GEOSHUFFLE: LOCATION-AWARE, CONTENT-BASED MUSIC BROWSING USING SELF-ORGANIZING TAG CLOUDS

Scott Miller, Paul Reimer
University of Victoria
Electrical and Computer Engineering
smiller@ece.uvic.ca

Steven Ness, George Tzanetakis
University of Victoria
Computer Science
gtzan@cs.uvic.ca

ABSTRACT

In the past few years the computational capabilities of mobile phones have been constantly increasing. Frequently these smartphones are also used as portable music players. In this paper we describe GeoShuffle – a prototype system for content-based music browsing and exploration that targets such devices. One of the most interesting aspects of these portable devices is the inclusion of positioning capabilities based on GPS. GeoShuffle adds location-based and time-based context to a user's listening preferences. Playlists are dynamically generated based on the location of the user, path and historical preferences.

Browsing large music collections having thousands of tracks is challenging. The most common method of interaction is using long lists of textual metadata such as artist name or genre. Current smartphones are characterized by small screen real-estate which limits the amount of textual information that can be displayed. We propose self-organizing tag clouds, a 2D tag cloud representation that is based on an underlying self-organizing map calculated using automatically extracted audio features. To evalute the system the Magnatagatune database is utilized. The evaluation indicates that location and time context can improve the quality of music recommendation and that self-organizing tag clouds provide faster browsing and are more engaging than text-based tag clouds.

1. INTRODUCTION

Portable mobile phones with strong multimedia capabilities and computational power are rapidly gaining popularity. As these devices frequently also function as portable digital music players it is important to investigate how music information retrieval systems can be adapted to the unique challenges and opportunities they present. In this paper we describe GeoShuffle a music browsing application designed to address the challenge of limited screen real estate and to take advantage of the opportunity of location information that smart phones provide.

Automatic music recommendation is an active topic of research. Such systems can be based on collaborative filtering, expert annotations, folksonomies, automatic content analysis and any of their combinations. However, all these approaches suffer from the limitation that their results are the same irrespective of the listening context. The preferences of a listener change depending on where they are and what they are doing. For example the music a student would like recommended when studying might be different from the music desired when riding the bus.

Location-aware devices based on technologies such as GPS are common. We propose that the quality of automatically generated playlists can be improved by taking into account this newly available location data. This information can be used to determine a user's listening habits while in transit to common destinations, as people often have daily routines such as return trips to work, school, social activities, and so on. It provides context to a user's listening preferences beyond general ratings. A user providing a rating to a song does not provide context about the conditions under which a user would enjoys listening to that song. For example, a high-energy song that a user rates highly may never be desired when the user wants to relax.

Another unique characteristic of smart phones is their limited screen real-estate. The size of personal digital audio collections is steadily increasing. Effective interaction with these large audio collections poses significant challenges to traditional user interfaces. Music management software typically allow users to select artist, genres or individual tracks by browsing long sortable lists of text. This mode of interaction, although adequate for small music collections, becomes increasingly problematic as collections become larger especially when screen estate is limited. A variety of alternative ways of browsing music collections have been proposed mostly in academic contexts. They typically rely on a combination of audio signal analysis to automatically extract features followed by visualization techniques to map the feature space to a 2D or 3D representation for browsing and navigation.

Tag clouds provide both an overview of the information space as well as direct search support that is particularly suited for mobile phones with small touch screens. In this paper, we present content-aware self-organizing tag clouds a technique that attempts to support querying, browsing, and summarization using the familiar information model of a tag cloud while taking into account automatic content analysis information as well as location based information.

2. RELATED WORK

Although there is existing work in location-based applications and automatic/semi-automatic playlist generation there seems to be a lack of published material on location-aware playlist generation. With respect to intelligent playlist creation, Flexer et al. have proposed using audio similarity based on Mel Frequency Cepstrum Coefficients (MFCC) and Gaussian models to create a similarity matrix and select songs that blend from and into a user-selected start and end track in a playlist [1]. Pampalk et al. have proposed using user behaviour based on track skipping to determine what artists, genres, rhythms, etc., the user prefers to pass-over [5]. With respect to location-aware playlist creation most existing work simply associates particular pieces of music with specific locations [7].

The current generation of mobile phones feature decent sized displays that also include touch functionality. Interfaces for managing large audio collections based on long lists of scrollable text are not particularly convenient in such displays. An alternative that has mostly been explored in research literature is the use of content-based visualizations of music collections [4].

Tagging systems allow users to add keywords, or tags, to resources without relying on a controlled vocabulary and have become ubiquitous in web-based systems. Tags are aggregated from many users forming "folksonomies" which, although not as accurate as well-designed ontologies, have the advantage of reflecting how users perceive the data and how their vocabulary and perception evolve over time. Tagging is simple and does not require a lot of thinking. Tags form an essential part of personalized internet radio and music community websites such as Last.fm [1]. Tag clouds are the most common way of visualizing tags. They are two-dimensional stylized visual representations of a list of words where the more prominent words are typically assigned a larger font. They are useful for quickly giving users the gist of a set of words. Tag clouds are in common usage on a number of different social networks such as Flickr [2] but trace their origins back at least 90 years to Soviet Constructivist art [16].

There has been considerable research in recent years into the design, use and effectiveness of tag clouds. A historical look at tag clouds is presented in Viegas and Wattenburg [16], which looks at the development of tag clouds since their inception a decade ago, and speculates about their development in the future. In the paper "Seeing things in clouds" [2], an extensive evaluation of different types of visual features in tag clouds, including font size, font weight, intensity, number of characters and area were investigated. Tag navigation in general has been examined in detail with particular focus on "Last.FM", an online social community for music [10]. A context aware browser for mobile devices that uses tag clouds is presented in Mizzaro et al. [11].

Islands of Music [12] is a a content-based visualization of music collections that uses Self-Organizing Maps (SOM) to generate a two-dimensional representation of a collection of music. MusiCream [8] is an interface that allows users to interact with a music collection using a dynamic visualization interface. MusicRainbow [13] is a similar system that uses web-based labelling and audio similarity to visualize music collections. Examples of visualizations for music discovery in commercial and research systems can be found in the Visualizing Music blog [3].

3. SYSTEM DESCRIPTION

Our proposed system takes as input the user's location, the current playing and associated metadata as well as content-based similarity information between all tracks in a user collection. This information is stored in a database for organization and retrieval. The system processes these inputs to generate location-based information such as common paths and make automatic recommendations based on them. Semantic information related to the generated playlists such as track names, artists, genres, tags, playlists are rendered based on self-organizing tag clouds that are computed based on automatically extracted audio features.

3.1 Location and Path Logging

We introduce the following terms to describe location information: **Paths** consist of a start and end location and a collection of **Path Segments** which consist of a start, end, bearing and segment speed. The **Path Segments** are determined by a list of **Location Points** which are instantaneous snapshots of what song is playing and where. This includes a track's metadata (artist, album, title, etc.), current coordinate and time, and whether a song started or skipped.

As a user's location or music changes and location points are generated, the system interpolates the user's current line-of-travel in real-time and generates a path segment consisting of a line between start and end coordinates. These path segments are then associated to a path from the start location of the first path segment to the end location of the last path segment. These paths can then be profiled by counting the songs that are played or skipped, the most listened to genres or tempos, etc.; therefore, as the user builds up a path history, it can be used to generate a more accurate representation of the user's listening tastes.

One of the challenges of determining path segments is that location estimates vary in accuracy and are sampled irregularly. In addition a user following the same path in different days (for example taking the bus to school) will not have exactly the same set of location points. Therefore we have developed an algorithm for determining determining path segments from a running list of location points. The basic idea is to first determine the bearing between the first two location points in a path segment. Subsequently

[1] http://www.last.fm
[2] http://www.flickr.com
[3] http://visualizingmusic.com/

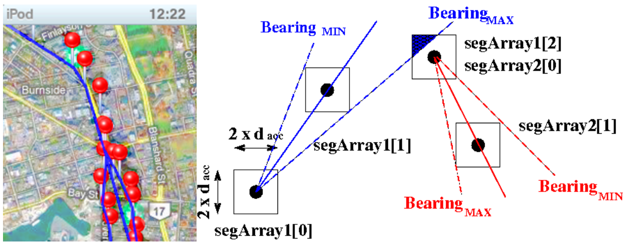

Figure 1. Visualization of paths and location points on a map and schematic of path finding algorithm

the bearing between the start point of the segment and subsequent points is determined. If the new point has the same bearing as the original pair, the new point becomes the end to the segment. This continues until a coordinate yields a bearing of the current segment's path. This basic algorithm works when travelling in very straight lines, and with very accurate positioning hardware, but in real world usage will generate segments between almost every pair of points, as any deviation in bearing will result in a new segment being generated.

In order to account for the accuracy of the positioning system, an algorithm was devised to allow for variation in the absolute location based on the intrinsic accuracy of the mobile device. Each absolute position is reported as a box bounded by the accuracy of the device. Consequently, any points in the bounding box are considered the same absolute coordinate. The same bounding box is used in calculating the bearing for path segments.

These located segments are combined from a start location to an end location in order to generate a path. Figure 1 shows a schematic diagram of the algorithm and a map with paths and location points overlayed. Currently, a path is started when the first change in a user's location is sensed. A path is ended when a user stays at a location for more than 15 minutes. Basic equations for finding distances based on decimal degree coordinates for latitudes and longitudes, and for finding the bearing between two coordinates are based on the WGS84 world representation (currently used by GPS systems).

3.2 Audio Feature Extraction and Recommendations

The goal of audio feature extraction is to represent each track as a vector of features that characterize musical content. First low-level features such as the Spectral Centroid, Rolloff, Flux and the Mel-Frequency Cepstral Coefficients (MFCC) are computed approximately every 20 milliseconds. To capture the feature dynamics we compute a running mean and standard deviation over the past M frames (the so-called "texture window" typically around 1 second). The result is a feature vector of 32 dimensions at the same rate as the original 16D feature vector. The sequence of feature vectors is collapsed into a single feature vector representing the entire audio clip by taking again the mean and standard deviation across the 30 seconds (of the sequence of dynamics features), resulting in the final 64D feature vector per audio clip. A more detailed description of the features and their motivation can be found in Tzanetakis and Cook [15]. For the calculation of the self-organizing map described in the next section all features are normalized so that the minimum of each feature across the music collection is 0 and the maximum value is 1. This feature set has shown state-of-the-art performance in audio retrieval and classification tasks for example in the Music Information Retrieval Evaluation Exchange (MIREX) 2008 and was computed using the free Marsyas audio processing framework [4]. Most audio feature sets proposed exhibit similar performance so we expect that any audio feature front end can be used.

Based on a distance matrix calculated between all pairs of tracks, 3 different recommendation algorithms are implemented. In the naive similarity case, a random seed-song is selected, and playlists of the ten most similar songs (based on pre-calculated Euclidean distances) were created. If the user skipped a song, a new seed is selected and a new playlist is generated along with it. In the similary-with-history case, a profile is constructed based on songs the user listened to at the same time and day of the week to recommend similar songs. A seed song is selected based on tracks that the user enjoyed at similar times (current time $+/-$ an hour) in the past and their three nearest neighbours. If a user skipped a track, a new seed based on their history is selected and a new playlist is generated. Using location information, the system predicted a path that the user is taking and selects a seed from a similar track that was listened to on that path previously. Finally we provide interactive control to the specificity of the generated playlists using the accelerometers included in more mobile devices. Shaking the device at varying levels results in selecting seeds scuh that recommendations are more similar if the shake is light and less similar if it is heavy.

3.3 Self-Organizing Maps

For creating the visualization layout we utilized the self-organizing map (SOM) which is a type of neural network used to map a high dimensional feature space to a lower dimensional representation while preserving the topology of the high dimensional space. This facilitates both similarity quantization and visualization simultaneously. The SOM was first documented in 1982 by T. Kohonen, and since then, it has been applied to a wide variety of diverse clustering tasks [14]. In our system the SOM is used to map the audio features (64-dimensions) corresponding to each track to two discrete coordinates on a grid.

The traditional SOM consists of a 2D grid of neural nodes each containing a n-dimensional vector, $\mathbf{x}(\mathbf{t})$ of data. The goal of learning in the SOM is to cause different neighbouring parts of the network to respond similarly to certain input patterns. The network must be fed a large number of example vectors that represent, as closely as possible, the kinds of vectors expected during mapping. The data associated with each node is initialized to small random values before training. During training, a series of n-dimensional

[4] http://marsyas.info

vectors of sample data are added to the map. The "winning" node of the map known as the *best matching unit* (BMU) is found by computing the distance between the added training vector and each of the nodes in the SOM. This distance is calculated according to some pre-defined distance metric which in our case is the standard Euclidean distance on the normalized feature vectors.

Once the winning node has been defined, it and its surrounding nodes reorganize their vector data to more closely resemble the added training sample. The training utilizes competitive learning. The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU. The time-varying learning rate and neighborhood function allow the SOM to gradually converge and form clusters.

3.4 Self-Organizing Tag Clouds

The technique of self-organizing tag clouds can be viewed as a fusion of concepts from text-based visualization interfaces and more abstract content-aware visualization interfaces. We use the term tag loosely to denote any metadata associated with a track such as genre, artist or year of release. Traditional systems based on long lists of sortable text such as iTunes provide little support for browsing, discovery and summarization. An alternative is visualization interfaces that are based on automatic analysis of musical content. By mapping the music collection onto a 2D or 3D representation they enable quick browsing and navigation especially in the case of music that is not known to the user or that has not been tagged.

Tag-clouds provide a simple, familiar interface that partly overcomes these limitations. For example they support both direct searching as well as browsing and navigation. However they come with their own problems. In order for a tag to assist search or browsing it is necessary for the user to have some notion of its meaning. For example a specialized term such as indie pop might be completely unfamiliar to a particular listener while at the same time essential to another. This problem becomes even more acute using the more generalized notion of tags that includes information such as artist or album. As one of the goals for an effective interface of music collection browsing is the discovery of new music by artists not known to the listener, this is an important disadvantage. Simple tag clouds do not provide the user with any information about the connections and similarity relations between tags. A final problem with any system based solely on tag information is that there is no way to access music tracks that have not been tagged (the so-called "cold start" problem). By contrast content-based visualizations allow any track to be accessed and do not require familiarity with the music explored.

We describe a new method for organizing music tag clouds that makes a persistent map taking into account the musical similarity between songs. Figure 2 shows an example of a self-organized tag cloud. Each label (artist, genre, tag) is associated with a set of tracks that have been annotated with it. As the tracks have been mapped to fea-

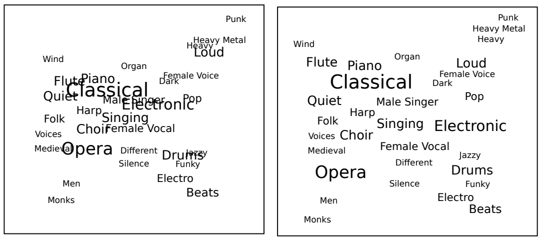

Figure 2. Self-Organizing tag cloud before and after mass-spring layout algorithm

ture vectors and subsequently to 2D grid coordinates by the SOM, each tag is associated with a set of 2D grid coordinates. The SOM process ensures that neighboring points (tracks) will have similar high-dimensional audio features and therefore similar musical content. The tags are placed on the centroids of their corresponding set of 2D grid coordinates. Their placement reflects the underlying musical content but results in visual overlap between them.

This initial layout contains many overlapping words, so the position of each tag is repositioned using a mass, spring and damper force-based algorithm for drawing [6]. In our implementation each tag is anchored to its original position using a spring and an electrostatic-like force is applied between every pair of tags that is proportional to the inverse of their squared distance. Therefore tags that are close and overlapping will be pushed away while still trying to remain close to their original location. An additional wall force term was added to keep all tags within the designated window. The font size for each tag was determined by counting the number of instances of that tag.

There are some interesting characteristics of the resulting visualization that we would like to highlight. The first is that tags that are not correlated with the acoustical content will correspond to tracks spread across the underlying self-organizing map and therefore their placement will be in the center. For example in Figure 2 the tags Male Singer, Singing and Female Vocal are near the center as they have a large variety of tracks that have been annotated with them. In contrast more specialized tags such as Heavy Metal or Monks are more localized. The second important characteristic is that faceted browsing is naturally supported. For example an artist name, that the user might not be familiar with, located near the left corner will correspond to the tag Monks. Finally a track for which there are no tag annotations will still be placed on the underlying self-organizing map and that way receive an implicit visual automatic tag annotation addressing to some extent the cold-start problem.

3.5 Implementation

The feature extraction, music similarity calculation and self-organizing map training are performed using the Marsyas audio processing framework. Our current prototype application GeoShuffle has been implemented for Apple Inc.'s iPhone or iPod Touch devices. The application dynamically generates music playlists that can be played in the default iPhone/iPod Touch music player based on location, path of travel, historical information and content similarity. To provide feedback to the user on their preferences by path, as well as to test the accuracy of the application, a Google Map generated map has been embedded into the application (see Figure 1). This map supports annotations in the form of paths or absolute location points. The device's positioning system provides real-time updates on the user's absolute position. This allows the user to visually trace their daily commutes and inspect their musical taste over each path.

4. EVALUATION

Evaluating a complex system and user interface such as the one described in this paper is challenging due to its subjective nature. We focus on two aspects of our work: 1) the use of self-organizing tag clouds as a way to explore large music collections that combines text and content information without requiring large displays 2) the use of location information to improve music recommendation.

For evaluation purposes we used a subset of the Magnatagatune dataset consisting of 1141 tracks with each artist represented by at most 3 tracks. This was chosen as a large enough dataset to have considerable variability while at the same time being manageable in the limited storage of the iPod Touch used for development. There are 341 artists represented and also 14 top-level genre labels. In addition to the regular meta-data information such as artist and genre, also includes tags derived from the Tagatune Game with a purpose [9]. The dataset has been made available to the scientific community for use in research.

For evaluating the self-organizing tag clouds, 14 participants were recruited from graduate Computer Science students. Three were female and 11 were male. All subjects had normal or corrected-to-normal vision, enjoyed listening to music and were experienced computer users. None of the participants had previous knowledge of the Magnatune dataset. The user study consisted of a 5-point system usability survey (SUS) [3].

The survey consisted of six questions, each rated on a five point scale, where "1" was labelled "Strongly disagree" and "5" was labelled "Strongly agree". The 6 questions were: 1) I thought the application was easy to use, 2) I needed to learn a lot before I could accomplish tasks with the application, 3) I think people would need technical support to learn how to use the application, 4) I think most people would learn to use the application very quickly, 5) Overall, accomplishing tasks using the self-organizing tag cloud was easy 6) Overall, accomplishing tasks using the self-organizing tag cloud was fun

Results from survey are detailed in Table 1. On average users rated Question 4 highest, which indicated that they thought most other people would be able to learn the application quickly. This question also had the lowest variance. In Table 1 we detail all the responses from the participants. We can see that two participants chose the middle check box, six chose the next one to the right, and six chose the checkbox labelled "Strongly agree".

Table 1. System Usability Survey

Question	1	2	3	4	5	Mean	Std
1	0	1	3	8	2	3.79	0.8
2	5	7	1	1	0	1.86	0.86
3	5	3	3	1	2	2.43	1.45
4	0	0	2	6	6	4.29	0.73
5	0	2	1	4	7	4.14	1.1
6	0	2	0	6	6	4.14	1.03

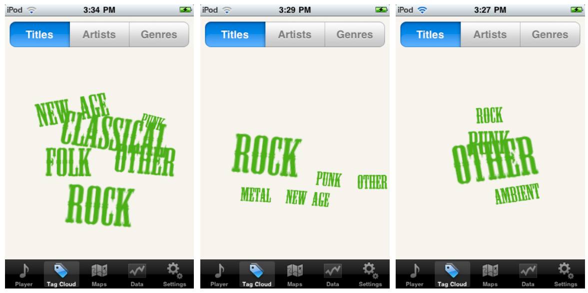

Figure 3. Screen shot of playlist visualization using the Self-Organizing Tag Cloud

In a similar vein, participants also rated questions 5 and 6 highly, although notably, two participants rated this question as one box to the right of "Strongly Disagree". This shows that certain users found our interface easy to use and fit in well with their expectations of an interface to explore music collections, but for other users it did not. For Question 2, the average response was 1.85, which implies that on average, users strongly disagree that they would have to learn a lot before accomplishing tasks with this application. It is important to include negative examples on such a user study to ensure that participants are not just choosing answers to questions randomly; this question performs this control function.

For evaluating the location-aware music recommendation component it was necessary to collect data over an extended period of usage. Usage data was collected from only one subject. The subject used the system over a period of three weeks through their daily routine. GeoShuffle logged their musical preference over the time period and generated sets of user paths (consisting of an origin, destination, and linear path segments). The device switched between four modes of recommendation without the user's knowledge (random, similarity, similarity with history, similarity with location-awareness) and logged which tracks were skipped throughout operation. These results were then used to determine the amount of user skips in each mode of recommendation without biasing the data.

Self-organizing tag clouds can also be used to visualize text information associated with a playlist. Figure 3 shows the self-organizing tag cloud text associated with three playlists (from left to right: random, similarity and path). The figure clearly shows the increase in specificity and the content distribution of the recommended playlists.

Table 2. Number of skips and genres present in playlists created with different generators

	Skips / Track Played	Genres in Playlist
Random	4.3	12
Similarity	1.7	7
+ History	1.2	3
+ Path	0.3	10

Table 2 shows the analysis of skipping behavior between different configurations of the system. We assume that playlists that result in less skipping are better and show the results as average number of skips per track played. The baseline of 4.3 corresponds to randomly selecting songs from the collection in similar fashion to the iPod shuffle. The similarity configuration returns tracks that are similar to all the tracks played in the logging period. The history configuration in addition to similarity takes into account the time of the day. The last configuration also takes into account information about paths taken during the day and is the only one that requires portable devices with location information. As can be seen there is a significant reduction in the number of skips when taking into account location information.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we describe our investigations in designing an interface for content-aware music browsing, discovery and recommendation that is designed based on the unique characteristics of modern smartphones. We propose using location information to improve the quality of music recommendations and introduce self-organizing tag clouds: a visualization of metadata information such as genres, artists, tags and playlists that takes into account automatically extracted musical content information. The specificity of the music recommendation algorithm can be interactively controlled using the accelerometers. The resulting interface is particularly suited for small screen real-estate and touch-screens. Our evaluation indicates that self-organizing tag clouds are an effective and fun way of exploring music collections and that location information can improve the quality of music recommendations.

There are many directions for future work. We plan to explore visualizing tag-based similarities as edges between tags with proportional thickness. Another interesting direction is the addition of social networking and collaboration features such as sharing playlists for particular paths or comparison of collections between different users. Several of the user study participants suggested using the same interface for personalized tag annotation. Finally we plan to conduct a wider ethnographic study where self-organizing tag clouds and location-based recommendation are used in personal music collections.

6. REFERENCES

[1] G. Widmer A. Flexer D. Schnitzer, M. Gasser. Playlist generation using start and end songs. In *Int. Conf. on Music Information Retrieval (ISMIR)*, Philadelphia, USA, 2008.

[2] S. Bateman, C. Gutwin, and M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proc. ACM conf. on Hypertext and Hypermedia*, pages 193–202, 2008.

[3] J. Brooke. Sus: a "quick and dirty" usability scale. In *Usability Evaluation in Industry*. 1996.

[4] M. Cooper, J. Foote, E. Pampalk, and G Tzanetakis. Visualization in audio-based music information retrieval. *Computer Music Journal*, 30(2):42–62, 2006.

[5] G. Widmer E. Pampalk, T. Pohle. Dynamic playlist generation based on skipping behavior. In *The International Society for Music Information Retrieval (ISMIR 2005)*, London, UK, 2005.

[6] J. Ellson, E.R. Gansner, E. Koutsofios, S.C. North, and G. Woodhull. Graphviz - open source graph drawing tools. *Graph Drawing*, pages 483–484, 2001.

[7] K. Eustice and et al. The smart party: A personalized location-aware multimedia experience. In *Consumer Communications and Networking Conference (CCNC 2008)*, Las Vegas, USA, 2008.

[8] M. Goto and T. Goto. Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[9] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proc. CHI 2009*, pages 1197–1206, 2009.

[10] C.S. Mesnage and M.J. Carman. Tag navigation. In *Proc. Int. Workshop on Social software engineering and applications*, pages 29–32, 2009.

[11] S. Mizzaro, E. Nazzi, and L. Vassena. Collaborative annotation for context-aware retrieval. In *Proc. of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 42–45, 2009.

[12] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[13] E. Pampalk and M. Goto. Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2006.

[14] Kohonen. T. *Self-Organizing Maps*. 1995.

[15] G. Tzanetakis and P. Cook. Musical Genre Classification of Audio Signals. *IEEE Trans. on Speech and Audio Processing*, 10(5), July 2002.

[16] F.B. Viégas and M. Wattenberg. Timelines : Tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008.