

# The Streams of Our Lives: Visualizing Listening Histories in Context

Dominikus Baur, *Student Member, IEEE*, Frederik Seiffert, Michael Sedlmair, and Sebastian Boring

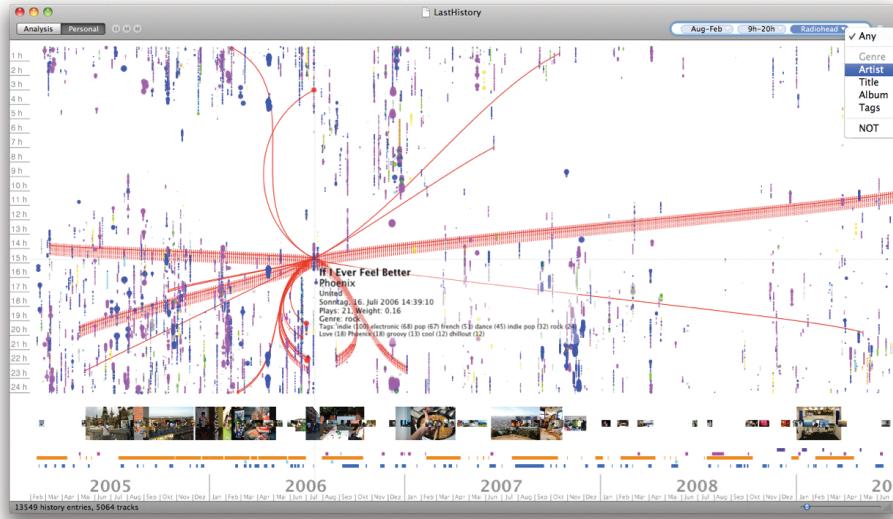


Fig. 1. LastHistory: Visualizing personal music listening histories, photo and calendar streams for analysis and reminiscing.

**Abstract**—The choices we take when listening to music are expressions of our personal taste and character. Storing and accessing our listening histories is trivial due to services like Last.fm, but learning from them and understanding them is not. Existing solutions operate at a very abstract level and only produce statistics. By applying techniques from information visualization to this problem, we were able to provide average people with a detailed and powerful tool for accessing their own musical past. LastHistory is an interactive visualization for displaying music listening histories, along with contextual information from personal photos and calendar entries. Its two main user tasks are (1) *analysis*, with an emphasis on temporal patterns and hypotheses related to musical genre and sequences, and (2) *reminiscing*, where listening histories and context represent part of one's past. In this design study paper we give an overview of the field of music listening histories and explain their unique characteristics as a type of personal data. We then describe the design rationale, data and view transformations of LastHistory and present the results from both a lab- and a large-scale online study. We also put listening histories in contrast to other lifelogging data. The resonant and enthusiastic feedback that we received from average users shows a need for making their personal data accessible. We hope to stimulate such developments through this research.

**Index Terms**—Information visualization, lifelogging, design study, music, listening history, timelines, photos, calendars.

## 1 INTRODUCTION

More and more personal information and artifacts of our individual pasts are available in digital form. A multitude of online services allow collecting and accessing information about one's life, replacing the formerly analog forms of such lifelogging. Despite the often-discussed privacy issues, millions of people spend their time recording their exercise and running habits<sup>1</sup>, their moods<sup>2</sup> and even their nocturnal activities<sup>3</sup>. These services often enrich the available data with statistics and small graphs but their main use is to record and allow direct access.

Still, such lifelogging data can quickly become quite complex,

especially when the information is no longer tracked manually by the user but automatically in the background. One such example is Last.fm<sup>4</sup>, a popular webradio that promises to deliver personalized music by analyzing one's listening behavior. The most common way to do this is installing the Audioscrobbler, a demon process that follows the currently played music in media player software and sends this data to the Last.fm servers (scrobbles it). The by-products of this process, the recorded listening histories, are meticulous representations of one's music consumption and already have become the actual reason for many people to use Last.fm. They can, however, quickly span tens of thousands of songs and become too complex to be understood from the chronological lists that Last.fm provides.

A large number of fan-created static visualizations and analytic tools are therefore available that range from timelines displaying the number of logged songs (Scrobbing Timeline<sup>5</sup>), via high-level com-

- D. Baur, M. Sedlmair and S. Boring are PhD students with the Media Informatics Group at the University of Munich (LMU), E-mail: {dominikus.baur, michael.sedlmair, sebastian.boring}@ifi.lmu.de
- F. Seiffert is a MSc student with the Computer Science department at the University of Munich (LMU), E-mail: ego@frederikseiffert.de

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

<sup>1</sup><http://www.nikeplus.com>

<sup>2</sup><http://www.moodstats.com>

<sup>3</sup><http://www.bedposted.com>

<sup>4</sup><http://www.last.fm>

<sup>5</sup><http://playground.last.fm/demo/timeline>

parisons of multiple users (Last.fm Explorer<sup>6</sup>) to visually appealing StreamGraphs of the artists from a listening history (LastGraph<sup>7</sup>). The main problem with these tools is that they might give an interesting and entertaining overview of the history but fall short in representing detailed information such as individual songs.

We believe that a ‘casual information visualization’ approach can prove valuable for making this personal information available to their creators [25]. In this paper we analyze the data domain of listening histories and present our findings on their structure and what possible user tasks and available patterns it might contain. Additional contextual information can trigger the memory of the user to reveal the reasons for listening decisions. As a second contribution, we give an overview of LastHistory (see figure 1), a visualization of personal listening histories from Last.fm that not only allows sophisticated analysis of the underlying data in a non-threatening way, but is also able to show contextual information in the form of photos and calendar entries to help the user remember this time of his or her life. For this design study, we discuss the applied transformations for interface and views, user interactions and – with more detail – the differences between the two usage modes (analysis, reminiscing). We also evaluated LastHistory with four casual users and present anecdotal evidence for its value. A subsequent large-scale online evaluation with a corresponding questionnaire shone more light onto usability and feature issues. After making LastHistory available, several thousand people downloaded it and left generally positive feedback. We conclude with an explanation how the techniques used in LastHistory can be applied to other forms of lifelogging data and what other approaches might be fruitful.

## 2 LISTENING HISTORIES

Listening histories of (popular) music have certain unique characteristics that have to be taken into account when visualizing them. Our first contribution is an encompassing description of this type of data.

The term *listening history* describes an account of consumed music by a single person and has become relevant recently, as the digitization of music and the proliferation of personal portable music devices made acquiring this information possible. We understand a perfect listening history as *a complete chronological collection of musical items where each one (1) is a pre-existing piece of music that can be identified based on some of its attributes (e.g., artist, title) and (2) has been heard by the user at least in parts during the recorded time interval*.

This definition means that meaningless jamming on the guitar should not be contained in a listening history, but hearing the live version of a song should. Also, music does not have to be actively chosen by the user: Listening to a song in a club or at a friend’s place also contributes to the listening history.

### 2.1 General aspects

From the viewpoint of information visualization, listening histories can be interpreted as multivariate time series data. For a univariate perspective, each point in time either contains music or it does not, as people tend to avoid having multiple music sources running in parallel. Based on the above definition of listening histories consisting of identifiable pieces of music, these songs represent a suitable basic unit of data. Each song has a certain start and end point and can appear more than once. A useful abstraction for the temporal aspect of listening histories is the identification of listening sessions or events. Sessions are chains of songs that are consumed at a stretch and can be found by analyzing the pauses between songs. Another important temporal aspect are repeating sequences of songs such as albums, user-generated playlists and mixtapes.

Going beyond the univariate characteristics of songs in a listening history, there is an abundance of additional metadata for the field of music. A common (but vague) abstraction that combines content- and contextual information is the *musical genre*. Genres such as “Rock”, “Pop” and “R’n’B” describe not only a defining sound and style but

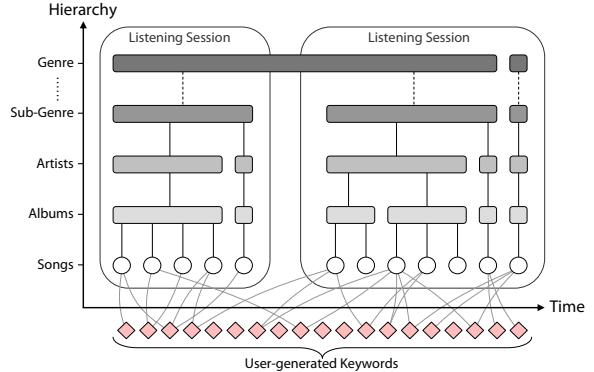


Fig. 2. Listening histories can be seen from a temporal or a hierarchical perspective. User-generated tags add a non-hierarchical layer to them.

also a certain context of the music (e.g., “Brit-Pop”). Genres are hierarchical in nature (“Alternative Rock” is a sub-genre of the more general “Rock”) and this hierarchy is commonly extended to artists and their albums [8] (see figure 2). Despite their downsides regarding the unambiguous classification of songs [2], genres are commonly used and easily recognizable even by non-musicians.

An overlapping way to look at songs are user-generated keywords or *tags*. Folksonomies can produce reliable metadata descriptions of items through the combined efforts of many people [20]. In the case of music, tags might contain their genre, but also attributes such as mood (“peaceful”), tempo (“slow”) or context (“seen live”). An automatic extraction of genre or mood is at the moment still hard to do reliably [2], so user-generated tags provide a directly available alternative to that. Also, non-hierarchical tags solve the problem of items belonging to two distinct categories.

In summary, listening histories can be classified according to Aigner et al.’s taxonomy [1] as consisting of *time intervals in linear time with abstract, multivariate data*. Hierarchical (i.e., artist, album, genre) or non-hierarchical (i.e., tags) metadata and clustering into listening sessions can work as *data abstractions*.

**Constraints of real-world data:** Data about real world listening histories is often incomplete and noisy. Looking at our data from Last.fm reveals two basic limitations to perfect listing histories as we defined it above: noise and data gaps. Noise in a listening history describes all songs that were tracked by the Audioscrobbler demon but were not heard by the user: another person is using the same computer for listening to music and forgets to stop the scrobbing process or the user leaves the computer while the music continues playing. The second restriction of real world listening histories are data gaps that can appear for many reasons: Users may have a non-supported device for listening to music while on the go, they possibly use legacy media such as records or CDs, or visit a concert, a party or a club where music is played. Another source of data gaps might stem from Audioscrobbler’s intention to detect only songs that are liked by the user: Scrobbing only happens once a pre-defined time interval has passed, so that quick skipping through multiple songs is ignored. Finally, data gaps can also be created intentionally: Last.fm allows users to delete tracks from their listening histories to remove noise or embarrassing songs. Adding past tracks and thus filling data gaps is not possible. Last.fm has no problem with (low) noise and data gaps, because its main incentive, creating recommendations, is sufficiently fuzzy to work with incomplete data. When users want to access their own listening histories it can however be frustrating to discover unknown music or find a personal favorite missing.

**Contextual information:** Music is often enjoyed within a certain context: Listeners follow strategies when choosing music and tend to choose fast, energetic music for powering their exercising or use calmer, more mellow one for regulating their moods [8]. Unfortunately, pure listening histories are void of this information as they

<sup>6</sup><http://alex.turnlav.net/last.fm.explorer>

<sup>7</sup><http://lastgraph.aeracode.org>

Table 1. Patterns in listening histories categorized according to duration, frequency and periodicity

	Music present / absent	Single songs	Song sequences (albums, playlists)	Genres and artists
Duration	Length of listening sessions, noise/data gaps	Long/short songs	Long/short sequences	Tendency to listen to the same genre/artist in a row
Frequency	Intensity of music consumption, special events (parties, holidays), loss/gain of interest in music	Intensity, most/least favorite songs	Intensity, most/least favorite sequences	Intensity, most/least favorite genres/artists
Periodicity	Daily/weekly/monthly routine (sleep-wake rhythm), changes in timezone	Turnover rate for songs	Turnover rate for sequences	Flexibility/changes in taste

contain only songs and timestamps. To make sense of them, context is helpful.

In this regard, episodic memory, the mental storage of autobiographical events, is relevant. Psychological research on it [31] acknowledges the importance and variety of pieces of data that can trigger a person's memory. The pure timestamp might not suffice, as people in general have problems accessing specific information based on temporal cues alone (the question "What did you have for dinner last Tuesday" is an example for that). A second psychological effect that is important here is the "generation effect" [17] that states that items (mostly words, but also pictures) which are generated by a person are easier remembered than ones that were just read. Blog posts, status messages, photos or calendar entries should therefore work better than items from automated sources. A study performed by Ringel et al. in 2003 on contextual landmarks such as photos, calendar appointments, news headlines and holidays for searching a personal information store showed that personal contextual information (photos, calendar entries) were much more relevant to participants than the impersonal news headlines [26]. Also, images are more powerful memory cues than non-visual ones [28].

All in all, information about the listening context is important for bare histories and can be triggered from a listener's memory by using self-generated pieces of information.

## 2.2 User Groups

Before describing user tasks it is necessary to define these users. First of all there are the producers of listening histories, users of Last.fm or similar services. A second user group constitutes researchers from psychology or sociology: Music and cognitive psychologists are eager to learn the reasons why people listen to music. Listening histories are therefore a valuable source of data for their research, although their interest in context often requires them to make use of more cumbersome manual data collection (e.g., [22]). A third user group are analysts and marketers. While they also try to gauge what makes good music (to produce more of it), their commercial interest also includes target demographics, the spread of songs through social networks and intellectual property. In our work, we focus on the first group of interested average users in a casual infovis approach.

## 2.3 Tasks and Patterns

Sellen and Whittaker recently presented an overview of user tasks in lifelogging applications and established a set of five typical user tasks: *Recollecting, reminiscing, retrieving, reflecting and remembering intentions* [28]. Given the background of listening histories we decided to focus on three of these: *Analysis* activities (comparable to *reflecting* in Sellen and Whittaker's taxonomy) describe the typical tasks of finding patterns and testing hypotheses based on a set of data (in this case: a listening history). The data set can come from an arbitrary user which makes this use case also suitable for music psychologists. With *Personal mode* we mean the more intricate and loosely defined activities of *recollecting* and *reminiscing* (we decided to combine these closely related tasks). Here, listening histories are used partially as a tool for learning about oneself and re-living prior sections of one's life,

but also as a vehicle of self-presentation and telling stories of one's life (see [5]). Therefore, the listening history and contextual information has to come from the user directly to make sense to him or her. Also, note that there is a certain overlap between the two tasks.

**Analysis** In the following we describe the patterns that are available in listening histories. We classify them according to level of abstraction and temporal aspects (see table 1). The level of abstraction starts with the binary presence or absence of any music and goes via single unique songs, repeating sequences of songs such as albums or playlists to the hierarchy of genres and artists. For the temporal aspects we use three common attributes: *Duration*, *frequency* and *periodicity* [19]. Interesting patterns always correspond with either uncommonly high or low values for these categories.

The simple presence or absence of any music contains information about the general intensity of music consumption, shows special events with uncommonly high or low values or a gradual loss/gain of general interest in music. Also, daily routines such as the time the user goes to bed or gets up and possibly changes in location and time zone can be seen.

Single songs and sequences of them act similarly. They can either be uncommonly long or short (compared to other songs and sequences) and their frequency shows the intensity of consumption (e.g., a higher playcount right after the release of an album) as well as the most/least favorite items. Finally, the turnover rate represents a user's tendency to listen to songs/albums only once or repeatedly.

Patterns for genres/artists are similar to songs and albums, but have far less values. Therefore, their duration shows how long one genre/artist is listened to in a row. The frequency gives the intensity of listening at a certain time and the most/least favorite genres and artists and periodicity shows the flexibility of the user and gradual changes in taste.

**Personal mode** In the previous description of possible insights that can be gained from looking at a listening history the notion of *reasons* often appeared. Certain patterns in a history might be clearly visible, but as discussed before interpreting such patterns can be difficult without certain background information. Therefore, the second use case enhances listening histories with additional information for understanding, reminiscing and storytelling. Providing contextual information can help the creator of a listening history with including it into the personal narrative of his or her life. To help the user gain access to the background information in his or her own head, a small trigger such as a photo or a description of a memorable event at the time can help [26].

When the background is available, patterns in a list of impersonal songs become personal milestones of one's life. Extreme changes in musical behavior may stem from trying to cope with a breakup or the loss of a dear person, repetitive use of mellow music stands for the lazy evenings at the beach and a major upswing in the popularity of a certain artist can reflect the memorable concert that s/he gave. But enhanced listening histories can also be read the other way around: Users might want to know what music they listened to during a memorable experience of their life and would like to be able to find that.

### 3 RELATED WORK

In LastHistory, we build on work from information visualization as well as human-computer interaction.

Freeman et al. presented the lifestream metaphor [11] for managing one's digital data, but without scalable methods of access beyond search and with no visualization. Temporal and timeline data is one of the most common data types in visualization research and, in consequence, various approaches to visualize it exist (see [1] and [21] for an overview). While the application scenarios often involve spatial and scientific information, there are also visualizations for non-spatial and biographical data: LifeLines [24] is a prominent example for general-purpose creation of personal timelines in medical or legal domains which is extended by PatternFinder [10] that focuses on event-based patterns. LastHistory similarly contains multiple coordinated timelines for listening histories. We further integrated the concepts of listening sessions and photo as well as calendar events. Periodic data and the daily routine is also important for music listening. However, for visualizing them we used a two-dimensional timeline with hours and days instead of spirals [7, 38] or calendar and cluster-based approaches [32]. The concept of Timeboxes [14] allows interactively creating a query for a period of time and a value range which we used in LastHistory for selecting songs that appear within longer time periods in certain hours.

Related concepts for browsing and searching in media collections exist. Photo browsers often also contain zoomable timelines (TimeQuilt [15]) or focus on displaying representative photos chronologically (Calendar Brower [13]). MusicLand [18] supports exploratory search within music collections. The Disc visualization [30] provides overview and access to large collections of that kind. Still, personalization is not in the focus of these approaches and neither usage nor browsing histories are incorporated.

Another chronological type of personal data are e-mail conversations. The work by Viégas et al. partially inspired LastHistory: PostHistory [33] and Themail [34] provide chronological insight into communication patterns. Reminiscing and making sense of the past are central activities in both approaches. For visualizing listening histories, a multitude of fan-created high-level visualizations for Last.fm are available (for an overview see <sup>8</sup>), but this type of data has only scarcely been touched in the information visualization field: Byron and Wattenberg used stacked graphs for abstract visualizations of histories [6]. In our own former work [4] we presented two playful visualizations for listening histories that work, however, only with up to a thousand songs and, most importantly, lack sophisticated analysis capabilities.

Listening behavior itself is used for automatically creating playlists [23] or for recommending music [29], but not for visualization. Some of the online lifelogging services mentioned above provide small automatically generated graphs (most prominently dayum<sup>9</sup>), but work for a different type of data and only encode small amounts of information. Finally, music psychologists want to find out what uses people have for music in their everyday lives. Data acquisition happens through questionnaires and observation [9] or by calling participants on their mobile phones at certain times of the day [22]. Analysis is based on statistics and gives a cumulative overview of people's tendencies and strategies when listening to music. Individual behavior as well as online data sources are not used.

### 4 LASTHISTORY

LastHistory (see figure 1) is a visualization tool for listening histories and personal context and should serve as a design study in this area. It aims at non-expert users and allows them to analyze, encourages reminiscing and supports storytelling. In the following section and as a second contribution of this paper, we describe the design rationale, the applied transformations to the data, the visualization itself, possible user interactions and the integration of contextual information from photos and calendar entries.

<sup>8</sup><http://build.last.fm>

<sup>9</sup><http://www.dayum.com>

### 4.1 Design Requirements

When designing LastHistory we were aware that our proposed user group and target domain had several unique properties:

*Non-expert users:* We wanted to make sure that non-infobis-experts were able to use LastHistory. Therefore, we first had to make sure that they were not immediately put off by a complex interface and make it appear "non-threatening". Also, while more sophisticated mechanisms from information visualization should be available, they should not be compulsive. Thus, users could simply launch the application, explore the basic features on their own and once they were sufficiently profound in them discover and learn more complex ones. Last but not least, we had to assume that there would be no special lifelogging data available.

*Non-vital tasks:* A second issue is the non-vitality of the proposed tasks of analysis and personal mode: While they provide interesting information which is nice to have, users would probably survive without access to it. Also, using LastHistory would for most people be a one-shot experience or at least one with a very low frequency. So it was important to avoid any annoyance that could gall the experience: Users had to be aware what value the tool had for them and be able to immediately draw a benefit from using it: Even before interacting with it, the static visualization had to be understandable and deliver insight. Also, it had to work and acquire necessary data out-of-the-box without a long and complicated installation. No training was preferable for immediate access. Last but not least, aesthetics were a central point for making the interface as appealing as possible.

*Missing and imperfect data:* Due to the nature of how listening histories are captured noise and data gaps are unavoidable. We had to make sure that users were aware of this incompleteness and did not blame it on the visualization. Also, the more contextual information one wants to integrate the harder it is to find people who actually have it available on their machines. And even if these contextual items are available, they might not be available for the time when the listening history was recorded. Therefore, we wanted to make sure that the application also works and is still useful with incomplete data either in the listening history or the contextual data.

### 4.2 Data and Interface

In the following, we describe the general interface of LastHistory and what data is available to the user. Then, we give the ideas behind the chosen transformations for data and visualization and their shapes within the application.

#### 4.2.1 Analysis and Personal modes

In order not to overwhelm users with a multitude of settings available, we decided to group them into reasonable defaults for our two use cases Analysis and Personal modes. Users are able to switch from one to the other using two buttons in the upper left corner of the interface. All settings can be changed in detail from the menu. *Analysis* (the default mode) serves as an entry point to LastHistory with the unweighted listening histories only, while *Personal* adds contextual information and song weighting (see below). Interaction techniques, filtering capabilities, etc. are identical which leads to a certain overlap, but also to consistency.

#### 4.2.2 Data used

Our main data source is Last.fm which provides listening histories and additional music metadata. Listening histories are sequences of time stamps, artist names and song titles. Based on them we acquire tags to extract the genre of a song.

For contextual information we used for one digital photos. Their ability to capture a moment and a user's tendency to preferably take pictures of (personally) meaningful events makes them great memory triggers. Also, they are mostly organized and stored digitally. Additional context information were calendars. We assumed that many people would manage their appointments both personal and professional on their computers and that this calendar data would be available. The repetitive minutiae of daily life would not help in triggering memories which is why we decided to only use calendar entries with

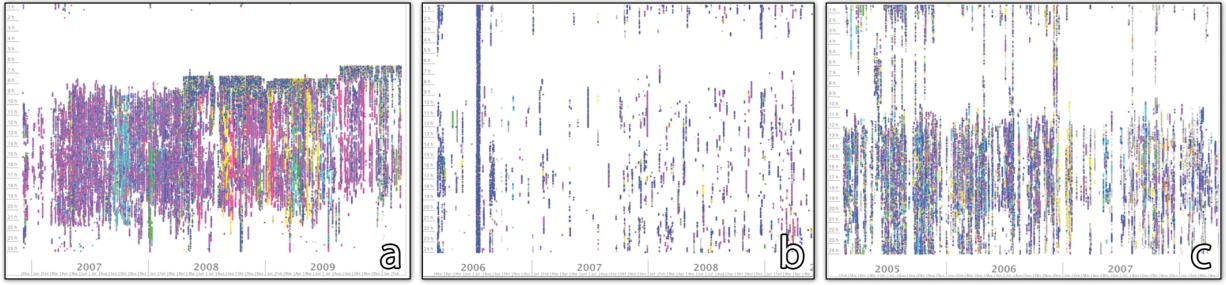


Fig. 3. Three exemplary listening histories in LastHistory. Several patterns are clearly visible: a) listening to music first thing in the morning b) several days of permanent music consumption (probably noise) c) regular listening history with sleeping intervals and some long nights

a length of at least one day: These events comprise vacation trips, holidays, conferences etc.

Once LastHistory launches, a login screen is displayed where the user has to enter his or her Last.fm username (a password is not required as all data is freely available online). From then on, all data is acquired automatically: The system downloads the user's history and corresponding meta-information (albums, community generated keywords). Also, photos and calendar entries are loaded automatically and included into the visualization without the user's involvement which helps in keeping the entry barriers as low as possible. For a discussion of additional data that could be included see below (section 7).

#### 4.2.3 Focus on time

As discussed above, in this data domain time is important enough to take up the largest space in our visualization. While time is actually one-dimensional, we wanted to take repetitive cycles into account and split it into two visual dimensions. When deciding what length of a cycle to choose, we took days since they can hold a manageable number of songs (ca. 400 for an average length of 3.5 minutes) and are a common unit in our lives. Therefore, days are lined up horizontally and hours and minutes of the day vertically in a 2D-grid (as  $f(day, hour)$  [32]). With this visualization, comparisons can directly be drawn between days and the course of listening can be followed through the hours. To provide the user with clear scales, months and years are displayed at the bottom and hours of the day at the left border. The actual date and time for a song can be seen in a tool-tip by hovering above it. The time of day itself is given by Last.fm as UTC and also displayed this way in LastHistory. It could be converted by having access to a user's location at the time, but profiles only supply the latest location if any.

All song instances are displayed as small circles. We described the various patterns that are available on a temporal level in section 2.3. Our visualization makes them available directly: the general amount of music listening (i.e., more circles than white background), daily habits such as the predominant time for music listening or the sleep-wake rhythm, the regularity and changes in this pattern in the form of multi-day events or holes in the listening history and their length all are available at first sight (see figure 3 for examples). Also, with this information being available from the static display, our required "immediate benefit" for users is given: Right after launching the application and even before touching the mouse, they can already learn something about their data. Pre-determining the axes lets them discover interesting patterns without having to give the settings much thought. Mapping other values to the y-axis (such as other frequencies, playcounts, genres, tags, etc.) could be interesting for more sophisticated analysis tasks, but would require more understanding of the subject matter on the user's side. Additionally, it might not always be suitable especially for nominal values.

#### 4.2.4 Display of songs

Songs are the basic units of listening histories and are represented in LastHistory as circles instead of, for example, bars symbolizing their length. Our reasons for this were several: first, a uniform display of

identical circles is less restless and distracting than thousands of lines of varying sizes. Second, acquiring the song length from an online data source would not have reliably reflected the actual listening event as scrobbling already happens after half a song. Third, for a complete listening history with tens of thousands of entries the additional information of song lengths only marginally increases the user's insight. And fourth, even with songs being uniform circles there is already overlap between neighboring songs as the screen resolution is not high enough to display them larger than one pixel and still showing several years of history.

To integrate the underlying musical taxonomy of genres we decided to use color-coding, since glyphs would have suffered from some of the same problems discussed above in the context of visually encoding the length of songs. The human visual system is very restricted in distinguishing colors [35] which is why we tried to keep the number of concurrent colors as low as possible. We ended up with eight different colors and genres: orange (classical), yellow (jazz), green (funk), turquoise (hip-hop), blue (electronic), purple (rock), pink (metal), grey (unknown/other). The colors were chosen to have the maximum distance on the HSV color scale and the genres stem from a literature and web survey. Songs are sorted into genres based on their user-generated keywords, where the most popular tag containing a genre is taken to be the actual genre. If no defined genre descriptor is found in the list of keywords, the 'other' genre is used.

Finding a non-ambiguous genre description for a song is often not possible because of complex genre hierarchies and musical hybrids. Having a low number of genres thus helps in reducing wrong and ambiguous classifications. Also, we decided to use a fixed set of genres instead of, for example, adapting the available genres to the specific history for two reasons: first, to keep the visual representations of different listening histories comparable (i.e., a purple circle in two listening histories always means rock). Second, once the user has learned the mapping between colors and genres s/he can immediately see what genres are dominant without having to re-learn the specific mapping for a certain history. Still, this restriction to seven genres means that uniform histories containing only very specific sub-genres show only one color.

To allow estimating the personal relevance of a track at a glance, we also added an optional representation for this song weight through a circle's size. Factors that should influence the size of a circle were the song's relative importance at the time, but also the overall importance (in a way that songs appearing often throughout the history but only once each month are still reasonably large). The weight  $w$  of node  $h$  is calculated with this formula:

$$w_h = \frac{|t(h)^P|}{|t(h)|} \cdot \frac{|t(h)|}{|t|_{\max}} \cdot m \quad \text{with } m = (1 - \frac{M}{2} + \frac{d(h)}{d} \cdot M)$$

where  $|t(h)|$  are the total history entries for track  $t$ ,  $|t(h)^P|$  the number of history entries in the time period  $P$  surrounding  $h$ ,  $|t|_{\max}$  the maximum number of history entries for any track,  $d$  the time interval between first and last entry in the listening history,  $d(h)$  the time interval between first history entry and entry  $h$  and  $M$  a constant describing the influence of  $m$  on the weight.  $|t(h)|$  does not influence the result,

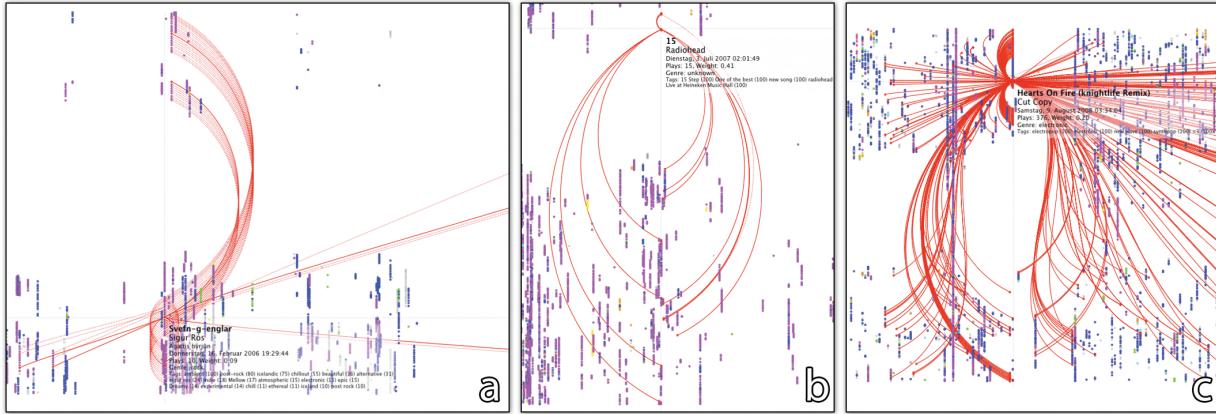


Fig. 4. All instances of a song and its neighbors that appear more than once in a listening history are highlighted while hovering. a) Several longer album sequences b) Repeatedly listening to the same song on one day c) Constant repetition of a single song

but we chose to show the extended version of the formula to make its derivation easier to explain.

The first fraction is the relative importance of a song in a  $P$  day period (we chose thirty days). The second fraction adjusts it for the number of entries for the most popular song, to make sure that long-term favorites have a higher weight than one hit wonders. Finally, the  $m$  factor makes allowance for the fact that older songs have a tendency for higher play counts:  $m$  is in the range of  $1 \pm M/2$ , with a song in the exact middle of the listening history resulting in  $m = 1$ . Therefore, younger songs receive a boost through  $m$  while the weight of older ones is reduced. Based on a sample of listening histories, we used  $M = 0.5$ . The size of the circles is adapted based on this calculated node weight (a minimal weight of 0.01 is the default to keep circles from disappearing). This feature can be activated by the user and is the default setting for the personal mode (see below). Additionally, the lower-left corner of the application shows the number of unique and total songs in the listening history or for the current filter-setting to provide statistical information.

### 4.3 Interaction

The two-dimensional visualization of listening histories already contains a lot of information about temporal as well as song patterns that is accessible at first glance. User interaction is required to receive details about certain parts of the listening history through filtering and zooming and find relevant sequences of songs.

The virtual canvas of songs can be zoomed and panned by using a mouse. Zooming happens with either the mouse scroll wheel or a dedicated zoom slider in the lower right corner of the window. The zoom is only one-dimensional and affects the x-axis, so the user is able to focus on a certain part of the timeline. Zooming does not affect the vertical axis which keeps the day structure as an anchor and prevents the user from feeling lost. The overlap between vertically adjacent songs is tolerable and can be further remedied through interactive exploration. Adding a rectangular marquee for zooming in on two dimensions (similar to timebox widgets [14]) and thus making it easier to focus on crowded sections of the canvas would have been possible, but we decided to use that interaction technique for creating playlists instead (see below). Panning happens by clicking on the canvas and dragging it as far as desired.

The upper right corner of the screen holds a textbox for filtering. The user is able to enter arbitrary filter terms to cause the system to only display songs whose metadata contains these terms. Multiple terms can be combined to further extend the filter (all items containing any of the terms are shown). The system suggests and corrects terms so that they fit the existing properties of the listening history. Also, the user is able to specify the type of the term to, for example, assess that 'Radiohead' means the artist and not a tag or title. In addition to the musical metadata, the filterbox can also be used for temporal

queries: It understands dates ("9h", "July") and periods of time ("Mo-Fr" for weekdays, "Jun-Aug" for summer months). Terms can also be negated, to display all items that do *not* contain them. The displayed number of unique and total songs in the lower-left corner is adapted to the results of the filter.

There are several advantages with this free-form filter approach: First, the filterbox allows creating arbitrary time frames and leaving the restrictions of the pre-defined day-hours cycles of the visualization. Songs from time periods that are of no interest can be hidden. Second, by being able to filter the visualization, more sophisticated hypotheses can be formulated and checked, for example, more music is consumed on the weekend than on weekdays. Third, using the filterbox is optional, which gives the non-expert users the chance to slowly get used to it or ignore it completely if they do not need it. It also resembles a regular text box known from other applications such as search engines and all the underlying boolean logic is well-hidden [16].

**Sequences:** Song sequences within listening histories are an important indicator for the type of music listener. Whether the person meticulously creates their own digital mixtapes or listens to the same song until fatigue sets in, the type of listener is captured in the sequence of songs. As songs are repeated frequently this sequential information is hard to encode graphically without overburdening the visualization, so we decided to make it available interactively. When hovering above a song node, all other instances of this song are graphically highlighted and connected with continuous curves to the original node. To provide the user with information about sequences, for all other instances of a song its predecessors and successors are also compared to the neighborhood of the node in focus. This comparison is sufficiently fuzzy to allow up to four "mistakes", i.e., songs that appear in one neighborhood but not the other. Therefore, sequences where the user for example listened to an album and skipped a certain song are still taken into account. These resulting chains of preceding and successive songs are connected to the focus ones by dashed curves (similar to Arc Diagrams [37]). This leads to several characteristic visual patterns for different types of listeners (see figure 4 for examples).

Instead of, e.g., animating or only highlighting other instances we used curves. Our reasons for that were their aesthetic appeal and the opportunity for the user to find other instances simply by visually following the lines coming from the focus node. For this purpose, curved lines also work better than straight ones [36] and curved lines also do not suffer from occlusion when connecting points along one axis (for example, several instances of the same song on one day, see figure 4b). The curves are in red which has a visual pop-out effect and is also not used to encode a genre. When the user wants to explore the other instances of a song but is afraid to lose track of them when moving the mouse cursor (and thus removing the connecting lines), s/he can press and hold the Shift-key to keep the lines on-screen.

**Detail information and playback:** Hovering above a song node not only shows other instances and sequences but also displays an overlay tool-tip with all available metadata, such as the precise date of the node, artist, title, and album information, play count, node weight, identified genre and user-generated tags (see figure 4). By clicking on a song node, this song is played from the user’s music collection if available (if not, the song shakes to and fro). The user can also draw a box around a certain part of the song nodes to create a longer playlist: All available songs within this box are played and the user is able to listen to music from a certain time period or typical for a certain time of day. Optionally, the songs are not played in their original order but sorted by their frequency within that time with the most popular ones first.

#### 4.4 Additions in Personal Mode

All analysis-based tasks that were mentioned before are still available in personal mode, but there are several additions. To support the reminiscing aspect the most obvious change is the inclusion of photos and calendar entries as memory triggers with the cost of screen space for the listening history. Both photos and calendar entries are shown at the bottom of the screen as additional synchronized timelines, but due to the spatial requirements only one-dimensional (see figure 5).

A common way to organize photos is separating them into events of up to a hundred photos [12]. The existing event structure from the user’s photo software (in our case Apple’s iPhoto, see 4.5) is also used in LastHistory, as well as the corresponding representative photo for more information at first glance. The center of this photo lies at the central date of the event and its size depends on the number of entries in the listening history for the corresponding time period. By hovering above the photo it is enlarged, additional metadata (date, number of photos) is displayed and the remaining photos from the event can be accessed: Internally, the width of the event represents a scale of all contained photos (the left-most border is the first photo, the right-most the last one). The x-position of the mouse cursor on this scale determines which photo is shown. A second way to trigger the user’s memory are calendar entries. Each calendar on the user’s machine gets its own timeline at the bottom of the screen. A calendar event that takes up at least one day (shorter events are less meaningful for the user, see above) is represented by a colored bar filling the time of its duration (the colors are taken from the user’s settings in his or her calendar application). Several calendars (e.g., private and from work) can be shown simultaneously. Calendars work as purely contextual information sources and their description, date, etc. can be shown by hovering above them. The synchronicity between the different timelines helps with reminiscing (see [26]).

One minor change in the personal mode is the activation of node scaling based on their weights (which can also be enabled from the menu). Finally, as a way to turn LastHistory into a background reminiscing tool, it contains a slideshow functionality. By clicking on a photo event a full-screen slideshow with the contained photos is started. Additionally, all songs from the event’s time period are added to the playlist.

#### 4.5 Implementation and restrictions

LastHistory has been implemented in Objective-C for Mac OSX. For an effective access to the contextual data from photos and calendar entries and for making sure that this information was available to a large number of users we decided to tie it in with the default applications (iPhoto, iCal) for these tasks. That, of course, also leads to the restriction of making LastHistory only valuable for users of these applications. Photos and events that were created with a different photo organization software (e.g., Adobe Lightroom) or online (e.g., Flickr) have to be converted and imported creating additional overhead. But as both iPhoto and iCal provide such import functionality it is at least possible.

Listening histories and their accompanying metadata are acquired from Last.fm through their API. When LastHistory is launched for a new user the initialization takes some time because parallel server requests are not supported and there is a short waiting period after each



Fig. 5. In personal mode, songs are enlarged to reflect their importance for a certain time period. Locating listening events in one’s personal narrative is supported with synchronized photos and calendar entries.

request. Two types of information are taken from this online source: the listening histories and user-generated keywords for each unique song contained. These tags are created by the Last.fm user community, are free-form and therefore contain all types of information. Tags also come with a percentage of how often they were used in relation to other tags, which allows estimating their importance.

In LastHistory, tags are shown when hovering above a song but also used as the basis for genre color-coding. For each of the seven included genres we created a list of fitting tags (“funk”, for example, encompasses “soul”, “disco”, “ska”, “reggae”, “worldmusic”, etc). The genre of a song is determined by traversing the list of tags from most to least important and taking the first known tag.

The web-based data retrieval beforehand costs some time but is in its entirety only necessary once (once this initial data set exists it can be updated with the latest changes in a much shorter time). An average listening history with 25,000 songs (7,500 unique songs, six years duration) plus tags takes about 45 minutes to download. Also, there are performance restrictions for large histories with 100,000 songs and above where extracting the song sequences (which happens during runtime) causes larger delays on a regular Macbook Pro.

## 5 EVALUATION

With LastHistory, we mainly aimed at non-expert, average users and had a two-fold evaluation strategy: first, we wanted to figure out in close contact with actual users how they used the application. Second, we also wanted to make sure that our approach of adding photos and calendar entries as memory triggers had any effect on their ability to remember these events and whether they worked as vehicles for story-telling. Naturally, talking about private events in a lab-setting to an experimenter would add an inevitable bias. In order to reduce that, we also made LastHistory available as a free download on our homepage and added a questionnaire that was shown after the user had spent fifteen minutes using the tool. With this approach we aimed to learn whether the use cases we hypothesized indeed existed and were relevant for average people. Also, we wanted to be able to receive unbiased feedback from regular voluntary users working with their own data and find out more about possible flaws and issues.

### 5.1 Case studies

We required our participants to have a listening history with at least 5,000 entries and bring their own photos (calendar entries and MP3s were optional).

**Setup** We recruited four participants from our university (ranging in age from 21 to 27 years, average 24.25, one female) each of which had been using Last.fm for at least 3.5 years. Their listening histories contained between 23,000 and 78,000 (average 45,000) entries and two of them had been using the iCal calendar. Each session was videotaped for later analysis and participants were asked to think aloud for impromptu feedback.

The experimenter gave a short introduction to the application. Participants then used LastHistory on their own to discover usage and patterns in their collection. To guide them along the various features of the application, we asked our participants to initially complete certain tasks, such as "Use the search feature to find your favorite genres and artists" or "Play photo slideshows for one or more of the photo events". They were free to switch between analysis and personal mode and could explore all features.

**Results** We analyzed the resulting videos and communication protocols to observe the participants' insights into their own past and how they interacted with LastHistory. It was interesting to see how patterns in listening histories which we so far could only interpret based on our own assumptions were enriched with personal backgrounds. We categorize the findings according to table 1, i.e., with an increasing level of abstraction and for duration, frequency and periodicity.

Insights regarding the absence or presence of music were mostly based on frequent or periodic patterns and the participants' daily routines: One participant quit listening to music during the week after finishing school and taking a job. Most of her music consumption shifted to the weekends. The former pattern re-appeared only in one week where she was on sick leave. Data gaps were also visible and could mostly be explained by our participants: three of them found gaps where they remembered being on vacation, one forgot to install the Last.fm client on a new computer. A sudden surge in additional history entries could be explained by one participant with him starting to also track music consumption on his mobile player. On the level of individual songs, an increased node size let participants quickly find the relevant ones. A subsequent interactive exploration uncovered in several cases the periodic rediscovery of certain songs. A higher frequency also often coincided with being at a concert by the artist or the release of a new album. In one very extreme case, one of the participants had listened to the same song over 100 times in a row due to the break-up with his girlfriend. Regarding playlists and albums, one participant noted how he used to always listen to the same playlist in the morning before leaving the house, which the visualization made evident by an hour-long single-colored band of nodes in the morning. Finally, for artists and genres, participants were able to find interesting facts and sometimes adjust their self-perception: One participant declared being a big fan of the Red Hot Chili Peppers, but found that he had mostly listened to their music two years ago and only sporadically afterwards. The small number of seven predefined genres meant that the rather uniform histories of two participants held no interesting global insight in this regard. One of the other two noted, though, how "*2009 is color-wise more mixed than 2008*" and another user found that he had mostly been listening to hip-hop in the first year after creating a Last.fm account and was now gradually shifting towards electronica. Grey nodes (unknown genres) filled one user with pride, noting that he often listened to what he called "*underground music*" that had not been assigned any tags on Last.fm. Also, two users referred to periods with pink (metal) nodes in the visualization as their "*hardcore phases*".

## 5.2 Online reception

LastHistory and its source code have been made available online in mid-February this year. In addition, we uploaded an introductory video<sup>10</sup>. We wanted to learn more about how LastHistory worked in real-life with various listening histories and what average users would think of it. We were especially interested in our assumptions regarding the use cases of analysis and reminiscing. We also wanted to test whether the application was really fit as infovis for the masses and could be used and understood.

**Setup** After having been mentioned on several blogs and other online news sources, LastHistory received a fair amount of attention from users and also spread through social networks. At the time of writing (June 2010) we have around 11,500 views for the introductory video, 5,000 downloads of the application itself from the central

repository where we made it available (several other download sites picked it up and put it on their own pages) and 243 completely filled-out questionnaires.

We decided against letting the application record personal data and chose a questionnaire, as we were more interested in the insights users gained than possible usability issues or durations between clicks. LastHistory showed a pop-up message after quitting the application when it had been used for more than fifteen minutes asking the user to fill out said questionnaire. Once the user clicked on the link the pop-up was disabled. The questionnaire itself was deliberately kept very short to entice people to fill it out (155 people left a partial response). It contained some demographical data, asked what data streams users had available (listening histories, photos, calendars), what their primary use of the tool was, how much they liked it and which insights they gained from it. Participants could also leave their Last.fm user name if they liked, which allowed us to find out when they had joined and how long their histories were.

**Results** The requirements for LastHistory, namely, having and using a Last.fm account led to a majority of 56% of participants with a background either in academia or technology and a dominance of male (95.1%) over female (4.9%) users. They ranged in age from 16 to 67 years with an average of 27.2 years. Nearly all participants (99.2%) used the tool for visualizing a listening history (97.1% their own, 4.5% another user's), a smaller number had photos available (37%) and even less calendars (18.5%).

Despite this lack of data for the personal mode, 46.9% used LastHistory for reminiscing and 8.6% for storytelling. When asked if LastHistory helped them with reminiscing, 72% agreed. In contrast to that, only 51.8% of all users said that the explicit personal mode was useful for them. The reason for this was most likely that the personal mode added only little information without photos or calendars available: 68.8% of the users who had photos found it useful and 75.8% who had both (also, these participants used LastHistory 57.6% for reminiscing and 12.1% for storytelling). However, according to the results we concluded that in some cases the listening history itself can already enough to trigger a person's memory. When asked for what they found using the personal mode, answers also represented the gap between data-haves and data-have-nots. They widely varied between descriptions of program errors and complaints about missing data and accounts of reminiscing ("*clicking on a photo gallery and listening to what I was listening to at the time was very powerful*", "*Being able to connect people and places with my listening habits*") and praise ("*I like this mode the best, it should be the default mode*").

The analysis mode was generally more popular (75.7% found it useful) and 76.2% said that they were able to find interesting facts about a listening history. Insights were mainly based on time ("*I rarely listen to music between the hours of 9-11 a.m., even on weekends*", "*Patterns across different academic terms*", "*noted the ... commuting listening pattern*"), and sometimes on musical taste ("*How often I listen to Grizzly Bear!*", "*I listen to Aerosmith around 7pm quite a lot of the time*", "*Those ruts where you get stuck in listening to one particular song*") or problems with the data ("*I listened to music for 4 straight days. Apparently my computer was on and played the music without my knowledge*").

One concern of ours when releasing LastHistory was that regular people would have a hard time learning and using it. But the results from the questionnaire showed the contrary: 75.7% said that they were satisfied with how easy to use and 75.4% with how easy to learn LastHistory was, despite the only guidance being the five minute introductory video. Note that we did not ask what features the participants had used, so it might well be the case that they did not use LastHistory to the full extent.

Suggestions for improvement often discussed the performance of the tool ("*It is very sluggish for me*", "*Speed it up!*"), adding additional features ("*Be able to add more personal feeds*", "*scrobbling from the application*") or more detailed explanations ("*more contextual information about what does what and what things mean*", "*some kind of key about the colours*"). Also, the released version contained a bug that sometimes caused a crash when retrieving the data from

<sup>10</sup><http://www.frederiksciffert.de/lasthistory>

Last.fm for the first time which was also mentioned repeatedly in the suggestions for improvement.

## 6 LASTHISTORY BEYOND MUSIC

Music shares certain characteristics with other types of lifelogging information which makes some of LastHistory's design decisions applicable to other domains. On a very abstract level and as a fallback to section 2, music listening histories are temporal sequences of unique items that possibly repeat themselves. They can also be abstracted through a hierarchical classification (in our example, songs, albums, musical genres, see figure 2). Interesting patterns in this regard are repeating sequences of items (that are potentially predefined in the form of albums) and sections with high and low values for duration, frequency and periodicity. Finally, for this personal information, much more data is locked in the user's memory and can be triggered through contextual cues.

With this abstract distinction of lifelogging data in mind, the visualization approach of LastHistory could be employed in other related domains. An obvious expansion are other media consumption histories (e.g., for movies, TV shows or books) where data is partially already available (eBook readers, for example, log reading behavior). To adapt LastHistory for these other types of data we have to take their attributes into account: Media where single items last longer (e.g., movies, books) have less unique items and also less repetitions. Therefore, it would make sense to replace the circles with bars depicting the item's duration (see section 4.2.4) and show the connecting arcs permanently (as repetitions are few). Adjusting the y-axis for lower frequencies (weeks instead of days) could lead to more insight regarding temporal patterns for these media. Finally, predefined sequences such as chapters in a book or episodes of a TV show could also be highlighted through arcs (we decided to leave arcs out for albums, as it would have contributed to the general visual overload and they were emerging through the sequences anyway).

When going beyond consumption, items actively created by the user such as emails, blog entries or status updates have the advantage that they allow analysis and serve as context at the same time. User-created content is, however, mostly text-based and lacks independent metadata. Extracting this information from the text requires some kind of automatic analysis. Depending on how structured the data is (email messages contain subject and receiver, twitter updates can consist of URLs or other users) different types of sequences (conversations about the same topic, follow-up blog posts) can be extracted. Going one step beyond that, hierarchies can also be extracted from the remaining unstructured text (see, e.g., [27]) to replicate the color-coded genres in LastHistory. An interesting point in this regard is the entanglement between production and consumption in the web 2.0. Reading a status update might lead to an immediate answer in the form of a comment or another status update which blurs the boundaries. Production- and consumption activities could be encoded by distinct glyphs to make the activity surrounding an item visible. Consequently, the complete web browsing history of a user could be integrated (similarly to, e.g., [3]). The question remains, however, how well such an adapted version of LastHistory would fare against explicit visualizations of, for example, email archives [34].

## 7 DISCUSSION & FUTURE WORK

The results from our lab study and the online reception were promising but also showed several areas for improvement. Regarding the positive feedback, we were glad to see our suspicions about the proposed use cases confirmed. People indeed wanted to have access to this information and learn about their own histories. They often shared this information via social networks or photo sharing sites and posted descriptions and explanations online. One very promising future extension of LastHistory therefore would be to integrate explicit features that address this social aspect: in-application sharing of (static) listening histories, connection to an online portal or an in-tool comparison of multiple listening histories.

Adding contextual information in the form of photos and calendars was warmly welcomed by the users. All blogs and media outlets that

discussed LastHistory especially emphasized this combination of multiple information streams. The feedback from the questionnaire also taught us the limits of this approach. In the multi-platform environment of the real world creating an application that works with arbitrary data out of the box is impossible. Nevertheless, people that were able to use our system seemed to like it.

A large part of memories was already triggered by the available context information, but there is always the possibility to add more. Self-created data should be preferred according to the generation effect and is readily available in the form of personal blogs or status updates à la Twitter. Additionally, news from the user's social networks, concerts and album releases, location, weather and so on can also be easily accessed. Regarding the filtering of the data, overlaying different results visually would make it easier to compare them but, of course, also result in an increase in complexity.

One aspect of LastHistory that was almost never mentioned in both lab and online study was the visualization of sequences. The interactive nature of their exploration unfortunately caused a higher overhead for using it which made our participants in the lab study rather use the filterbox than hovering. The discovery of interesting aspects often happened coincidentally and not as part of confirming a hypothesis or reminiscing. For discovering these types of patterns, a different mapping might be more advisable. An explicit sequence-centric visualization approach similar to the *Tangle* visualization could help here [4]. Changing the mapping of the y-axis could also be used for displaying sequences: The most distinct and frequent sequences could be stacked alongside the vertical axis for immediate access. A future version that is more geared towards visualization experts than novice users might also include several other, more sophisticated mappings for the vertical axis, depending on the goal of the analysis.

With novice users, higher sophistication in the available tasks tends to increase the complexity of the interface as well. As we tried to prevent scaring casual users with a complex-looking application, we reduced the number of available filter- and manipulation capabilities to cover the most interesting patterns. This display of raw data and limitations regarding the interaction have, of course, also their downsides, especially as the amount of information is high and the user might not be aware of what to look for. An interesting course for future work could therefore be integrating automatic analysis for the raw data to support users in finding interesting patterns and steering their attention. Having a certain understanding of the nature of the underlying data (see section 2) allows finding such patterns and giving users hints towards their existence. Interesting sequences (heavy repetition of a track, cyclical re-appearance of an album) or patterns regarding the user's taste (a listening session containing only uncommon songs, the connection between artists and seasons) that are not immediately visible could be highlighted. The confidence level of the automatic analysis could be used for prioritizing discovered patterns and deciding whether to point the user towards them. In this way, a middle ground between overburdening the interface with too much hints and concealing interesting facts could be found.

## 8 CONCLUSION

In this paper we presented an overview of listening histories as a personal type of data and additional contextual information in the form of photos and calendars. We also created an interactive visualization for this domain: LastHistory has been evaluated in a lab study and made available online. We discussed the results of these evaluations and the positive reception of the tool by real-world users. LastHistory has been released under an open-source license and is available to all developers who want to work on and extend it. Active discussion groups exist that share feature requests and stories about their findings. We therefore hope that the development of LastHistory is not at an end, that additional features will be implemented and lead to new insights and that users will come back to it in a year to find changes and memories.

## REFERENCES

- [1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data-a systematic view. *Comput. Graph.*, 31(3):401–409, 2007.
- [2] J. Aucourtier and F. Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.
- [3] E. Z. Ayers and J. T. Stasko. Using graphic history in browsing the world wide web. Technical report, Georgia Tech's Institutional Repository, 1995.
- [4] D. Baur and A. Butz. Pulling strings from a tangle: visualizing a personal music listening history. In *Proc. IUI '09*, pages 439–444, New York, NY, USA, 2009. ACM.
- [5] F. Bentley, C. Metcalf, and G. Harboe. Personal vs. commercial content: the similarities between consumer use of photos and music. In *Proc. CHI '06*, pages 667–676, 2006.
- [6] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, pages 1245–1252, 2008.
- [7] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *Proc. UIST '98*, pages 29–38, New York, New York, USA, 1998. ACM Press.
- [8] S. Cunningham, M. Jones, and S. Jones. Organizing digital music for use: an examination of personal music. In *Proc. ISMIR '04*, 2004.
- [9] T. DeNora. *Music in Everyday Life*. University of Exeter, 2000.
- [10] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data : Finding patterns of events across multiple histories. In *Proc. VAST '06*, Baltimore, USA, 2006.
- [11] E. Freeman and S. Fertig. Lifestreams: Organizing your electronic life. *AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval*, 1995.
- [12] D. Frohlich, A. Kuchinsky, C. Pering, A. Don, and S. Ariss. Requirements for photoware. In *Proc. CSCW '02*, pages 166–175, New York, NY, USA, 2002. ACM.
- [13] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. *Proc. JCDL '02*, pages 326–335, 2002.
- [14] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.
- [15] D. Huynh, S. Drucker, P. Baudisch, and C. Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In *CHI '05 extended abstracts*, pages 1937–1940. ACM, 2005.
- [16] B. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227, March 2000.
- [17] H. Kinjo and J. Snodgrass. Does the generation effect occur for pictures? *The American journal of psychology*, 113(1):95–121, January 2000.
- [18] H. Lam and T. Munzner. Increasing the utility of quantitative empirical studies for meta-analysis. In *Proc. BELIV '08*, page 2. ACM, 2008.
- [19] S. Laxman and P. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.
- [20] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. *Computer Mediated Communication*, pages 1–13, 2004.
- [21] W. Müller and H. Schumann. Visualization methods for time-dependent data-an overview. In *Proc. Winter Simulation*, volume 1, pages 737–745, 2003.
- [22] A. C. North, D. J. Hargreaves, and J. J. Hargreaves. Uses of music in everyday life. *Music Perception*, 22(1):41–77, September 2004.
- [23] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behaviour. In *Proc. ISMIR '05*, pages 634–637, 2005.
- [24] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *Proc. CHI '96*, pages 221–227, New York, New York, USA, 1996. ACM Press.
- [25] Z. Poussman, J. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1145–1152, 2007.
- [26] M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz. Milestones in time: The value of landmarks in retrieving information from personal stores. In *Proc. INTERACT '03*, pages 184–191. IOS Press, 2003.
- [27] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA, 1999. ACM.
- [28] A. J. Sellen and S. Whittaker. Beyond total capture: a constructive critique of lifelogging. *Commun. ACM*, 53(5):70–77, 2010.
- [29] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proc. CHI '95*, pages 210–217. ACM, 1995.
- [30] M. Torrens, P. Hertzog, and J. Arcos. Visualizing and exploring personal music libraries. In *Proc. ISMIR '04*, pages 421–424. ISMIR, 2004.
- [31] E. Tulving and D. M. Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352–373, 1973.
- [32] J. Van Wijk and E. Van Selow. Cluster and calendar based visualization of time series data. In *Proc. InfoVis '99*, pages 4–9. IEEE Comput. Soc, 1999.
- [33] F. Viégas, D. Boyd, D. H. Nguyen, J. Potter, and J. Donath. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *Proc. HICSS '04*, pages 1–10, 2004.
- [34] F. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Proc. CHI '06*, pages 979–988. ACM, 2006.
- [35] C. Ware. *Visual Thinking for Design*. Elsevier, 2008.
- [36] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, June 2002.
- [37] M. Wattenberg. Arc diagrams: visualizing structure in strings. In *Proc. InfoVis '02*, pages 110–116. IEEE Comput. Soc, 2002.
- [38] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Proc. InfoVis '01*, page 7, 2001.