# Adaptive Timeline Interface to Personal History Data

Antti Ajanki, Markus Koskela, Jorma Laaksonen and Samuel Kaski*
Department of Information and Computer Science
Aalto University School of Science
PO Box 15400, FI-00076 AALTO, Finland
firstname.lastname@aalto.fi

## ABSTRACT

As the growth of stored personal digital information, such as photographs and emails, is continuously increasing, new tools for browsing and searching are needed. We introduce an intelligent mobile information access tool for personal data. The data are presented in an adaptive timeline where the displayed items function as search cues. The novelty is that the visualization is dynamically changed to emphasize relevant items, which makes them easier to recognize and select. The relevance is inferred during usage of the system from user feedback. In a user study, the dynamic timeline-based interface on a mobile device was shown to require less effort than conventional textual search.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process,relevance feedback*;
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*interaction styles,evaluation/methodology*

## General Terms

Algorithms, Experimentation, Performance

## Keywords

Human-computer interfaces; information retrieval; intelligent systems; machine learning; mobile applications; relevance feedback

## 1. INTRODUCTION

The increasing pervasiveness of mobile computers and digital recording devices in daily life has made it easy to capture and store a lot of data in digital form. Consequently, the sizes of digital photograph collections, textual document archives, lifelog recordings and other forms of personal digital histories are growing rapidly. A new breed of search engines aim to make it possible to search personal information collections without having to first manually organize everything [7]. New kinds of retrieval tools are required to effectively find data items from such collections.

We introduce a *personal history browser* (PHB) mobile application for recording and browsing events, images, notes and other personal data. The novelty is that the interface intelligently emphasizes potentially relevant items on a timeline, so that an item can easily be recognized and selected for further inspection. The purpose is to support fast retrieval of heterogeneous personal data. For example, PHB can help retrieve notes the user wrote in a particular spatial location, or find an article she remembers reading after a discussion with a particular person.

When the personal history is extensive, all of the items cannot simultaneously fit on the screen. The standard approach is to list some fixed number of most relevant items. We propose a novel variant of this approach: we allocate the screen space in proportion to the predicted relevance of each item. The sizes of the items change dynamically during a search session when new relevance feedback is received.

If items are shown in a meaningful order, as is the case with personal history items along a timeline, the emphasized items have another important role as *search cues*. If the user remembers temporal relationships between the target item and some displayed items, she can focus or limit the search or browsing in a matching region of the timeline [23]. People tend to use approximate dates as navigation anchors, for example, when searching family photograph collections [33]. By showing only a few salient items at a time, we hope to lower the cognitive load required to recognize the items. We have previously explored a similar approach in a desktop setting [1].

In our our PHB interface, data items are primarily represented as thumbnail images. The purpose is to allow quickly spotting familiar images. Humans are known to be to able to recognize large numbers of previously seen images [28]. To ensure general applicability of the method, we base the relevance predictions on a *multi-domain* representation of the history items. The items can consist of associated images, videos, textual notes, or other descriptions recorded during the same calendar event or otherwise close by in time. Our system learns a separate *ranking function* on each domain and combines them to produce the final prediction.

Because a PHB is inherently a mobile application, it can be used to record events during daily life. Mobility is also

*A. Ajanki and S. Kaski also with Helsinki Institute for Information Technology HIIT, and S. Kaski is also with Department of Computer Science, University of Helsinki.


*ICMI'13,* December 9–12, 2013, Sydney, Australia

http://dx.doi.org/10.1145/2522848.2522870.

important for searching, because one is likely to find information when needed, if the browser is handily available in a mobile form. However, because the recorded data can be stored in the cloud, the retrieval functionality is available also on desktop computers.

We conducted a user study to compare the dynamic timeline of our PHB prototype to a more conventional textual search in a news story retrieval task. The test subjects were able to find relevant items with less effort using our interface.

The remainder of this paper is organized as follows. First, we review related research on search interfaces for personal data in Section 2. Section 3 describes features that we find essential for a PHB application. The results of the user study are reported in Section 4. Finally, we provide concluding remarks in Section 5.

## 2. RELATED WORK

In the past few years there has been an increasing interest in storing emails, visited web pages and other digital artifacts as well as image or video recordings of events in one's life for later retrieval, recollection or analysis [9, 26]. Such *lifelogs* can be captured by instrumenting computer applications to record interactions [22] or by using wearable cameras that automatically take pictures at certain intervals [14]. In this paper, our main focus is on the retrieval of information from lifelogs with less emphasis on the capturing. The remainder of this section gives an overview of previous work on retrieval tools and interfaces for personal information collections.

Time ordering is a natural way to organize personal data for browsing, and many previous works have introduced intuitive interfaces that place items along a timeline (e.g. Stuff I've seen [11]). Time-based visualization encourages utilizing remembered temporal relationships between items as search cues. [2] proposed visualizing textual search results on a timeline to facilitate exploration and understanding of the temporal relationships of the documents. Harada et al. [13] found out that a mobile zoomable timeline-based photo browser performed at least as well as a traditional hierarchy-based interface that required additional effort from the user to categorize the photos. iRemember [32] was a mobile audio capture and retrieval tool. Users were able to retrieve audio recordings with a combination of phonetic search and timeline navigation. The iCLIPS search tool [6] for personal photo collections combined a timeline and a versatile filtering system.

Browsing the data is easier if one can recognize familiar items and use them as cues on what is being displayed. Several researchers have proposed to annotate data visualizations with various search cues. Ringel et al. [23] displayed personal data on a timeline aligned with selected public news headlines, personal calendar events, and photographs. The cues were meant for helping in quickly localizing oneself on the timeline. They reported that searching with the cues enabled was significantly faster than without them. A lifelog can also be annotated with user-generated content from image and video sharing web sites [10].

Another way to provide memory cues is to emphasize important items in the dataset. Lee et al. [20] showed images sized proportionally to their visual uniqueness on the assumption that unique images are better search cues than commonly occurring views. The PhotoMemory interface [12] grew the sizes of the images that matched visual, spatial, temporal or other filter values set by the user. The thumbnail sizes in the earlier applications were static. We, in contrast, build upon our earlier work [1] and dynamically grow or reduce images according to the relevance feedback collected during browsing.

Many personal information retrieval tools have utilized faceted search interfaces that allow querying and filtering both by content and by rich metadata depending on what the user remembers [8, 11, 30]. Faceted search works well when the user remembers several aspects of the target items and is thus able to restrict the number of results. In some cases, however, the user cannot or does not want to specify the query in detail up front either because she remembers only few details or because she finds constructing a complex query too laborious. The use of relevance feedback in our approach encourages more browsing with the query evolving during the search session.

Another approach is to use a structured, context-sensitive information management application which constantly captures the user's context as unobtrusively as possible [25]. This allows proactive retrieval of relevant information items from the user's personal knowledge space. For example, in the CONTASK task management system, the user's interaction is tracked and used to model the performed tasks to facilitate proactive task prediction [22].

Our conception of the personal history browser supports two types of search behavior: keyword-based search and browsing relevant items on a timeline. The option to navigate by other means than keyword search is important. People often prefer browsing over direct search, sometimes also even when the target is known. A well-known example of this is traversing file system hierarchies on personal computers [4]. Navigation is preferred especially when the users cannot fully articulate what they are looking for, as proceeding towards the target in a series of small steps is cognitively less demanding than coming up with perfect keywords [5]. Furthermore, stepwise navigation helps to understand the context of the answer and its relations to other results [29]. Another reason to treat the search as a sequence of steps is that the information need may change during a search session [3]. The information need evolves because the searcher focuses or broadens the scope after seeing some results, or because new associations pop into her mind.

Relevance feedback [24] is commonly used to improve an estimate of the user's preferences in information retrieval. Relevance feedback is an iterative process where the searcher enters an initial query and the search engine returns a set of items. The searcher can then mark some items as relevant or not relevant, and ask the search engine to retrieve more items. The next set of items selected by the engine will then be similar to the initial query and the items marked as relevant. The process continues until the searcher finds a satisfactory answer. Our approach employs relevance feedback in deciding which items should be shown saliently.

## 3. PERSONAL HISTORY BROWSER APPLICATION

In our vision, a PHB application consists of two main operational parts: the history recording subsystem and the browsing and searching subsystem. The latter can further be seen as constituted of two subunits: the user interface and the actual algorithms used for providing the user with the

retrieval results. These subsystems can to a large extent be independently designed, and we will discuss each separately below.

## 3.1 History Recording

The aim of the history recording subsystem is to collect available data about the activities and interactions the user is involved with, together with contextual information. The collected information will then be used for two purposes: Firstly, it will be presented to the user as her personal history. Secondly, it will be used as hidden or visible search cues when the user decides to browse her recordings.

Actions for recording the personal history can be either *explicit* or *implicit*. Explicit actions could mean systematically creating images and documents and storing them in a well-defined collection structure, or using a special diary-like application for that purpose. Implicit personal history creation would be more based on the device's automatic processes. Not only self-taken images and self-written or self-selected textual documents would be stored, but also all digital objects the user has read, seen or heard in the real world, in the Internet and in emails and other messages, together with calendar data, GPS coordinates and other locational and contextual data.

The implicit or automatic collection of the personal history can in practice be implemented with an application that is autonomously working as a background process in the mobile device. If a PHB application's input devices include passive recording devices such as the SenseCam [14] or a head-worn wearable camera, their outputs could be directly utilized. It would be beneficial, however, if the application could also *proactively* encourage the user to enrich her history data, for example by asking her to take a photograph in a place where she — according to her previously recorded personal history — has not earlier visited.

An effective history recording application needs to have at least the following two *roles* that are carrying out their tasks in parallel. The first one is the collection and storing task and can be implemented by hooking the application as a set of action callbacks in the process loop of the device's operating system. The second role involves offline processing of the collected data. An important subtask here is that of *linking* related information items together. Relatedness can arise e.g. from calendar data by which the documents written and read and images taken and seen during one calendar event can be linked to constitute one multi-domain information item.

The computationally intensive processing of the data can be implemented partly locally in the mobile device, but more likely as a backend server process in the cloud. Server-implemented background analysis processes could include face detection and recognition, as well as object and location recognition, as all of these require both computational resources and reference data that cannot practically be available or fit in a mobile device. An additional benefit from using a cloud service backend is that the personal history data will then also be available for heterogeneous platforms and a number of devices instead of the single device that has been used to record it.

## 3.2 Browser Interface

In our approach, a PHB application contains two main interface components: the textual search functionality together with a list view for the search results, and the timeline supporting free browsing. The relative sizes of the items on the timeline are dynamically updated based on the current estimate of the relevances. Both components also support closer inspection of any of the shown items and entering of positive or negative relevance feedback for the retrieval algorithms.

A query session in a PHB can be instigated either by an explicit search action of the user, such as entering keywords or giving feedback, or by requesting items related to the current context. In the latter mode, the system *proactively* suggests related information, immediately when the mobile device is activated, with minimal effort by the user. The proactive mode requires that the system tracks the environment with the mobile device's location and other sensors and uses the sensor values as a query.

We consider textual or faceted search as the primary starting point for explicit retrieval. With a textual or faceted query, the user is assumed to be able to guide the search roughly into the correct direction. After the initial search, the user has then two options: either to refine the query due to shortcomings in the retrieval results, or to continue the session using relevance feedback. An alternative approach is to initiate the search with feedback. This is generally preferential for the most recent history items only, as they are readily discoverable from the timeline and for them the user can be assumed to have a fresh memory trace.

The precondition for a dynamic updating of the relevances during user interaction brings severe time requirements for the retrieval algorithms and data communication. The updates of the display should happen in real-time with minimal delays after any user feedback. Therefore, the processing of the queries should be performed on the client device as much as is pragmatical. This can partly be implemented using local caching of the recent history recording and previous search results.

Finally, the application has to support editing of the timeline items with corresponding updates made to the server databases. This includes the possibility of entering corrections to results of automatic processes such as that of linking related information items.

## 3.3 Retrieval Algorithms

Personal digital histories consist of many types of data, such as photographs, emails, time and location and other sensory data. Therefore, a search engine for personal history must be able to search or filter over multiple domains. This is in contrast to web search engines, where pure keyword searches have been the standard practice. In our approach, the information domains are connected by co-occurrences; for example, an image may contain GPS coordinates, or several emails may have been linked to the same calendar event.

From the user's point of view, it is important to be able to switch between different search criteria at any time, because people may remember different aspects of the item they are looking for [8]. One might, for example, remember that a topic was discussed in a meeting with particular persons and want to use that as a search key to find details about the topic in the meeting notes. The user interface should therefore allow feedback on any type of data, and the retrieval engine should integrate multi-domain feedback into one consistent view.

**Table 1: Components of the PHB application (rows) enabled in the three interface variants (columns)**

| | Textual baseline | Relevance order | Timeline order |
|---|---|---|---|
| Keyword search | X | X | X |
| Textual relevance feedback | X | X | X |
| Visual relevance feedback | | X | X |
| List/image mode switch | | X | X |
| Images in relevance order | | X | |
| Images in time order | | | X |

There exists different strategies to combine multi-domain feedback. Perhaps the simplest option is *filtering*: the user is asked to provide a range of acceptable values on the different domains and items matching the values are returned. Another option is *multiview learning*, in which feature representations from the different views are projected into a common space and a single ranking function is learned there. A third approach, which we take in Section 4, is to learn a separate ranking function on each domain and combine the individual rankings by a *rank aggregation* operation.

# 4. EXPERIMENTS

In these experiments, we focused on testing the browser subsystem which is the pivotal part of any PHB system. To this end, we conducted a user study to evaluate the benefits of using dynamical timeline browsing, compared to a more traditional textual search interface.

## 4.1 Experimental Procedure

The experiments consisted of search tasks conducted on the three variants of PHB summarized in Table 1. A baseline variant had a purely textual search interface without a dynamic browser. We refer to this as the *textual baseline*. We compared it against a variant with a dynamic *timeline*. To quantify how much of the difference between the baseline and the dynamic timeline variants is contributed by the ability to browse by time information, we constructed a third variant that replaces the time ordering of the images with a *relevance* ordering.

The task of the participants was to find information about or images related to a given topic (see Table 2 for a list) with the help of one of the interface variants. They were instructed to "bookmark" items they considered to be related to the topic by pushing a button that marks the items as relevant. Another button was available for marking items as non-relevant. The subjects were further instructed that, in case they cannot immediately find highly relevant images, they can guide the system towards more relevant items by marking also only marginally relevant items.

A task was finished either when the test subject indicated that she had found five (in her opinion) good examples or after five minutes. Each test subject completed the same nine search tasks. The allocation of interfaces and topics was balanced so that each subject used each interface equally many times and that each topic was completed with each interface equally many times. This was realized by using a Latin square experimental design with the test subjects as rows, the topics as columns and the interfaces as treatments. The presentation order of the tasks across the test subjects was also balanced.


**Figure 1: Prototype dynamic timeline interface in our PHB system. The textual search box is at the top, the list of the most relevant items with relevance feedback buttons in the middle, and the scrollable view of time-ordered images scaled by their relevance at the bottom. Relevance feedback for the items can be given with the plus and minus controls.**

Before the actual experiments, the functionality of system was explained to the test subjects during a training phase, where the test subjects completed one or more training tasks using each of the interface variants. The experiment took about 80 minutes in total including about 20 minutes for the training phase.

## 4.2 Participants and Apparatus

Nine test subjects were recruited (seven male, two female, aged 24–32); they were doctoral students in a computer science department.

All variants were implemented on a Nokia N900 mobile phone with a 3.5 inch touchscreen and a physical keyboard for text input. Figure 1 shows a screenshot of the timeline interface variant. The bottom of the screenshot shows the scrollable view of time-ordered images with sizes proportional to the predicted relevance. The images whose sizes would be below a threshold are hidden completely and replaced with a small marker (ellipsis) because they are predicted to be irrelevant. Scrolling the timeline component doubles its height for a short period of time making the thumbnails large enough to be recognizable [16]. The top part of the screen shows a keyword query input box and the most relevant data items in a list. In the timeline and relevance variants the top of the screen can be switched to show one selected image in large size (this mode is not shown in the screenshot). Full text of an item can be opened by clicking a text snippet.

## 4.3 Relevance Estimation

A central constituent of our PHB interface are the image sizes that change according to relevance estimates. We estimated the relevance by combining three rankings: one static ranking was generated by keyword search, and two dynamic rankings were estimated by considering the relevance feedback separately on visual and textual feature spaces. This subsection gives details on the three ranking functions and their combination.

The keyword search was implemented by using the Lucene search engine[1]. Lucene scores the text documents according to how similar they are to the query in a vector space. Both the query and the documents are represented as TF-IDF (term frequency, inverse document frequency) vectors. Stemming was applied to account for inflections. Documents that contain many informative terms similar to the query are ranked first.

The effect of relevance feedback on the texts was captured by a separate ranking function. A ridge regression [15] was trained in the TF-IDF space. The documents marked as relevant were used as positive and non-relevant documents as negative training examples. The documents were then ranked according to the scores given by the learned regression function. The value of the regularization parameter was optimized beforehand on separate training queries.

Visual ranking values were calculated by using our on-line PicSOM content-based image retrieval (CBIR) system[2] [19]. The system uses the two-dimensional surface of the Self-Organizing Map (SOM) [18] as a model of the latent space for storing the image relevance information. Three visual feature spaces were used: 5000-dimensional, dense-sampled *color SIFT* with two-level spatial pyramid [21, 31], 1302-dimensional *census transform-based texture feature* [34] and 256-dimensional *scalable color descriptor* from the MPEG-7 standard [17]. This selection of visual features has been found to perform well in visual similarity based search tasks in our earlier CBIR studies (e.g. [27]). Processing time for the images consist mainly of the feature extraction stage which takes approximately one second per image. Images that were visually the most similar, in all three feature spaces, to the images marked as relevant by the user, were given the highest ranks in the visual ranking.

The three rankings were combined into a single ranking using the inverse rank position aggregation. It orders the items (indexed by $n$) in the decreasing order of $s_n$, the sum of the inverse rank position in the constituent lists:

$$s_n = \frac{1}{r_n^{(\text{keyword})}} + \frac{1}{r_n^{(\text{textualFB})}} + \frac{1}{r_n^{(\text{visualFB})}}, \qquad (1)$$

where $r_n^{(\text{keyword})}$ is the rank of item $n$ in the keyword based Lucene ranking, $r_n^{(\text{textualFB})}$ is the rank in the textual feedback ranking, and $r_n^{(\text{visualFB})}$ is the rank in the PicSOM visual feedback ranking.

This rank-based fusion approach has been found efficient in our preliminary fusion studies. The procedure depends only on the ranks, not on the actual score values assigned to the items by the base rankers. Therefore, the procedure is generally applicable for any base rankers, even producing scores on different scales as in our case. To set the image sizes on the timeline we scaled the scores $s_n$ linearly into the range $[0, 1]$.

Figure 2 shows first the timeline before any feedback and then how the images on the timeline have been rescaled after the system has received new relevance feedback. After an image has been marked as relevant, the updated rankings and the $s_n$ value were computed for all items, and the sizes of the images were allocated in proportion to the corresponding items' $s_n$ values.

[1]http://lucene.apache.org/
[2]http://picsom.ics.aalto.fi/picsom

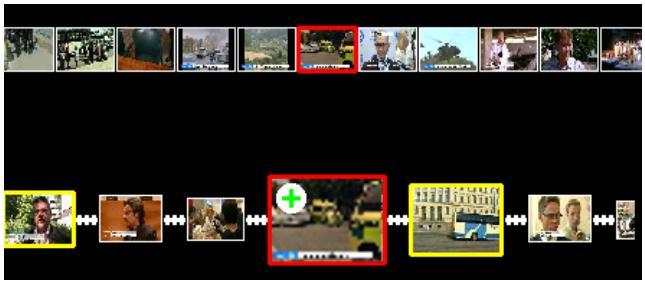


**Figure 2: Image sizes change when relevance feedback is given to the items. The upper row shows the timeline before feedback is given. The lower row shows changed image sizes after the selected image (the one with red borders) is marked as relevant. Images similar to the selected one are made larger and dissimilar ones are made smaller or hidden with an ellipsis.**

## 4.4 Corpus and Tasks

To be able to compare the interface variants with multiple users, we used a news corpus as a "shared history" in the experiments. The assumption is that the test subjects remember at least a rough time course of news events and are able to direct their search using what they remember, as they would when browsing real personal data. While the multimodal nature of real personal digital histories is not fully portrayed in such a shared history, it is our aim to focus in this paper on the retrieval of relevant items using a timeline interface. We assert that for this purpose using a common corpus is justifiable.

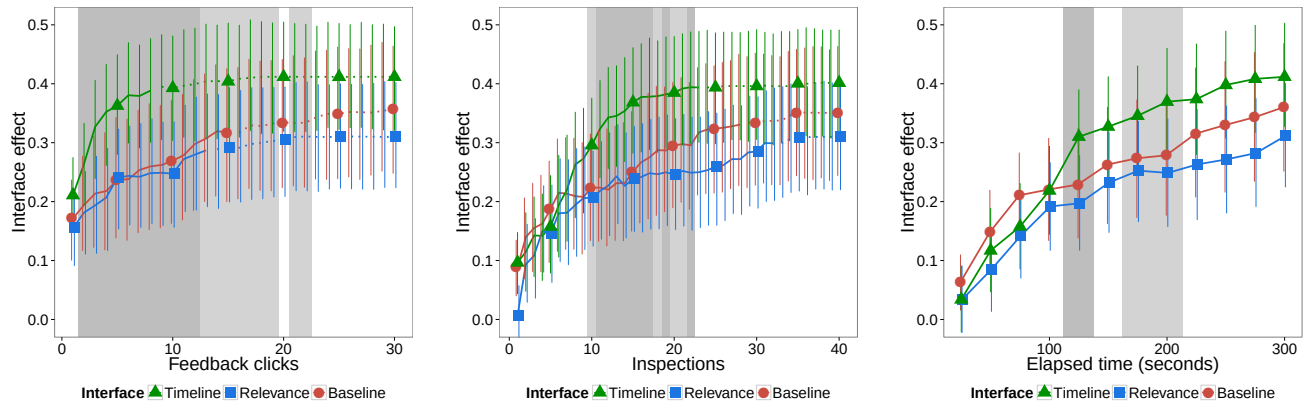
The corpus that was used in the experiment was an archive of daily TV news broadcasts from the Finnish Broadcasting Company YLE. The dataset included 3893 news events from the main evening news broadcasts of each day of 2011. One keyframe was extracted as a visual representation of each news event. Each news event was accompanied by one paragraph of text extracted from the broadcaster's archival system and describing the event.

The test subjects were asked to find images describing given topics. The topics were major news stories 2011 that were covered extensively in the dataset. The nine topics were selected beforehand by browsing the news archive for a diverse set of news stories. The tasks were presented as short textual descriptions shown in Table 2. Because we assume that, when browsing personal histories, people use the time of events as a major search cue we included a time dependency in most tasks.

## 4.5 Results

The performance of the interfaces was compared by evaluating the goodness of the rankings of the data items as returned by the system while the test subjects were interacting with the application. We used a standard information retrieval performance measure average precision (AP) to measure the goodness of a ranking. The average precision is the average of the precisions at the relevant ranks:

$$\text{AP} = \frac{1}{R} \sum_{n=1}^{R} \frac{n}{r_n}, \qquad (2)$$




(a) (b) (c)

**Figure 3: ANOVA estimates and 95% confidence intervals for the interface variant specific effects as a function of (a) the number of feedback clicks, (b) the number of inspected items, and (c) the elapsed time. In all plots, the higher values are the better and the shadings indicate the regions where the timeline interface performs significantly better than the baseline interface ($p \leq 0.05$ for dark, $p \leq 0.1$ for light gray region). The dotted lines mark the region where more than 2/3 of the search tasks have finished.**

**Table 2: The nine search tasks in the experiment**

| | |
|---|---|
| 1. | Early stages of the 2011 revolts in the Arab world |
| 2. | A moment in nature during summer time |
| 3. | Major moments in European financial crisis during the first half of 2011 |
| 4. | First signs of resolution in Finnish government negotiations |
| 5. | News items anticipating a positive upturn in the economy |
| 6. | First items of news related to spring |
| 7. | Scandals related to politicians during the second half of 2011 |
| 8. | Events in Egypt after the downfall of President Mubarak |
| 9. | Problems at Nokia before the Microsoft collaboration was announced |

where $R$ is the total number of relevant items, and the $r_n$ are the ranks of the true positive items such that $r_n < r_{n+1}$. A higher AP value is better and means that the test subject has found many relevant items so far and the system's current ranking contains relevant items at top positions.

Computing AP requires known relevance values for the news events. The ground truth was obtained by an expert judge, who double-blindly considered all news items that any of the test subjects had marked as positive, and either accepted them as true positives or marked them as false positives, which were treated as non-relevant in the subsequent analysis.

Optimally, we would like to measure AP as a function of the cumulative mental effort spent during the search task. Because the true mental effort is not directly observable, we use three *proxies*: *the amount of feedback* given, *the number of inspected items*, and the elapsed *time*. An item was considered inspected when either an image was selected on the timeline or a full text was opened in the search result list. Because we cannot directly measure how much attention the test subjects paid to the text snippets that were not opened in the list, we assumed that the search list is scanned sequentially and considered all items above an opened one as inspected, as well. The second proxy variable considers AP at the instants when an item was marked as relevant or not relevant. The third variable, the elapsed time, is not directly proportional to the mental effort because there was no time pressure to complete the task, other than the five minute limit.

Each of the nine test subjects completed three tasks with each interface totaling 27 tasks per interface variant. The number of feedback clicks was highest on the textual baseline interface (15.0 clicks on average with the standard deviation of 11.6), with the timeline interface (12.0 clicks, sd 7.6) following closely. The smallest amount of feedback was given on the relevance interface (7.8 clicks, sd 4.4). The total time to complete the task was on average similar on all interface: 260 seconds (sd 55) on the timeline interface, 270 seconds (sd 48) on the relevance interface, and 250 seconds (sd 56) on the textual baseline. The highest average precision at the end of a task was achieved on the timeline interface (0.41, sd 0.25). The average precisions for the relevance and the textual baseline interfaces were 0.31 (sd 0.20) and 0.36 (sd 0.23), respectively.

To separate the effect of the interface variant from any task and subject specific effects, we analyzed the AP values obtained from (2) with the following three-way repeated measures ANOVA:

$$\mathrm{AP}_{ijkt} = \mu_t + \alpha_{it} + \beta_{jt} + \gamma_{kt} + \epsilon_{ijkt}, \tag{3}$$

where $\mathrm{AP}_{ijkt}$ is the observed average precision for the combination of the interface variant $i$, subject $j$ and task $k$ computed at time $t$. $\mu_t$ is an intercept, $\alpha_{it}$ is a fixed effect term for the interface variant $i$, $\beta_{jt} \sim N(0, \sigma^2_{\beta t})$ is a random effect term for the subject $j$, $\gamma_{kt} \sim N(0, \sigma^2_{\gamma t})$ is a random effect term for the task $k$ and $\epsilon_{ijkt} \sim N(0, \sigma^2_t)$ is a noise term. We overload the notation so that the index $t$ may refer to any of

the three proxy variables described above, not just the time. The assumption of normality of the residuals was checked by graphical inspection of the normal probability plot.

The estimated interface variant effects $\alpha_{it}$ for the three interfaces $i$ (lines) and the three proxy variables $t$ (subplots) are shown in Figure 3. All the three plots indicate that the dynamic timeline interface performs better than the textual search baseline at least on a certain interval. The effect is clearest as a function of feedback clicks in Figure 3a, which shows that the users of the timeline interface were able to give feedback more effectively.

Figures 3b and 3c show all interfaces performing equally at the beginning and the performance of the timeline variant increasing faster with both the amount of inspections and the elapsed time. Figure 3a shows a gap already in the beginning because the timeline attracted more browsing before the first feedbacks were given. The tails of 3a and 3b show the lines leveling off as some users finish tasks earlier than others. The dotted line regions denote the intervals where more than 2/3 of the search sessions have already finished. In particular, the users of the timeline interface hit the time limit after fewer feedback interactions than the users of the two other variants.

The relevance ordered variant performs about equally or slightly worse than the baseline on all measures. Based on this finding, we argue that the majority of the performance improvement we observed in the timeline case is because the timeline-based interface is more suitable for effective navigation. In other words, the searchers are able to utilize temporal relationships when such relationships are presented. On the other hand, we argue that the performance of the visual similarity ranking algorithm does not have a crucial effect on the relative performance of the relevance ordered variant. Improving the visual ranking subsystem would thus benefit also the performance of the timeline-based interface.

After completing the search tasks, the test subjects were asked in a questionnaire to rank the three interface variants in the order of their usefulness during the experiment. Most (7 out of 9) test subjects preferred the timeline over the baseline. One of the two who preferred the baseline was mostly using only the textual search even when the timeline or relevance ordering was available, because he found it the most familiar.

# 5. CONCLUSIONS

In this paper, we have presented our view of a personal history browser, a mobile application aimed for automatic collection of digital memories and for their intelligent retrieval. We think that the intelligent browser is one of the pivotal components for the success of such a memory assistant. Consequently, we have focused our effort in studying and evaluating novel methods for incorporating computational intelligence in information access.

The methodological novelty in this work has been the presentation of the information items on a dynamically updating timeline where the sizes of the displayed image thumbnails are changing to reflect their estimated relevance to the search task at that moment. In our approach, we assume that the recorded personal history is made of multi-domain information items, typically consisting of textual and visual documents together with contextual metadata such as location information. The relevance was predicted based on learning separate ranking functions for textual and visual contents and then combining the rankings with a rank aggregation operation.

Controlled user experiments were carried out by using a simulated personal history, common to all test subjects and constructed from an archive of TV news broadcasts of one whole year. Nine different search tasks were instructed among the total of nearly 3900 history items in the stored material. The evaluation setup allowed us to assess the utility of the proposed dynamic timeline approach to personal history browsing when compared to the more conventional text-only and simple relevance-ordered search strategies. The results of the small-scale experiments in a controlled setting proved that the dynamic timeline interface was significantly more effective than and preferred over the reference methods. Further experiments with real recorded personal data are needed for more comprehensive conclusions about the generalizability of the results to multimodal personal digital histories.

# 6. ACKNOWLEDGEMENTS

This work was financially supported by The Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170), Aalto MIDE programme (project UI-ART), and EIT ICT Labs (Activity 11863 Crosslinking Visual Information).

# 7. REFERENCES

[1] A. Ajanki and S. Kaski. Probabilistic proactive timeline browser. In *Proc. 21st International Conference on Artificial Neural Networks (ICANN), Part II*, pages 357–364, 2011.

[2] O. Alonso, R. Baeza-Yates, and M. Gertz. Exploratory search using timelines. In *SIGCHI Workshop on Exploratory Search and HCI Workshop*, pages 23–26, 2007.

[3] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989.

[4] O. Bergman, R. Beyth-Marom, R. Nachmias, N. Gradovitch, and S. Whittaker. Improved search engines and navigation preference in personal information management. *ACM Transactions on Information Systems*, 26(4):20:1–20:24, Oct. 2008.

[5] O. Bergman, M. Tene-Rubinstein, and J. Shalom. The use of attention resources in navigation versus search. *Personal and Ubiquitous Computing*, 17(3):583–590, 2012.

[6] Y. Chen and G. J. F. Jones. Augmenting human memory using personal lifelogs. In *Proc. 1st Augmented Human International Conference Article*, pages 24:1–24:9, 2010.

[7] E. Cutrell, S. T. Dumais, and J. Teevan. Searching to eliminate personal information management. *Communications of the ACM*, 49(1):58–64, 2006.

[8] E. Cutrell, D. Robbins, S. Dumais, and R. Sarin. Fast, flexible filtering with Phlat. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 261–270, 2006.

[9] M. Czerwinski, D. W. Gage, J. Gemmell, C. C. Marshall, M. A. Pérez-Quiñones, M. M. Skeels, and T. Catarci. Digital memories in an era of ubiquitous

computing and abundant storage. *Communications of the ACM*, 49(1):44–50, 2006.

[10] A. R. Doherty and A. F. Smeaton. Automatically augmenting lifelog events using pervasively generated content from millions of people. *Sensors*, 10(3):1423–1446, 2010.

[11] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: A system for personal information retrieval and re-use. In *Proc. 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, 2003.

[12] D. Elsweiler, I. Ruthven, and C. Jones. Towards memory supporting personal information management tools. *Journal of the American Society for Information Science and Technology*, 58(7):924–946, 2007.

[13] S. Harada, M. Naaman, Y. J. Song, Q. Wang, and A. Paepcke. Lost in memories: interacting with photo collections on PDAs. In *Proc. 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 325–333, 2004.

[14] S. Hodges, E. Berry, and K. Wood. SenseCam: A wearable camera that stimulates and rehabilitates autobiographical memory. *Memory*, 19(7):685–696, 2011.

[15] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[16] W. Hürst, C. G. M. Snoek, W.-J. Spoel, and M. Tomin. Size matters! How thumbnail number, size, and motion influence mobile video retrieval. In *Advances in Multimedia Modeling: 17th International Multimedia Modeling Conference*, pages 230–240, 2011.

[17] ISO/IEC. Information technology - Multimedia content description interface - Part 3: Visual, 2002. 15938-3:2002(E).

[18] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, Germany, third edition, 2001.

[19] J. Laaksonen, M. Koskela, and E. Oja. Class distributions on SOM surfaces for feature extraction and object retrieval. *Neural Networks*, 17(8-9):1121–1133, October-November 2004.

[20] H. Lee, A. F. Smeaton, N. O'Connor, G. Jones, M. Blighe, D. Byrne, A. Doherty, and C. Gurrin. Constructing a SenseCam visual diary as a media process. *Multimedia Systems Journal*, 14(6):341–349, 2008.

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[22] H. Maus, S. Schwarz, J. Haas, and A. Dengel. CONTASK: Context-sensitive task assistance in the semantic desktop. In *Proc. 12th International Conference on Enterprise Information Systems (ICEIS 2010)*, pages 177–192, Funchal-Madeira, Portugal, June 2011.

[23] 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 2003, the Ninth IFIP TC13 International Conference on HCI*, pages 184–191, 2003.

[24] G. Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, NJ, 1971.

[25] S. Schwarz. A context model for personal knowledge management applications. In *Proc. Second International Workshop on Modeling and Retrieval of Context*, pages 18–33, 2006.

[26] A. J. Sellen and S. Whittaker. Beyond total capture: A constructive critique of lifelogging. *Communications of the ACM*, 53(5):70–77, 2010.

[27] M. Sjöberg, S. Ishikawa, M. Koskela, J. Laaksonen, and E. Oja. PicSOM experiments in TRECVID 2012. In *Proceedings of the TRECVID 2012 Workshop*, Gaithersburg, MD, USA, November 2012.

[28] L. Standing. Learning 10,000 pictures. *The Quarterly Journal of Experimental Psychology*, 25:207–222, 1973.

[29] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proc. SIGCHI conference on Human factors in computing systems*, pages 415–422, 2004.

[30] D. Tunkelang. *Faceted Search*. Synthesis lectures on information concepts, retrieval, and services. Morgan & Claypool Publishers, Palo Alto, CA, 2009.

[31] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proc. of IEEE CVPR 2008*, Anchorage, Alaska, USA, June 2008.

[32] S. Vemuri, C. Schmandt, and W. Bender. iRemember: a personal long-term memory prosthesis. In *Proc. 3rd ACM workshop on Continuous archival and retrival of personal experiences (CARPE)*, pages 65–74, 2006.

[33] S. Whittaker, O. Bergman, and P. Clough. Easy on that trigger dad: a study of long term family photo retrieval. *Personal and Ubiquitous Computing*, 14(1):31–43, 2010.

[34] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, 2011.