

# MusicalNodes – the visual music library

Lisa Dalhuijsen and Lieven van Velthoven  
Media Technology  
Leiden Institute of Advanced Computer Science  
Leiden University  
Niels Bohrweg 1, 2333 CA Leiden  
The Netherlands  
[lisa@waks.nl](mailto:lisa@waks.nl)   [lievenv@gmail.com](mailto:lievenv@gmail.com)

**Out of dissatisfaction with currently available software, we built a music management program for large digital music libraries. We propose a system that solves some of the difficulties of organizing these libraries. The system introduces a new visual interaction style with the user's music collection. A physics system coupled to album's genre information allows the user to spatially order his or her music on screen and make selections. Also, the graphical appearance of the albums on screen reveals some of their properties, like age or genre, at a single glance. This, combined with an advanced search mechanism and the ability to add custom properties and information to the music, ensures rapid retrieval of music from even the largest of databases. User evaluations confirmed that all functions form welcome improvements over those of well-known text-based alternatives.**

*Data visualisation. Digital music. MP3. Interface design. HCI. Physics simulation. Prägnanz.*

## 1. INTRODUCTION

We are both heavy computer users and music lovers. We listen to music on our computers daily – if not all day – and have quite a big collection of MP3 files. We started this project out of frustration with existing software for playing and managing music. While our collections are ever growing, it is increasingly hard to keep them well-organised. But even in an organised library it might be very difficult to find a specific song. We miss a certain degree of flexibility and creativity in today's well-known music managers. And when software is being used on a daily basis, even minor inconveniences can lead to great dissatisfaction.

We wanted to investigate new ways of how a digital music library looks and behaves, and propose our prototype solution '*MusicalNodes*' in this paper.

## 2. MOTIVATION

Firstly, in our opinion, popular music libraries like Apple's iTunes and Nullsoft's Winamp are very limited in their search and browse functionality. Both rely on a text-based approach, which means users with extensive collections will have to wrestle through huge lists in order to find something (Vignoli, 2004). Of course these programs offer search functions, but the one in Winamp for

example is only useful when you (partially) remember something from the name of the song, album, artist or file path.

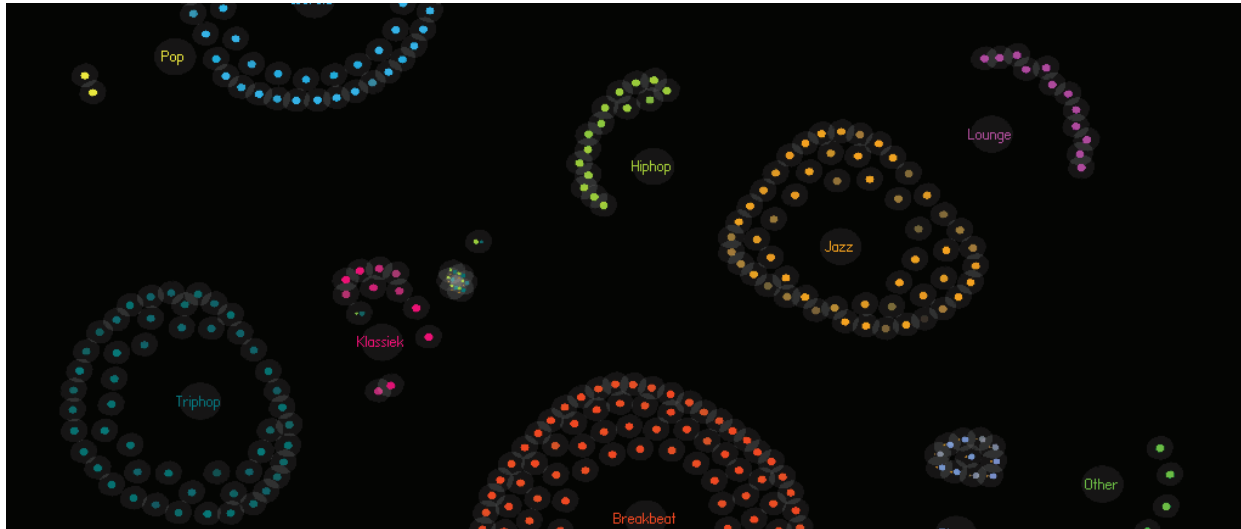
Imagine someone wants to search for a reggae song from before 1980 by an artist whose name was probably 'Bob', but is not too sure about that. Actually he doesn't know the title of the song either. In Winamp, for example, this is almost impossible. He could search for 'Bob', but then may get hundreds of hits, while looking for only one. It can take quite some time to find a particular song in a large library, because search functions offered by existing music players are often not very flexible.

Another big constraint is the way in which digital music is classified into genres. An mp3 file has a so-called ID3 tag containing text that can be edited by anyone. These tags contain information about the music, like the name of the artist and the year in which the music was produced. It can also contain the genre of the piece of music, like 'rock' or 'classical'. Newer versions of the standard are able to carry much more information, but most music players still use the old genres and pre-defined fields. However, our experience is that most music does not fit into one category. Music is often a mixture of different styles, for example funky jazz or electronic reggae. Therefore, we wished it could be possible to categorise music more accurately and allow users to 'do it their way'

(Nilsson, 2000; Aucouturier & Pachet, 2003; Seibel, 2005).

The final main issue is that in available music managers it is hard to get a good overview of the music in the collection. Usually, one can browse by

artist, album, etc., but this results again in huge lists of text, somewhere containing the music you are dying to hear. Since humans are just not very fast at scanning large amounts of text, we feel that there is a lot of room for improvement in this area (Dyson & Haselgrove, 2001).



**Figure 1:** An example of a medium sized music collection

To ensure that we were not the only ones with this opinion, we sent out a questionnaire to people who listen to music on their computers regularly. We asked them how they organise their music, what software they use and how they use it to search for music. It turned out that people like to categorise their music in many different ways, and almost everyone wanted more flexibility in doing this. Some people told us they look for particular albums or songs in their collection, while others look for a particular kind of music – for example relaxing, happy rock songs. This confirms that the standard, pre-defined fields of ID3 tags are just not sufficient, as different people want different ways of finding or browsing their music.

### 3. APPROACH

So, the goal was to build a system that makes it easy to view, search and manage a music database. It was not yet intended as a music player, but rather as a library from which users can select music and send it to their favourite player. In this section we will explain the different parts of the interface, and any novel aspects involved.

#### 3.1 Multiple genres

One of the first problems that we wanted to assess is that music often does not fall into one genre. This can lead to an explosion of different subgenres, making searching or browsing by genre much less useful. A way to counter this is to make the

classification more general, at the cost of losing some information.

In our music manager, we simply allowed a song or album to have *multiple* genres. When the user adds another genre to a piece of music, all he has to do is indicate with a slider how much that music belongs to it from 'not at all' to 'completely'. This way an album could be 50% jazz, 30% funk and 20% electronic while keeping the genre list uncluttered and searchable.

#### 3.2 Custom properties

We also added the ability to create your own categories besides obvious ones like artist or genre. MusicalNodes has a function that lets the user create a property, such as 'record label' or 'heard at ...' and assign a value to it for any song in the database. Numerical properties also exist and are represented by a slider: for example 'happiness / sadness' of the music.

#### 3.3 Visualisation

Since humans are very good at recognizing visual properties and patterns, another important goal was to provide a graphical overview of the music in the database. We believe that a smart way of visualizing the collection can reveal much more information to the user at a single glance than standard text representations (Sternberg, 2003). To achieve this, MusicalNodes' visualisation relies heavily on principles of Gestalt theory; especially

the law of *prägnanz*. This 'law' says that people tend to group visual elements into collections if they have similar properties. These properties can be form, colour, size or brightness, but also spatial configurations and motion patterns. Studies like Mishkin, Ungerleider, and Macko (1983) and Livingstone and Hubel (1988) have shown that many of these cues are processed independently and effortlessly, not just by our brain but even at lower levels of our vision system. This implies that a visual representation of the database can be interpreted much faster than large amounts of text, if it successfully exploits the strengths of human vision.

This also opens up other opportunities. While searching for new ways of organizing music, we realised that people are very much used to organise things spatially (i.e., storing items in a cabinet, cleaning up a desk, etc.). We wanted to allow this behaviour in our software as well and preferably directly on the visualisation. This way, people could for example place their jazz and soul to the left, electronic music to the right, and everything else in between.

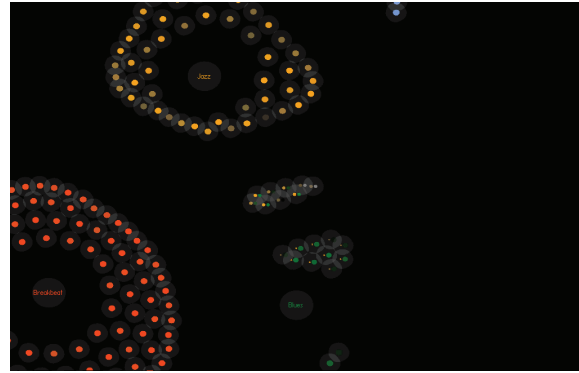
### 3.4 Nodes

The first part of our solution was to represent all music albums by *nodes*: little 'balls' with different properties. These properties visualise various characteristics of the music. Currently, the nodes are built up as follows:

A transparent gray circle that lights up when the node is selected.

- The name and artist of the album on mouse-over.
- One or more coloured dots indicating the album's genres and age.
- When an album has multiple genres, a coloured dot is shown for each genre. The size of the dots indicates the extent to which the music belongs to each of the genres.
- The year in which the music was produced is represented by the dots' colour saturation: the grayer, the older the music is.

Many other aspects of the music might be visualised in the future. For instance, the tempo of the music could be coupled to a 'jumping' movement of the nodes, shape could indicate 'mood', size represent rating, etc.



**Figure 2:** Example of nodes floating in between genres

### 3.5 Spatial organisation

Now that we have nodes to visualise the albums, we need a meaningful way of organizing them on screen. We decided for now to base this on genre information only, although many other properties could be used in the future.

Still, this is easier said than done. We had the desire of not being limited a pre-defined set of genres, which was supported by many people in the initial questionnaires we sent. But since albums can belong to any combination of genres with different ratios, grouping them nicely together on a two-dimensional screen is not trivial. The albums are in multi-dimensional genre space and need to be mapped onto two- or three-dimensional graphics space. Statistics and graph theory provide many approaches to this problem, but they usually do not afford intuitive interaction or manipulation. We wanted to let users organise the visualisation directly, while keeping related music automatically grouped.

### 3.6 Physics

So, to complete the visualisation of the genres, an important innovation in MusicalNodes is the addition of a real-time physics system. Musical genres are also represented by nodes, and attract the albums that belong to them. On the other hand, album nodes push each other away a little bit. This results in clouds of nodes on the screen, where a cloud corresponds to a collection of albums that belong to a specified genre. For example, you could have a jazz cloud, a blues cloud, a hip-hop cloud and so on. An album that belongs to multiple genres is attracted by multiple clouds at the according strengths. This way an album could be categorised as 30% near funk, 60% near jazz and 10% near blues. The Grandmaster Flash node would be at the centre of the hip-hop cloud, while Nina Simone would be floating somewhere between jazz and blues.

The genres can be dragged over the screen and placed wherever the user wants; the albums immediately follow. If a user wants to hear music that is jazzy as well as bluesy, this would be just a matter of seconds: he would drag the two genres away from the rest and simply select the nodes that float between them. This allows on-the-fly creation of playlists according to the user's mood.

In order to keep everything visible as the collection grows, the view can be zoomed and panned in any direction. In the future, different zooming levels might even show different levels of detail; i.e., genres will be visible from far away, albums from closer up and individual songs when fully zoomed in.

### 3.7 Selection and filtering

The final aspect that we wanted to focus on was the searching and selection functionality. The criteria were that it should be both easy to find a specific song, or to make a more general selection for playlists or tag editing.

MusicalNodes has an iTunes-like search function that simply looks for keywords in the entire database. Besides this, we allow users to search in any property of the ID3 tag and file info, or the ones they invented themselves. One can search in numerical properties for exact matches or ranges, and filter text-based properties by whether they contain the query, end with it, match it exactly, etc. The search can be applied to the entire database or just the currently selected (filtered) music, which means that searches can be used sequentially to narrow down search results. A list tracks the filters that are currently in use and allows the user to delete or 'undo' any previous one.

The visualisation and a 'currently selected' list reflect which music is selected. Whenever a filter is added, the albums that do not match are darkened to make the selected ones stand out. This way, searching the collection for American music would tell the user in just a second that most of it is old bluegrass, for example.

Selections can be made in the visualisation, the search function or both. Users can select individual albums or make drag-selections with the mouse. Right-clicking on an album allows you to view its details or find other albums from the same artist.

Once a selection is made, you can assign to it new genres and properties or send it to a playlist. Newly added music automatically receives the 'uncategorised' tag, which helps to identify music that may need your attention.

## 4. RESULTS AND EVALUATION

The graphical overview of the music succeeds at giving insight into the whole collection at a single glance. This is not only faster in many cases than having to scroll through lists, it is also much more inviting. (Re-) exploring your music database becomes as easy as saying 'anything that contains jazz' and seeing what genres and ages pop up.

It is very easy to focus on certain groups of albums as a whole, like albums that have the same genre or are from the same era in time. The idea of exploiting prägnanz principles works very well for this application, and we plan on adding other visual properties to the nodes.

Another main advantage of the visual representation is that it enables direct manipulation of the data. This is useful for building genre-based playlists or just organizing the music spatially. While manipulating a genre-nodes' location, the mind instantly recognises the albums belonging to that genre by their colour and movement.

The physics system works remarkably well as an interactive clustering method and produces perfectly separated genre clouds, no matter where the user places them on screen. The behaviour of albums with multiple genres could be further improved upon in this respect, although it works sufficiently already.

We found similar work has been done, for example that of Vignoli and Pauws (2005). Our system has a different approach in a number of ways, one of which is that it does not assume that the geometrical dimensions have any meaning. It just keeps the clusters separated. The user is free to arrange them, and we rely and prägnanz laws to convey similarities.

Sadly, there currently is no specific order *within* a cloud of same-genre albums. By linking additional 'forces' to other properties this could easily be changed.

Unintendedly, the physics system provides the software with an attractive, highly animated look when the user moves his or her music around.

### 4.1 User evaluations

During the design process we tested our prototype twice on users, and another time after it was finished. The first thing we wanted to test was the usability of the search function. We consider this a very important part of the program and wanted to make sure that it works quickly and easily, even though it is more complex than, say, iTunes's. Without any instructions on how to use the

program, we asked 17 test users to perform 10 different search tasks defined by us. Most test persons seemed to learn how the filter function works quite naturally, also for more advanced searches. The average search time converged already after the fourth task from 13 to around 5 seconds. Also, the idea of sequentially adding filters or removing them from the filter list proved intuitive. Across the board, they stated that they would really use the extended search functionality.

All test users were positive about the visual aspect of the program. They were convinced that it offers a useful advantage over conventional interfaces. Besides that, they quite enjoyed playing around with the nodes on screen.

## 5. CONCLUSION

We believe that with MusicalNodes we have a great proof of concept for the ideas introduced here, which form a novel interaction paradigm. The key aspects of the program; the physics-based visualisation and direct manipulation of the music collection, the extensive search function and the custom properties all add up to form a powerful music library. Intended users who tested the system found it quite useful and welcome and appreciated the novel aspects it features. The result is a piece of software that is easy to learn for beginners while at the same time flexible and powerful for more advanced users.

## 6. FUTURE WORK

We happily realise that this is just an early prototype. Some features could be improved upon and many ideas still wait to be implemented or tried out. Below is a selection of some of these ideas.

- Assigning even more properties of the music to the appearance of the nodes.
- Adding a hierarchical 'level of detail' to the zoom function
- Adding the possibility to arrange the nodes on screen differently. Instead of the genre-clouds, arrange the nodes by record label, happiness or otherwise.
- A world map view to show countries of origin.
- Linking subgenres to main genre-nodes, so that dragging a genre around will result in the related subgenres following.
- Integration with existing music player software or inclusion of a built-in one.
- A multi-column text view of the collection with search boxes and sort buttons for each property column.

- Generate order within a cloud of same-genre albums.
- Improve the behaviour of albums that are connected to multiple genres for even better clustering.
- Free-hand 'lasso selections' in the visualisation.
- And many more ideas about the user interface.

Visit <http://www.musicalnodes.com> for a demonstration of the software.

## 7. REFERENCES

- Aucouturier, J.J. and Pachet, F. (2003) Representing musical genre: A state of the art. *Journal of New Music Research*, 32 (1), pp. 83–93.
- Bartram, L. (1997) *Perceptual and interpretative properties of motion for information visualization*. Technical Report CMPT-TR:1997-15, School of Computing Science, Simon Fraser University.
- Dyson, M.C. and Haselgrove, M. (2001) The influence of reading speed and line length on the effectiveness of reading from screen. *International Journal of Human-Computer Studies*, 54, pp. 585–612.
- Livingstone, M. and Hubel, D. (1988) Segregation of form, color, movement and depth: Anatomy, physiology, and perception. *Science*, 240, pp. 740–749.
- Mishkin, M., Ungerleider, L.G.. and Macko, K.A. (1983) Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6, pp. 414–417.
- Nilsson, M. (2000) ID3 tag version 2.4.0 – Main Structure. <http://www.id3.org/id3v2.4.0-structure> (1 April 2010).
- Seibel, P. (2005) *Practical COMMON LISP*. Apress, Berkeley, CA. pp. 335–336.
- Sternberg, R.J. (2003) *Cognitive Psychology*, 5<sup>th</sup> edition. Wadsworth, Belmont, CA.
- Vignoli, F. (2004) Digital music interaction concepts: A user study. In *Proc. 5th International Conference on Music Information Retrieval*. Barcelona, Spain.
- Vignoli, F. and Pauws, S. (2005) A music retrieval system based on user-driver similarity and its evaluation. In *Proc. 6th International Conference on Music Information Retrieval*. London, UK.