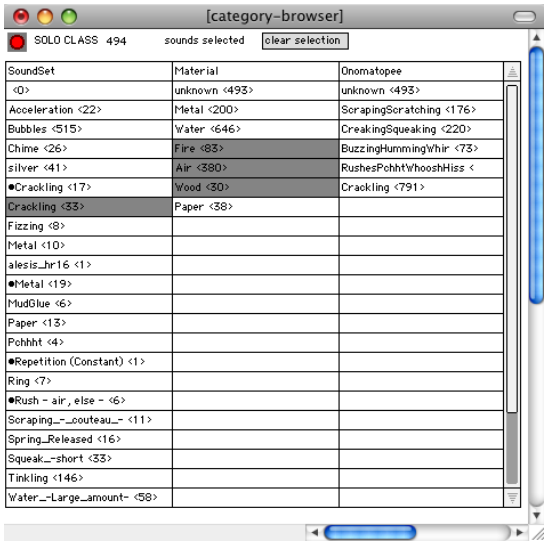

Figure 3. *Category Browser:* choose categories to view.

data structures such as matrices and dictionaries, and the algebraic, mapping, and statistical operators that work on these [1]. This allowed the rapid building of the prototype application to test the search-by-interaction paradigm.

Differing from the CATART system for real-time corpus-based concatenative sound synthesis [17], the sound explorer prototype does not need, and indeed can not keep all the sounds in memory. Instead, they are played back directly from disk, with a latency of about 15 ms for the harddisk access, which does not perturb navigation.

6 CONCLUSION

We described a system for efficient interactive sound search by navigation in databases of sounds and descriptors, based on three algorithms that are crucial for scalability to large databases, their improvements, performance and implementation, and novel concepts and methods for efficient visualisation and interaction in a 2D interface. First, the efficient logarithmic-time kD-tree search algorithm, where we added the limitation to a search radius, and weights for the descriptors. Second, the mass–spring–damper model for intuitive layout optimisation of points in a 2D interface, where we added repulsion. Third, the hybrid multi-dimensional scaling algorithm for dimensionality reduction for visualisation, based on the MSD model, where the use of the kD-tree speeds up the initialisation, allows more precise preplacement, and thus faster convergence.

All three algorithms together make the paradigm of interactive sound search by navigation scalable to very large sound databases.

Then, we described a prototype application based on these algorithms and an efficient visualisation architecture,

that allowed us to experiment a number of innovations and facilities in the user interface, such as class filters and a multi-grid visualisation, to organise search by navigation in large databases.

7 ACKNOWLEDGEMENTS

The research presented here is partially funded by the French National Agency of Research ANR within the RIAM project *Sample Orchestrator*. The authors would like to thank the project partners for their fruitful collaboration, and Joel Bensoam and Bram de Jong for invaluable assistance.

8 REFERENCES

[1] F. Bevilacqua, R. Muller, and N. Schnell, "MnM: a Max/MSP mapping toolbox," in *New Interfaces for Musical Expression*, Vancouver, May 2005, pp. 85–88. [Online]. Available: http://mediatheque.ircam.fr/articles/textes/Bevilacqua05a/

[2] J. J. Burred, C. E. Cella, G. Peeters, A. Röbel, and D. Schwarz, "Using the SDIF sound description interchange format for audio features," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Philadelphia, USA, 2008.

[3] G. Coleman, "Mused: Navigating the personal sample library," in *Proc. ICMC*, Copenhagen, Denmark, 2007.

[4] W. D'haes, D. van Dyck, and X. Rodet, "PCA-based branch and bound search algorithms for computing *K* nearest neighbors," *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1437–1451, 2003.

[5] S. Heise, M. Hlatky, and J. Loviscach, "SoundTorch: Quick browsing in large audio collections," in *AES Convention 125*, San Francisco, CA, USA, Oct. 2008.

[6] ——, "Aurally and visually enhanced audio search with SoundTorch," in *CHI EA '09: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*. Boston, MA, USA: ACM, Apr. 2009, pp. 3241–3246.

[7] N. Montgermont, "Modèles physiques particulaires en environnement temps-réel : Application au contrôle des paramètres de synthèse," MSc Thesis (DEA ATIAM), University of Paris 6, 2005.

[8] A. Morrison and M. Chalmers, "Improving hybrid MDS with pivot-based searching," in *IEEE Symposium on Information Visualization*, 2003, p. 11.

[9] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the Cuidado project," Ircam – Centre Pompidou, Paris, France, Tech. Rep. version 1.0, Apr. 2004. [Online]. Available: http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

[10] G. Peeters and E. Deruty, "Automatic morphological description of sounds," in *Acoustics 08*, Paris, France, Jun. 2008.

[11] P. Roy, J.-J. Aucouturier, F. Pachet, and A. Beurivé, "Exploiting the Tradeoff Between Precision and CPU-time to Speed up Nearest Neighbor Search," in *ISMIR*, London, UK, 2005.

[12] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM—Complex Data Structures for Max," in *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, Sep. 2005.

[13] D. Schwarz, "Data-driven concatenative sound synthesis," Thèse de doctorat, Université Paris 6 – Pierre et Marie Curie, Paris, 2004. [Online]. Available: http://mediatheque.ircam.fr/articles/textes/Schwarz04a

[14] ——, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, Mar. 2006, special Issue on Audio Mosaicing.

[15] ——, "Corpus-based concatenative synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 92–104, Mar. 2007, special Section: Signal Processing for Sound Synthesis.

[16] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Digital Audio Effects (DAFx)*, Montreal, Canada, Sep. 2006.

[17] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based concatenative synthesis," in *Journées d'Informatique Musicale (JIM)*, GMEA, Albi, France, Mar. 2008. [Online]. Available: http://mediatheque.ircam.fr/articles/textes/Schwarz08a

[18] D. Schwarz, N. Schnell, and S. Gulluni, "Scalability in content-based navigation of sound databases," in *Proceedings of the International Computer Music Conference (ICMC)*, Montreal, Canada, Aug. 2009.

[19] B. Shneiderman and C. Plaisant, *Designing the User Interface*. Boston, USA: Pearson, 2005, ch. Information visualization, pp. 580–603.

[20] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 128–131, Mar. 2008.

[21] S. Streich and B. S. Ong, "A music loop explorer system," in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, Northern Ireland, Aug. 2008.