

Musician Map: visualizing music collaborations over time

Ji-Dong Yim

Chris Shaw

Lyn Bartram

School of Interactive Art and Technology
Simon Fraser University
Surrey, BC, CANADA
{jdyim,shaw,lyn} at sfu dot ca

In this paper we introduce MusicianMap, a web-based interactive tool for visualizing relationships among popular musicians who have released recordings since 1950. MusicianMap accepts search terms from the user, and in turn uses these terms to retrieve data from *MusicBrainz.org* and *AudioScrobbler.net*, and visualizes the results. MusicianMap visualizes relationships of various kinds between music groups and individual musicians, such as band membership, musical collaborations, and linkage to other artists that are generally regarded as being similar in musical style. These relationships are plotted between artists using a new timeline-based visualization where a node in a traditional node-link diagram has been transformed into a TimelineNode, which allows the visualization of an evolving entity over time, such as the membership in a band. This allows the user to pursue social trend queries such as “Do Hip-Hop artists collaborate differently than Rock artists”.

1 INTRODUCTION

In this paper we introduce MusicianMap, a web-based interactive tool for visualizing relationships among popular musical artists across time and band membership. Our data is drawn from the popular online databases *MusicBrainz.org* [24] and *AudioScrobbler.net* [3]. MusicBrainz is a user-maintained community database of *metadata* about music recordings. Music metadata is information such as the artist name, the release title, and the list of tracks that appear on a release. The AudioScrobbler system is a database that tracks listening habits and uses statistical reasoning to generate information about *similarity* of musical genre and taste. Thus these sources contain data about popular music recordings from around 1950 onward, but do not explicitly provide information about collaborations between artists or about the progression of their careers and how they intersect across types of music and time. The goal of the tool described in this paper was the exploration of these artistic collaborations: a social network of artists and groups [19] derived from their products (i.e. recordings) and mapped onto dimensions of *similarity* (type of music), *association* (membership in a group, if the group is the primary artistic unit) and *time* (progression of artist career and group longevity). The challenge we face is that the kinds of visualization tools or techniques typically applied to these types of information are quite visually disparate and simply combining them results in displays of unusable complexity.

The types of data we are interested in examining are as follows: the individual membership in a band; the recordings released; the time period

when the band was active; and the musicians who have contributed their efforts to the band’s work. Queries on this data can range from the simple to the complex. A simple database query can answer a question such as: “When was *Exile on Main Street* released by the Rolling Stones?”, for example.

However, other queries are somewhat more difficult to answer with a simple request for a single value: “What was the most productive period of the Rolling Stones?”, or “Do Hip-Hop artists collaborate differently than Rock artists?”. These longitudinal queries require a more in-depth analysis of a musician’s *social network* of collaborations over time, and do not admit a simple query or a simple aggregate measure such as mean or standard deviation. MusicianMap is aimed at providing support for these more wide-ranging queries about social trends that can be observed in data about popular musicians. To support this more longitudinal type of query, we have developed a new timeline visualization technique called a TimelineNode that allows the evolution of an entity over time to be examined in the context of node-link network visualization.

2 RELATED WORK

2.1 Visualizing music relationships

Online music shops and music web services have functions for convenient browsing and discovery of new music [2]. For example, *Last.fm* (the “consumer side” of *AudioScrobbler.net*) provides a music recommendation system that retrieves data on the basis of similar artists, tags and play-lists made by other anonymous users’ ratings [3], but simply provides text lists in a general HTML/website

format.

A number of other tools have explored the use of scatterplots and node-link graphs to represent the connections between recordings or songs, such as [4][5][8]. These techniques typically use this approach to show similarity along some aspect, critical for music recommender systems. Crampes et al. developed a concept mapping technique for visualizing individual recordings from a musical artist [4]. They also described a technique for managing interaction with the various abstract representations that supported their music visualization. Donaldson and Knopke developed a music recommendation system that displays a scatterplot of songs that have been selected out of a database of user's playlists [5]. Songs are viewed as being more related the more often they appear together in user's playlists. The authors apply multidimensional scaling to the matrix of song-to-song links to generate a 2D cluster scatterplot of song relatedness. Scholz has developed a javascript-based visualization of musician data that plots recordings as nodes and edges as relationships between recordings, such as the same song in each recording, or the same person in two groups, among other relationships [8]. Yensen's Hubmed system displays node-link diagrams that display AudioScrobbler data, where nodes are music acts and links are derived from AudioScrobbler's similarity data [10]. Musicplasma has a similar search facility that plots bands as nodes, and uses link color to indicate the kind of relationship between entities, such as individual membership or similarity [11].

Another approach to the task of recommending new music is to analyze the music itself (see Scaringella [12] for a recent survey). For example, Lamere and Eck use music similarity functions to locate nearby music to the user's interest and plot similar items in 3D [7]. The DataBionic Music miner [13] classifies music using a self-organizing map and plots music using a fanciful geographic metaphor that the user may browse and use to compose playlists.

The task that these systems are aimed at supporting is the recommendation of music that is similar to the music that has been entered as a search term. This task has an obvious commercial impact in that a system will presumably prompt a user to buy music by enabling the user to find more music which he/she may like.

However, the kinds of tasks we are interested in supporting involve the exploration and discovery of aspects of the *social environment* of commercial music through the use of music metadata. In addition to the more-like-this question, music metadata can contain sufficient information to analyze the careers

of individual musicians, and through the sampling of metadata, can help broader queries like "Do Hip-Hop artists collaborate differently than Rock artists?" Gleiser begins to explore similar questions of collaboration through his visualization of the community of jazz artists (Figure 1), where each node represents a musician, and each link represents a shared band membership [19]. Rather than explore the similarity of music or artists from the perspective of choosing something related, however, Gleiser's work explores the social environment of jazz musicians and is one example of a *social network*.

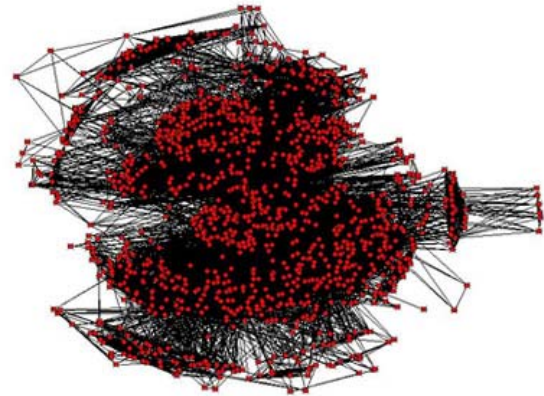


Figure 1. Visualization of the community structure of jazz musicians [19]

2.2 Social network visualization

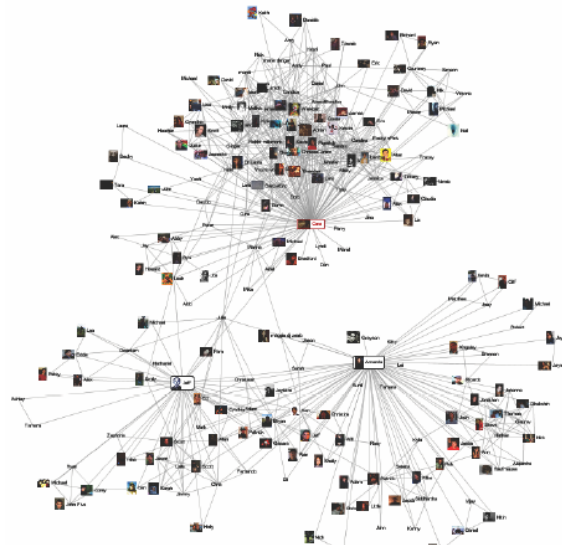


Figure 2. Vizster social network graph [18].

Many of the standard approaches to depicting some form of relation are based on the traditional node-link graph and are more generally used in the domain of social network visualization. While there are many variants of social network visualization (the

reader is directed to [20] for a selection of good examples), one of the best-known tools is Vizster [18]. Vizster uses a spring-embedded graph layout. The visualization in Figure 2 depicts relationships between individuals and the resulting visual clusters make it easy to discover three intersecting social networks.

These types of diagrams facilitate the discovery of relationships and are well-suited to determining *how* elements may be related, but they provide no insight into *when* and for *how long* these relationships might have arisen. Their layouts do not afford a linear view of progress over time.

2.3 Time-based visualization

Time is most commonly used in visualizations as a horizontal axis, or timeline, on which events and intervals are plotted (Figure 3).

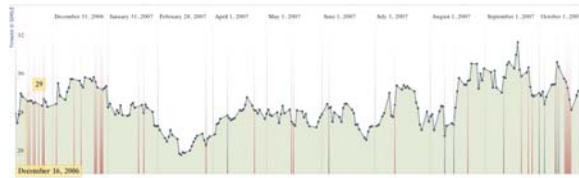


Figure 3. Common timeline view[17]

There have been several approaches to depicting how elements are related in time-based views. The TimeLine widget (Figure 5) uses bars and colour clusters to show a drilled-down or exploded view of individual events. LifeLines [22] stacks timelines to cluster facets of a patient's medical history, plotting both extended periods and discrete events (Figure 4).

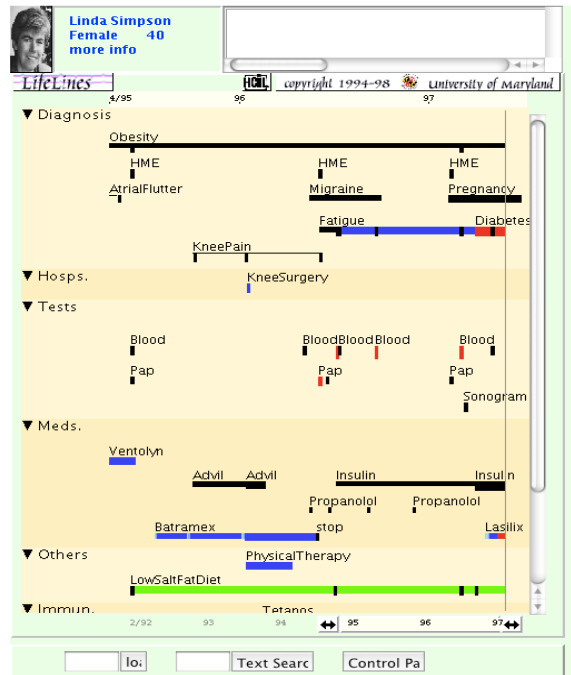


Figure 4. LifeLines

Phan et al. [21] use a series of small timelines stacked both horizontally and vertically in a table-like view to enable comparison of patterns and events in a network monitoring task. This approach is based on the design guideline of small multiples and enables a timeline as a pivot from which further, more detailed timeline views are generated.

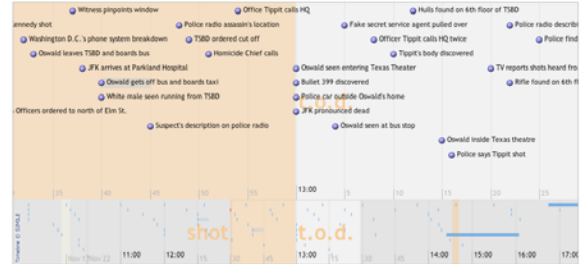


Figure 5. Time clusters [16]

As can be seen, however, these kinds of time-based views make it difficult to ascertain the kinds of visual clusters that show similarity or association that is not time-based. Approaches include using colour and shape to indicate association (as in LifeLines [22]) but it is difficult to see how one might include the kinds of social networking views previously discussed. Current work in knowledge visualization is exploring these questions. Chen [23] has been plotting the histories of document co-citations and how they cluster. In their work on visualizing the evolution of research, Morris et al. [15] plotted clusters of document citations on stacked timelines of research topics and drew links between documents that had similar citations or cited each other (Figure 6).

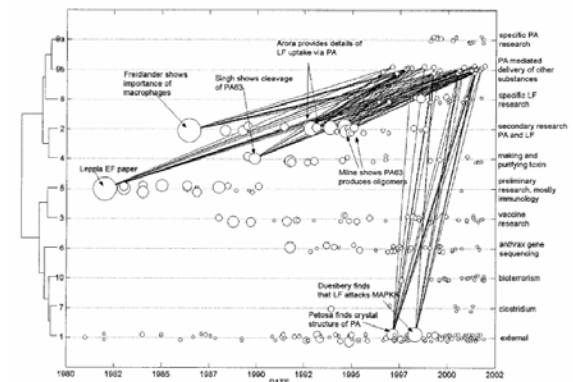


Figure 6. Graphing across timelines [15]

3 MUSICIAN MAP

The purpose of this research is to explore various possibilities with which entities and relationships can be dynamically expressed using scaled node-link diagrams. To address this, our proposed application should satisfy the following requirements;

- 1) A Web based search application that enables

limited, directed retrieval of data to a manageable subset of the entire database. The number of entities displayed should be on the order of tens to hundreds.

2) An interactive application for viewing node-link diagrams that facilitates user interactions with various graphical codes such as shape, color, size, transparency etc.

3) An informative application that supports the distinction of various types of relationships between the displayed musical entities as possible.

4) An assistant application that leads users to discover new information by browsing and searching the underlying dataset.

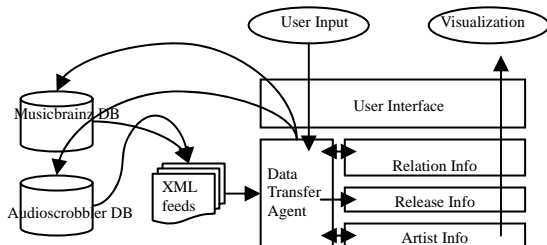


Figure 7. MusicianMap architecture

Our resulting application, MusicianMap, is a web-based application that visualizes a subset of the MusicBrainz and AudioScrobbler musician metadata databases. It receives an artist's name as a keyword from a user, generates and executes a series of queries, and visualizes the search result. The program is written in Java using the prefuse visualization toolkit [25], and consists of three main parts; User interface, data transfer agent and artist relationship framework [Figure 7].

When a search process is triggered by a user, the program first requests an artist ID to be searched. The artist ID is unique and commonly shared in the MusicBrainz's and the AudioScrobbler's databases. Then related query strings are generated for a given artist ID to retrieve general information of an artist such as name, ID, type, date of birth/death, albums, related artists and similar artists. The retrieval data forms up an entity object of an artist.

Once the primary artist query has been satisfied, the framework filters out unnecessary fields and returns a list of artist IDs abstracted from related and similar artists to be queried successively. The system puts a depth limit in place for each search queries to manager the data in a reasonable amount that can balance the number of entities and the computing performance.

3.1 User Interface

The user interface consists of the visualization area and the control panel. The visualization area on

the left shows the search result with two different views, node-link and timeline graphs, where the visualization is gradually formed as query results arrive from the remote database server. In this area, a user can emphasize a graphical object by placing the mouse cursor over a node or a link. For example, when an artist node is focused, it pops up with the nodes and links which are strongly related to the artist and other less related entities are dimmed out. Similarly, when the cursor is over a link object, it shows its relationship properties with two linked artist popped up. A user also can drag a node to a different location to draw a more recognizable layout.

The control panel is located in the right side of the application and includes search box, timeline toggle, filters, legend, albums and related artists list. The search box is currently designed to search only the Artist field of MusicBrainz data. When a user inserts a keyword and hits the Search button, the program automatically seeks for the artist whose name appears first in the MusicBrainz's recommendation list. The Timeline button toggles the view of visualization result between node-link graph and timeline based distribution.

3.2 Data Transfer Agent

The data transfer agent is based on the HTTP Dom class object in Java. The main role of this module is to manage the list of queries to be sent to remote databases and to convey the XML feeds to the artist relationship framework. There are four types of queries used to formulate a single artist node-links set. The first three queries are sent to MusicBrainz: One converts an artist name into an artist ID, the next uses this ID to retrieve general information about the artist (date of birth, collaboration history, etc), and the third retrieves the artist's release list. The fourth query is sent to AudioScrobbler to retrieve a list of artists that are similar to the one described by the artist name.

3.3 Artist Relationship Framework

The artist relationship framework consists of artist nodes and relationship links. The artist nodes have three different types; person (solo musician), group (band) and unknown (not loaded). An artist node has multiple child nodes, which are the artist's releases, set from the list of their official albums. Each album has information about its sales, and the sum of the sales values is used to determine the size of the parent artist node in the map.

For groups, there are a number of relationships existing between the group and individual musicians. One of the most significant relationships in the map is the *member of band* attribute. Band memberships exist over some fraction of the lifetime of a band, and

are the closest relationship between a band and musicians.

Next in significance is the *collaboration* between a pair of entities (between individual artists, between bands, or between artist and group). In practice, collaborations are of shorter duration than band membership.

The third type of relationship is *supporting musician*, which usually denotes a shorter-term relationship than collaboration, and typically means that the artist/group has hired the supporting musician for his/her services.

Finally, the *similar artist* relationship attribute indicates that anonymous database users have drawn an association between the pair of artists from their subjective opinion, whereas the first three attributes are facts. Similarity has a weight factor from 0 to 1 which is the identity between two objects made by ratings from numerous anonymous users on the web. According to the ratings, a relationship link pulls together or pushes apart two artist nodes to distribute them in a map.

4 VISUALIZATION

As MusicianMap gathers data in response to the query, it incrementally populates the node-link diagram with the new data as it arrives. This incremental population arises because of the query-rate limitations enforced by the data providers, but it has the advantage of allowing the user to see how the picture of related musicians evolve as the query list gets fulfilled.

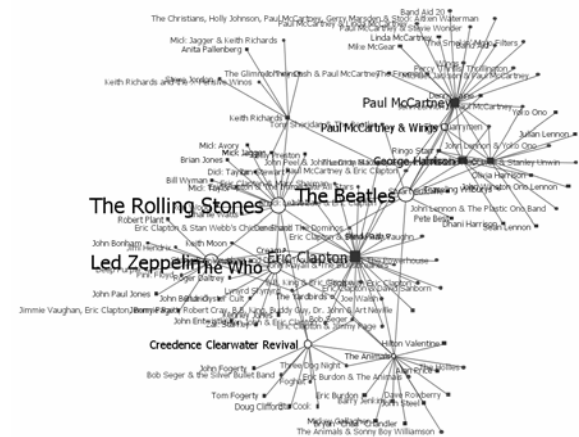


Figure 8: Node-link diagram of a number of artists. Individual people are shown with black square nodes, bands are hollow circles, and unknown entities are displayed with gray circles.

Nodes are laid out by a force directed layout where each type of artists is rendered differently; A person node appears as a dark rectangle, a group is

drawn with a hollow circle, and an unknown entity is displayed as a transparent gray circle [Figure 8]. Node distance is computed from the scalar measure of relatedness outlined in the previous section. The shorter the distance is, the more important or the stronger relationship we can expect. The visualization is continuously animated until the blending function yields no location change.

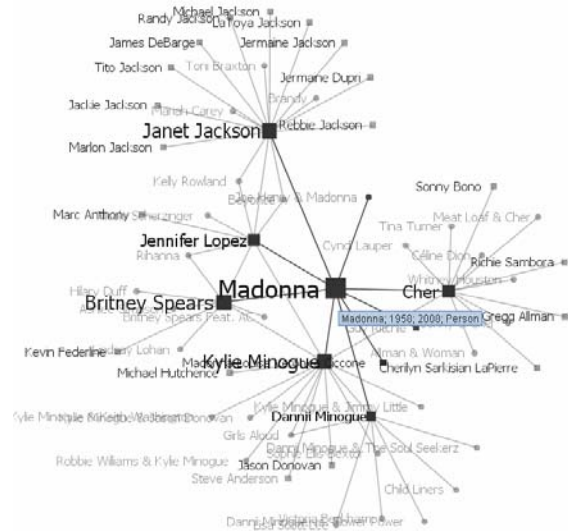


Figure 9. Madonna and related musicians.

Figure 9 shows the visualization result to the keyword “Madonna.” As the data loads, the user can track the target artist and observe how the relationships are linked in the musician map. A user can drag artist nodes to customize the map into a new layout, so that a relationship he/she is especially interested in can be illustrated in a more easily recognizable. In the graph, Madonna is emphasized by dimming out unrelated nodes and edges, and the edges are colored to show different types of relationships. Madonna is rated by the public as being somewhat similar to Cher and Kyle Minogue, and somehow related but less similar to Britney Spears, Janet Jackson and Jennifer Lopez.

To manage the amount of details being presented at any moment, user-selected nodes and their linked entities are expanded, and their nodes and edges are drawn in more intense color. All of the primary links from an artist are shown fully saturated, while more details are drawn of the selected artist. Artist details such as official releases and the list of related artists are shown in the control panel to the right of the visualization window.

4.1 Timeline View

The standard node-link view has the usual

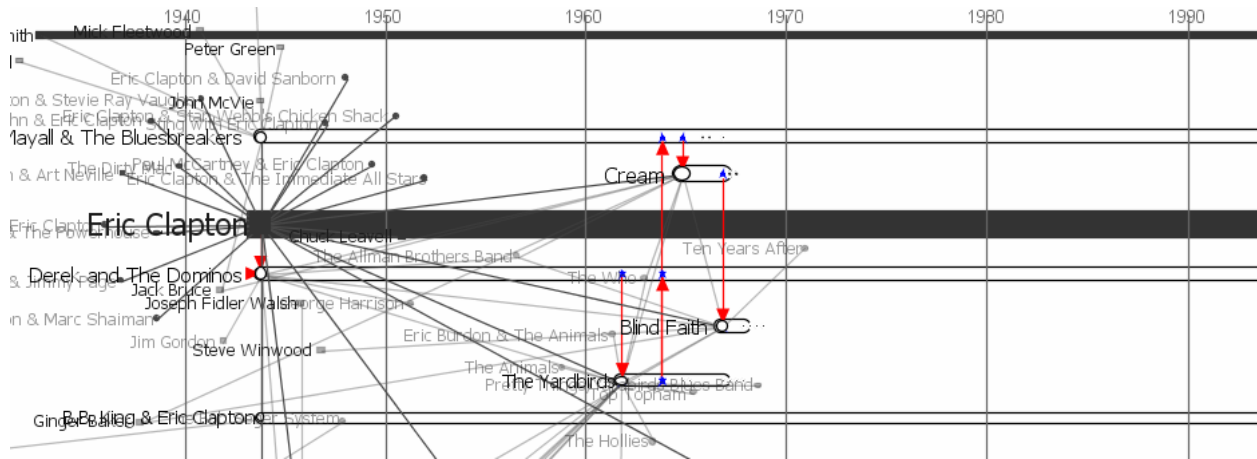


Figure 10. Eric Clapton's career. The red links up and down show the period when Clapton joined the Yardbirds in 1963, left to join Cream in 1966, and left Cream to join Blind Faith in 1968. At the beginning of the Derek and The Dominos timeline, we can see there is a data error since Clapton joined the band and left when he was born.

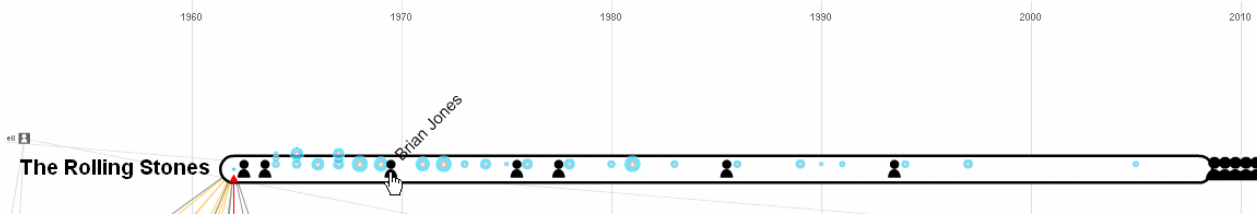


Figure. 11. Query = Rolling Stones

advantages and disadvantages of such visualizations. There is a limited amount of space to draw further information in a small node icon, so it is not obvious how the artist collaborated and have been related through their careers.

To enable more information to be drawn about the lifetime of an artist, we developed the timeline view, which extends the node-link idea by stretching out an artist node along a timeline of activity. The trade-off of this view is that there are 2 degrees of location freedom in a standard force-directed layout, whereas, in the timeline view, the horizontal dimension has been dedicated to the timeline.

The core rendering technique for the timeline view is the `TimelineNode`, which is an aggregate node that shows events of a musician or group along the time line. The primary piece of information is the lifetime of the group or individual that the node represents. Along this line is plotted record releases, and in the case of groups, members arriving or departing from the group. When an individual is selected, that person's joining and leaving events for a group are drawn in red as shown in the Eric Clapton visualization (Figure 10). It is now easier to understand his journey among 1960s' famous rock bands. He started his music with The Yardbirds and

move to the John Mayall & the Bluesbreakers, Creams, Blind Faith and so on. He had a big hit album in the Blind Faith in 1969, and has been working alone for almost 40 years until now.

The visual language of nodes and links is similar to the one shown in the regular node-link view, where the selected node is drawn in more intense colors and the secondary nodes are grayed out. Layout uses a similar pseudo-force algorithm, where the vertical distance is the only degree of layout freedom. In this case, we needed to increase the "opposing force" that would keep away items from each other in order to reduce overlap between items. The greater horizontal extent of all items makes overdraw more likely.

4.2 Sample Questions

Do hip-hop artists collaborate differently than Rock artists?

Figure 12 shows a visualization that sheds some light on this question. The image shows the search term "Run-DMC" the number of band entities shown in this visualization is 13, while there are 12 individuals who are similar to or collaborated with Run-DMC. Investigation of these individuals (*JaRule*,

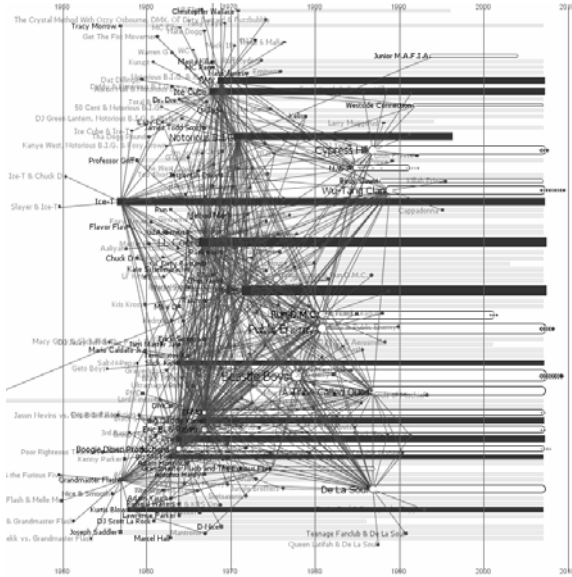


Figure 12. Query = Run-DMC Many solo acts, few bands in the Hip-Hop genre.

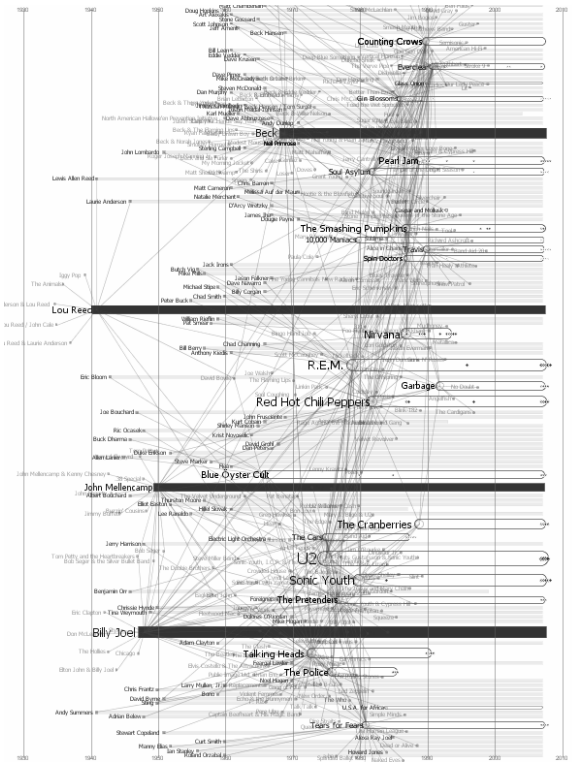


Figure 13. Query = R.E.M. Rock musicians have comparatively more groups and fewer solo acts

Big Daddy Kane, Snoop Dogg, LL Cool J, Busta Rhymes, etc al.) reveals that many of these artists have each released a between two and ten recordings each. By comparison, the search term “REM” (a band roughly contemporary with Run-DMC) yields about

25 bands but only 4 solo artists (*John Mellencamp, Lou Reed, Billy Joel and Beck*).

On the other hand, there is a much stronger set of collaborative links in the Hip-Hop genre. It seems evident from these displays that Hip-Hop is an individual art form that relies on external collaborations, while Rock relies more on a fixed set of personnel.

What was the most productive period of the Rolling Stones? Rock historians will suggest that the albums released between 1968 and 1972 were the Rolling Stones’ best, and the visualization seems to bear this out (Figure 11). The group had significant success during the 1960’s but the height of their commercial success in terms of record sales occurred for the records released between 1968 and 1972. In the visualization, Brian Jones is highlighted. He left the band in 1969 and died shortly thereafter. Since Jones was with the band since around its founding, one could speculate that this had some sort of impact on the remaining band members.

5 Conclusions

We have introduced a new system for analyzing relationships among musicians and musical groups by visualizing linkage relationships among metadata about musicians. Our web-based system *MusicianMap* enables search-based query on the MusicBrainz and AudioScrobbler databases, and develops an on-the-fly visualization of these data in two views. The first is a standard node-link view that lays out Artists according to similarity links of various levels. The second is the Timeline view, which extends the notion of a node to include time-based information about an artist, such as recordings issued, quantity of sales, and for groups shifts in personnel over the years. This visualization allows the user to examine how artists have evolved and changed over the years, and allows these to be viewed in context with other similar artists.

The combination of features allows a user to visualize longitudinal social queries such as the differing ways in which various genres work, and the way that lineup changes have affected a band’s output.

The reader can try this visualization system at:

<http://www.sfu.ca/~jdvim/musicianMap/>

6 REFERENCES

- [1] Ware, C. Information Visualization – Perception for Design, 2nd Edition, Elsevier Inc. (2004).
- [2] Pohle, T., Knees, P., Schedl, M., and Widmer, G. “Meaningfully Browsing Music Services”, In Proc of The International Conferences on Music

- Information Retrieval and Related Activities (ISMIR 2007), 23-30 September 2007, Vienna (Austria), 115-116
- [3] Last.fm Ltd., <http://www.last.fm/>
- [4] Crampes, M., Ranwez, S., Villerd, J., Velickovski, F., Mooney, C., Emery, A., and Mille, N. "Concept Maps for Designing Adaptive Knowledge Maps", *Information Visualization* (2006) 5, 211-224
- [5] Donaldson, J. and Knopke, I. "Music Recommendation Mapping and Interface based on Structural Network Entropy", In Proc of The International Conferences on Music Information Retrieval and Related Activities (ISMIR 2007), 23-30 September 2007, Vienna (Austria), 181-182
- [6] SunMicrosystems, <http://research.sun.com/projects/dashboard.php?id=153>
- [7] P. Lamere and D. Eck. "Using 3d visualizations to explore and discover music". In Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007), 2007.
- [8] <http://kylescholz.com/projects/speaking/tae2006/music/#B00005QXXO>
- [9] <http://www.hubmed.org/touchgraphs/audioscrobbler.php>
- [10] Yensen, J.A.P. (June, 2005). Editorial: Almost Everything About HubMed. *Online Journal of Nursing Informatics* (OJNI),
- [11] <http://www.musicplasma.com/>
- [12] N. Scaringella, G. Zoia, D. Mlynek, "Automatic genre classification of music content: a survey", *IEEE Signal Processing Magazine*, March 2006, 23(2):133-141.
- [13] Moerchen, F., Ultsch, Noecker, M., Stamm, C.: Databionic visualization of music collections according to perceptual distance, Proceedings 6th International Conference on Music Information Retrieval (ISMIR 2005), London, UK, pp. 396-403
- [14] Xu, Changsheng; Maddage, N.C.; Xi Shao; Fang Cao; Qi Tian, "Musical genre classification using support vector machines", *IEEE Intl Conf on Acoustics, Speech, and Signal Processing*, 2003. Volume 5, Issue , 6-10 April 2003 Page(s): V - 429-32 vol.5.
- [15] Morris, S. A., Yen, G., Wu, Z., and Asnake, B. 2003. Time line visualization of research fronts. *J. Am. Soc. Inf. Sci. Technol.* 54, 5 (Mar. 2003), 413-422.
- [16] Timeline Project, MIT. <http://simile.mit.edu/timeline/>. 2006.
- [17] Laconic Security. TJ Maxx Timeline Visualization. www.laconicsecurity.org/tj-maxx-timeline-visualization.html. 2007.
- [18] Heer, J. and Boyd, D. Vizster: Visualizing Online Social Networks. *Proceedings of IEEE Information Visualization*, 2005.
- [19] Gleiser, P. Community Structure in Jazz. http://www.visualcomplexity.com/vc/project_details.cfm?id=115&index=22&domain=Social%20Networks. 2007
- [20] www.visualcomplexity.com/vc/index.cfm?domain=Social%20Networks
- [21] Phan, D; Paepcke, A.; Winograd, T. Progressive Multiples for Communication-Minded Visualization, *Proceedings of Graphics Interface 2007*.
- [22] Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B. Life Lines: Visualizing personal histories. *ACM CHI '96 Conference Proc.*, 221-227. 1996
- [23] Chen, C. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3), 359-377. 2006.
- [24] MetaBrainz Foundation, <http://musicbrainz.org/>
- [25] Heer, J., S.K. Card, J.A. Landay. prefuse: A Toolkit for Interactive Information Visualization. *CHI 2005, Portland, OR*, 421-430.