

Facebook Bytes: Creating Your Unwritten Autobiography

Akshay Kothari, Ankit Gupta and Albert Lai

Abstract—We describe a visualization for Facebook data that was designed to present a unified view of a person's digital life. The information presented consists of text (status messages and wall posts) and photographs (events and tagged photos) that make up a user's interactions with his/her friends. There are two levels of view - an 18 month summary and a single month view. The 18 month view provides a month by month presentation of aggregate statistics for a user's activity. Drilling down into a month reveals the user's actions plotted on a timeline.

Index Terms—Timelines, Temporal Visualization, Social Data, etc .

◆

INTRODUCTION

Actively documenting your life using media such as blogs and photo albums requires a lot of time and effort; in the past, people have used journals and scrapbooks to record their memories for posterity. However, many people now use social networking sites like Facebook, where their activities are recorded through regular use.

In addition to personal data, relationships with friends are also captured through wall posts and tagged photographs. We aim to visualize these events temporally to provide a compelling glimpse into one's personal history. Browsing this data when it is accurately represented on the time axis tells a very personal story.

We chose Facebook as our source of data because it has over 350 million users, many years of recorded data, and also provides a well documented API to retrieve one's data. There are several types of data representing certain actions one can make in Facebook, each with a timestamp: *Statuses* are short blurbs that users write to describe what they are doing at that time. *Wall posts* are messages users write on their friend's profile page (also known as a 'wall') and vice versa. *Photos* of users can be tagged if he/she appears in it. *Events* are created and hosted by users. They have a description and an associated image.

These actions result in two types of data - text and images. In order to present this content in an intuitive manner, both a summary view and individual view for the data were implemented.

The aggregate view is a familiar looking bar chart detailing the amount of activity per month, whereas the individual view reveals the actual content of messages or images posted on Facebook. In this mode, each piece of data has to be placed on the timeline at its corresponding time without overlaps. With a mixture of text and photos, we've attempted to recreate the look of a scrapbook while trying to maintain readability.

Facebook itself presents user data in a vertical list format with limited browsing capabilities. Here we describe a system used to enable easy exploration and trend identification between a user and his/her friends

1 RELATED WORK

There has been a lot of work in storing and visualizing life events in the corporate research world as well as in the academic research world. The MyLifeBits [1] project at

Microsoft has developed software, which leverages SQL server to store all personal information found in PCs. It initially focused on capturing and storing scanned and encoded archival material e.g., articles, books, music, photos and video as well as everthing born digital e.g. office documents, email, digital photos. It has evolved to have a goal of stroing everything that could be captured. The MyLifeBits shell, which is the main user interface, allows queries to be viewed as a list, variable sized thumbnails and a timeline. It enables refinement or pivoting according to metadata and links, and provides for creation of text and voice comments.

The EveryBit project at the Nokia Research Center is a similar project, with a focus on mobile devices. It is a community-focused web-based filesystem, which supports data archiving, search and retrieval. It also allows users to publish and share their data with others. This service is still under field tests within the Nokia Research Center-Palo Alto group.

There have also been a lot of timeline-based visualizations. The perspective wall [2] was an early attempt to develop techniques, which allowed access and management of large information spaces, and thus supported large-scale cognition. It used hardware support for 3D interactive animation to turn 2D timeline layouts into 3D visualizations that imitate the human eye and integrate details and context.

The spiral wall [3] focused on the serial periodic data, which occurs frequently. Arranging data in a spiral provided the user with visual cues to both serial and periodic aspects of the data, along with interactions such as the change in period over time. In some ways, it was comparable to trees (which display hierarchies) or tables (which display data categories on two axes).

Lifelines [4] is an interactive visualization tool for visualizing temporal categorical data across multiple electronic medical records. It advocates a framework of simple operators such as Align, Rank and Filter to allow users to manipulate multiple records simultaneously to understand relative temporal relationships across records.

TimelineCompare [5] is a tool that allows the user to dynamically select two different timelines and display them in a manner that would facilitate the discovery of synchronicities or the mapping of a known domain to unfamiliar material.

Inspiration was also found in news websites, particularly New York Times and BBC. The latter's timeline on

“Persecution and Genocide under the Nazis” is worth noting for its slideshow feature.

2 FACEBOOK BYTES DESCRIPTION

Facebook Bytes is a temporal visualization of facebook data, including user’s statuses, wall posts, photos and events on a timeline. The visualization not only allows the user to reflect on his/her facebook usage in terms of monthly analytics, but also enables him/her to go back to a particular time in history and visualize their life events in more detail. The following sub-sections explain the motivation for the project and data retrieval process.

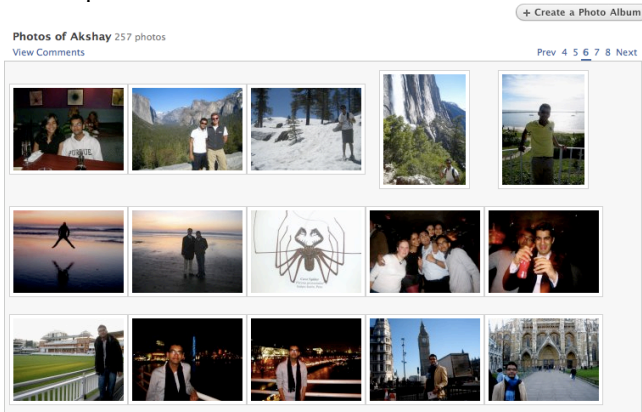


Fig 1: Facebook Photo Browsing Interface shows only 15 photos at a time

2.1 Motivation

A lot of people document their life events in their own ways. Some write their thoughts using a pen and a diary, while some blog on their own website. Some upload pictures on Flickr, while some others maintain it on their hard drive. And then there are people, who wished they had an up-to-date blog or a flickr page, but “just didn’t find time” to maintain it.

Knowingly or unknowingly, a lot of us have already been documenting our lives on Facebook. It contains a great amount of rich social information. It provides several years

facebook Bytes

CLEAR >

- Events
- Images
- Statuses
- Posts on my wall
- My posts on friends' wall



Fig 2: Bird's Eye View

worth of text communication along with rich media such as photos and videos. For early adopters, facebook contains several years of this data.

Now, a person who maintains a personal diary or scrapbook can quickly flip over the pages to go back to her high school days. Skimming over the diary can lead to discovery of rich emotional moments of her life. Unfortunately, Facebook along with most other digital diaries (including blogs, flickr) does not allow the user to see his/her past activity in a meaningful emotional manner. In order to see communication with a particular friend, a facebook user has to keep clicking “next” for every 10 messages exchanged. Similarly, for a user who has 300 photos, it will take at least 5 clicks to find his/her first image. (Fig 1)

Facebook Bytes visualizes your life on the social network on a rich interactive timeline, to enable discovery of interesting moments of your life that have been documented. The unique panning interface allows you to go back in time with a single click.

2.2 Data Retrieval

Facebook contains an Application Programming Interface (API) that allows an application to retrieve data from it. Once a user authorizes the application, it can access the user’s data till it remains authorized. We used the Facebook Query Language (FQL) to access users’ status messages, events, photos they are tagged in and wall posts. Facebook provides a timestamp for each activity that was essential to retrieve its time and display it on a timeline.

3 DESIGN OF FACEBOOK BYTES

This section describes individual elements of the visualization’s design in detail.

3.1 Timeline

Our goal with Facebook Bytes was to build a visualization system that end-users of facebook could use to create an autobiography of their life with their online content. Since time is the most important dimension in this goal, a timeline is the core of our visualization. A timeline allowed us the flexibility to zoom in to show a Bird’s Eye View (section 3.2) and a Zoomed in 31-Day View (section 3.3) by simply

changing the position and labels of marks. The timeline also allowed a user to stay anchored in a concrete timeframe while navigating through the system using various interaction mechanisms.

3.2 Exploration: Bird's Eye View

To provide a perspective for large time frames, we chose to aggregate the data for each month. The total number of statuses, wall posts, events, and pictures are plotted at each interval. Each item is represented by a colored square, where each color corresponds to the type of data. Squares are stacked on their respective months, creating a bar chart.

Because there has to be a differentiation between wall posts made by the user on their friends' profiles and posts written on the user's wall, we used the direction in the y axis to encode the target. The user's post on other people's walls is shown as a bar in the negative Y-axis, whereas posts on the user's wall appear on the positive axis. Statuses and tagged photos are also in the positive direction since these objects can be interpreted as being posted on the user's wall.

This view (Fig 2) provides a quick way to identify which time periods were particularly active by spotting the months that contain spikes. Moreover, the balance between inbound and outbound actions can be observed by comparing the length of the bars in the positive versus negative Y direction.

An alternative to plotting the bars in opposite directions is to have everything be positive. By doing this, it is easier to compare whether a user sent or received more

posts since the bars will appear side-by-side. However, having five bars per month, each representing a different type of activity, hampered the ability to focus on just one type of activity. By putting wall posts on friends' walls alone in the negative direction, this category becomes very salient while reducing the number of different categories to discern between in the positive direction.

Once the user decides to zoom in on a month, the summary view moves to the bottom of the visualization to act as a map for the person's life. The bars are scaled down in the Y direction, but the colors and placement remain the same. Thus, months with large amounts of activity are still visible so a user can jump straight to interesting months easily if they want to see the month with the most pictures, events, or posts. The current month is designated by a rectangle overlay. This view covers eighteen months because it provides enough separation between the groups of bars to indicate that they represent the same month. Additionally, the time frame is long enough to show some long term trends. For example, activity tends to have many peaks and valleys; holidays and birthdays tend to pick up a lot of activity, as do the start and end of a school year.

3.3 Exploration: Zoomed In 31-Day View

We picked a 31-day window to provide a zoomed in view of a smaller time frame. This (Fig 3) allows a user to look at a month's content at once enabling him/her to easily view the activity in the period as well as draw patterns and conclusions from it.

Our first challenge was to pick visual representations

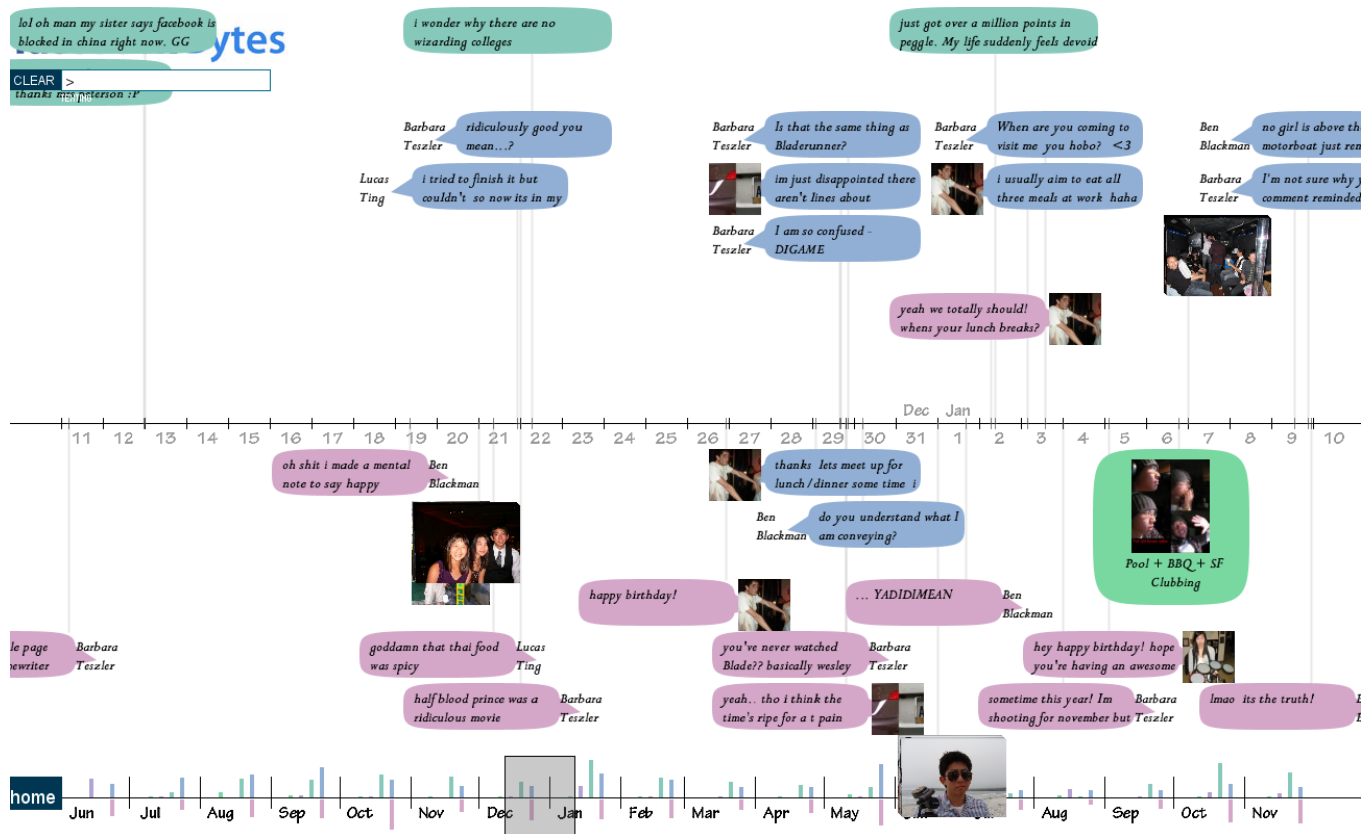


Fig 3: Zoomed in 31-Day View

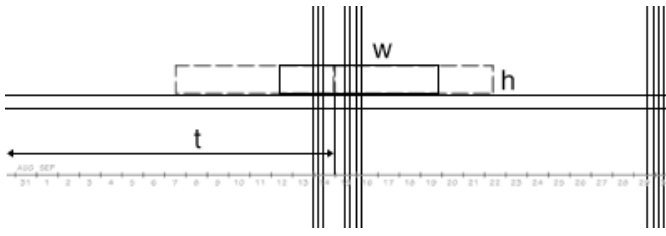


Fig 4: Possible positions of an object

of each type of content such that they "feel" real. We chose a balloon metaphor for data objects with the view that activities on facebook are like content filled balloons added to one's life. Status updates are thus represented as curvy text blobs. Continuing the balloon metaphor, post to ones wall by a friend are represented as a chat bubbles originating from the friend. This was achieved by placing a square 50x50 thumbnail of the profile picture of the friend (or just their name if the picture wasn't accessible) to the left of the text blob. Similarly, posts to a friend's wall were represented as chat bubbles going into the friend. This was achieved by placing the thumbnail or name of the friend to the right of the text blob. We identified that a good representation of an event can simply consist of its date, image and name to aid recollection. Since we already display the objects on a timeline, we represented an event as a balloon with its name below the image. Tagged images on facebook are always associated with a particular album. Since they are mostly uploaded in a group, several of them need to be displayed at the same spot. We chose a stacked pile of images as their visual representation (with each album having its own stack) denoting multiple images at the time instance.

Our next challenge was to devise a layout algorithm that renders the objects at their right place without overlaps. Layout of objects is, in general an NP Complete problem [6]. We devised several heuristics to achieve a good enough layout in a short amount of time. Our algorithm started with splitting the screen into a grid of size 1200 x 700 with each block being of size 10x25. This allowed us more control along the x axis (time) since real estate was quite limited on the time axis. We also split the object into two categories: *Visual* (Events and Photos) and *Textual* (Status, Posts). We placed each object at a node in the grid. The algorithm starts with the first day in the 31-day window and assigns a location to all objects in the day before doing the same for the next day. In the following discussing, (say) an object had a width w , height h and was supposed to be connected to the time axis at $x = t$ (Fig 4)

We required our system to produce a final layout that was both visually appealing as well as looked like a scrapbook of one's life. Towards this goal, our algorithm scatters the *Visual* objects somewhat randomly in the y direction. The statuses are favored to appear in at the top of the visualization.

We start by first finding a place for the *Visual* objects. To make the placement of these objects seemingly random (as in a scrapbook), we assign a random value to the initial y coordinate and $t-w/2$ to the initial x coordinate. If the object overlaps with some other object already displayed, we

try a different position by decreasing the x coordinate by 10. We check for overlaps again. We keep repeating this process till the x coordinate becomes less than $t - w$. At this point, we reset the x coordinate to $t - w/2$ and started increasing the x coordinate by 10. We continued this process till the x coordinate became t ; after which, we reset the x coordinate to $t-w/2$ and increase the y coordinate by 25. If the y coordinate goes out of the visualization, we set the y coordinate to 100 and continue the process. After each new position, we check for overlaps with existing objects. If we find a position without any overlaps, we display the object at that position. If we can't find any such position, we discard the object.

We then find a position for status messages. Since we wanted to status messages to appear mostly at the top, we initially place the status message at $\{t-w/2, 0\}$. If it overlaps with existing objects, we try to find a better x coordinate with a similar approach as above. If that doesn't remove the overlap, we increase the y coordinate by 25. We keep repeating this process till we either find a position without overlaps or the y coordinate forces it off the visualization.

For finding the position of a wall post, the algorithm is very similar to the status messages. The only difference is that for posts that the user has written on a friend's wall, the initial y estimate is the bottom of the visualization with next estimates decreasing the y coordinate.

For posts that a friend has written on the user's wall, the initial y estimate is 100 with the next estimates increasing the y coordinate. This continues the view of inbound posts being mostly above the timeline and outbound posts mostly below from the Bird's Eye View.

Once each object's position is determined, a very light gray line is drawn from the timeline to it. Since the text gets clipped if it overflows its blob, we picked the sizes of the objects manually to allow for a sizeable number of content to be displayed in the view without much loss.

3.4 Basic Interaction

Basic Interaction is done with simple mouse gestures. One of the main ways of exploring the content is via hovering. Hovering over a stack of images (Fig 5) blows it up showing a bigger size of each image arranged in a compact fashion. Hovering over an event expands the blob adding its venue and tagline to it. Hovering over a text blob expands it showing the full text. Hovering over a post also displays the name of the sender/receiver below the text blob. Hovering over any object fades all other objects emphasizing the hover action. It also emphasizes the

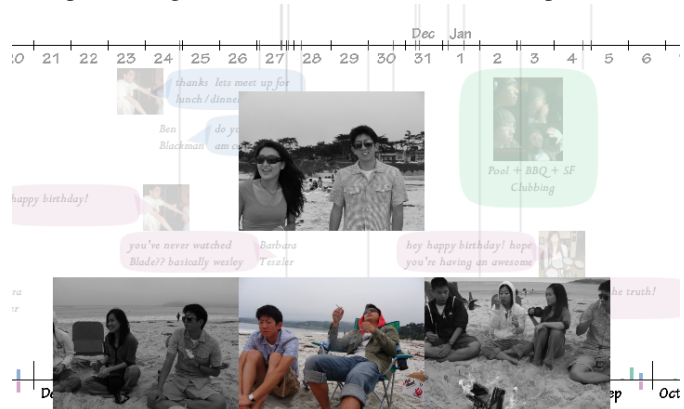


Fig 5: Hover expands images in album

line connecting it to the timeline by increasing its width. This mechanism allows users to get more information about the particular object as well better understand its position on the timeline; increasing the scope of the visualization while maintaining the context of other content in the background. The system smoothly fades the objects on hover and smoothly unfades the objects as well when the hover is released, thus preserving the fluidity of the design.

Mouse click on a wall post filters the content from the particular friend involved in the post. This function is the same as searching by the friend's name, as described in the next section. This allows the user to actively pursue his/her interest in a looking at all interactions with a friend with a direct manipulation based interaction.

3.5 Search

The ability to view how one's relationship with another person evolves over time provides some unique insights on one's life. Whether certain friends drifted away, became closer, or barely talked at all, mapping the frequency of communication spatially makes these observations manifest.

To facilitate this type of explorations, we created a search box to filter the visualization by friend to reveal a back and forth conversation between wall posts. The name submitted must be an exact match for the name of the friend on Facebook, but changes can be made so that the search could be case insensitive, or even auto-completed.

For the 18 month view, results are filtered (Fig 6) by coloring

the squares that correspond to the queried friend red and reducing the alpha for all other boxes. This provides a quick representation of the user's interaction with another person, showing where the hot spots of activity are. Since all of the unfiltered squares are the same color, plotting the wall posts made by the user in the negative Y direction makes a clear distinction between posts to and from a user.

In the one month view, all activity not related to the friend in question is not shown. For consistency, the layout of the posts are not recalculated so the only transition between a filtered and unfiltered view is the fading of irrelevant activity. Since the 18 month view is pushed to the bottom when viewing a single month, the red coloring of the friend's posts provide landmarks for browsing, allowing the user to quickly find where posts to and from that friend is while scrolling.

Filtering is applied by performing a basic string comparison between the query and the name in the "to" or "from" field for a data object. Strings that match will be considered a friend interaction and those that do not are faded out. The string must perfectly match the friend's name on Facebook, which may be a hassle. The query is also compared to a post's "to" and "from" field every time the post is redrawn; as an optimization, a visibility flag could be implemented. Additionally, a user can click on a post in the 31 day view to set the query as well as the search box text to that post's corresponding name. This will have the same effect as typing that person's name in the search box, but allows a user to filter while browsing if he or she comes across a post by a person

facebook Bytes

CLEAR Jeffrey Modlin

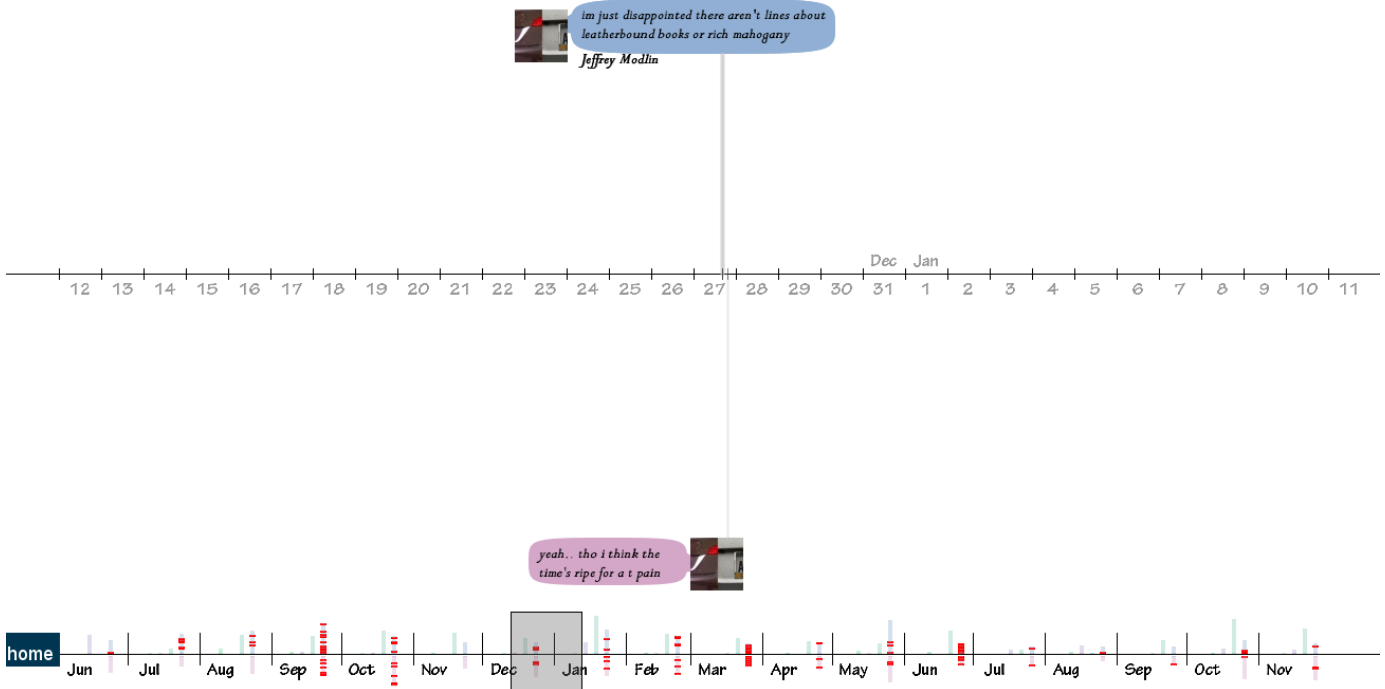


Fig 6: 31-Day Filtered View

of interest. To remove a filter, the user must click the "clear" button next to the search box.

3.6 Navigation

Navigation through the visualization was designed to be as intuitive as possible. In the 18 month view, when the mouse is over a particular month, the month is highlighted by a dark gray background to indicate which month would be viewed if the user clicks. Clicking will then expand the view and present the 31 day perspective for the selected month.

While in the monthly view, the user can scroll forward and backwards within the 18 months of the visualization by using the miniature 18 month view on the bottom or by clicking on the main view and dragging the timeline left or right. By dragging left, the content is moved to the left, essentially progressing forward in time and vice versa.

When using the 18 month view to scroll, the rectangle (Fig 7) that denotes what days are currently being shown is also used to navigate. One can click on the rectangle and drag it around to scroll forward or backwards in time.

An unfortunate result of these two navigation methods is that when the user clicks on the main view and drags left, the visualization moves forward in time. However when the user clicks on the scrolling rectangle and drags left, the visualization moves backwards. While this could be confusing, the behavior makes sense within the context of where the user is clicking.

3.7 Information Scent

Although Bytes primarily focuses on effectively visualizing one's content, the design also encapsulates several visual cues to provide information scents to aid a user's understanding of his/her online behavior. Scented Widgets [7] talks about providing social activity cues to improve social data analysis in a social visualization. Information Foraging Theory [6] provides a more general view in support of this argument by claiming that information scents help a user in the sense-making phase of information consumption.

Each object in the 31 day window is connected to its location on the timeline (representing the hour of post) by a

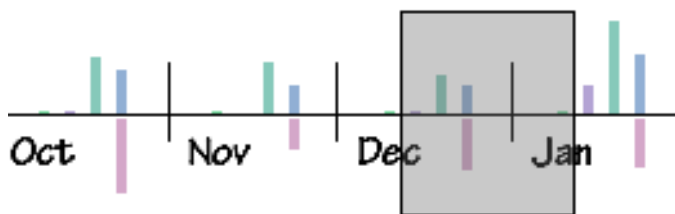


Fig 7: Pan by clicking and dragging the region

thin semi-transparent gray line. The frequency of the lines allows a user to understand the distribution of activity in the time frame. Since the lines are semi transparent, overlapping

lines increases the alpha, thus encoding stronger activity with darker color. In addition to providing the context of time, the overall timeline at the bottom of the 31-day view also contains the bars of activity from the 18-month view. They allow a user to jump to periods of interest based on the information perceived from them.

3.8 Aesthetics

Based on user feedback, the layout, animation and functionality were iterated several times to make it look aesthetically pleasing. Although overlapping content might have allowed us to put more information on the timeline, we chose to give each bubble its own space for a cleaner look. Each text bubble was also restricted to only 2 lines so that a long post does not take over the timeline. Hovering on top of it reveals the full post.

Since pure text visualization tends to take significantly more cognitive power, a higher priority was given to photos and events. To cater to the scrapbook feel of the visualization, the photos were placed randomly on the Y-axis. The panning interface was tweaked a few times so that it mirrors the user's ability to flip through their book. The current implementation allows the user to hold and scroll from anywhere on the timeline.

A great deal of importance was given to color. Using resources such as ColorBrewer [8], we picked a combination of 4 colors that were distinct enough to figure out the narrow bars in the bird's eye view, and at the same time look well saturated in the 31-day window. Finally, different fonts were tried to strike a balance between legibility and cursive handwriting.

4 IMPLEMENTATION NOTES

Facebook Bytes was written in Processing, utilizing its capability of image, shape and text rendering, string utility functions, image download and Java data structures. We used Processing's asynchronous image download to improve the visualization's response. Images were only downloaded when required. We also used the ControlP5 [8] library in Processing for buttons and the search box.

Facebook's Application Programming Interface was used to access a user's data. We implemented FQL queries in a ruby script to download the data in temporary comma separated files to be then read by processing. To support day-wise layout and rendering, each object was stored in a hashmap with the day being its key.

5 USAGE OBSERVATIONS

The response to the visualization was generally very positive, with many expressing the wish to view their own personal data; however due to the time consuming nature of the data retrieval, only the test user's data was available to explore.

The filtering of friends seemed to have a bigger impact on the 18 month view since one can quickly scan the image and immediately see the nature of the relationship. In the 31 day perspective, a lot of scrolling is needed to view the data.

The hovering behavior was well received since it allows inspection of individual items. The lines connecting a piece of data to the timeline also made the visualization more clear because the actual sequence of events can be ambiguous based off only the position. Using the images of a friend on both posts by and to that person was confusing even with color distinguishing the two actions. The presence of the image lead people to believe that the friend was the actor in both scenarios.

The general layout of the 31 day view was seen as easy to navigate, not too cluttered, and aesthetically pleasing at the cost of not rendering anything that does not fit. This runs into trouble when a user is searching for a friend, but the post isn't rendered due to layout space. They can see the red mark on the 18 month view, indicating a message should be there, but since the post isn't rendered nothing shows up on the screen.

People already familiar with Facebook seemed to interact more naturally with the visualization while those who did not. The legend uses Facebook vocabulary of "wall post" and "statuses" which may not be immediately clear to anyone new to Facebook.

Clicking on the main view to pan sometimes resulted in mistaken filtering when the user clicks on a wall post. There was some confusion regarding the 18 month view. Since there were 4 bars per month, some assumed each bar corresponded to a week instead of each representing a particular data type. The legend is not present when the 18 month view moves to the bottom of the 31 day view, so there is no clear indication that each color maps to a type.

6 FUTURE WORK AND CONCLUSION

The layout algorithm can be greatly improved by utilizing a metric to rank an item's importance. Posts can have comments and 'likes,' which indicate a post's popularity. By using this information, the posts with the most comments or 'likes' can be rendered first.

Additionally, when something is not rendered, there could be an indication on the timeline that there is activity there, but it simply isn't rendered. The content can then be revealed on a click or mouse over. The 31 day view would end up being a better representation of the activity frequency for that month with this addition. The visualization can also benefit by running the Facebook API calls live to retrieve the data instead of downloading it all beforehand.

Facebook contains a lot of data about people's personal lives that usually gets lost once it leaves the user's front page. By retrieving this data and presenting it in a manner that can be browsed, users can look back on their past and reflect on their relationships with others. Our visualization allows users to easily go through their personal data and explore their past, spotting when they were most active and who they might have lost touch with. People reacted positively to the information being presented in a scrapbook-like manner and our approach can be extended to other sources of data as well, for example Twitter.

Acknowledgments

The authors wish to thank Jeffrey Heer and Michael Bostock for their helpful comments and suggestions throughout the entire quarter. Their critique on projects and homeworks is especially appreciated. The authors also wish to thank all students of CS448b for their valuable comments and suggestions during the course of the project.

REFERENCES

- [1] Jim Gemmell, Gordon Bell and Roger Lueder, MyLifeBits: a personal database for everything, Communications of the ACM, vol. 49, Issue 1 (Jan 2006), pp. 88-95.
- [2] Jock D. Mackinlay, George G. Robertson, Stuart K. Card: The perspective wall: detail and context smoothly integrated, Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, p. 173-176
- [3] John V. Carlis and Joseph A. Konstan: Interactive Visualization of Serial Periodic Data, UIST'98, the 11 annual Symposium on User Interface Software and Technology
- [4] Taowei David Wang, Catherine Plaisant, Ben Schneiderman: *Lifelines2: Discovering Temporal Categorical Patterns Across Multiple Records*
- [5] Minakshi Mukherjee, Owen Otto and Lois Wei: TimelineCompare - <http://www.raccoonandzebra.com/timelinecompare/>
- [6] D.S Johnson, A. Demers, J.D. Ullman, M.R Garey and R.L Graham. Worst Case Performance Bounds For Simple One Dimensional Packing Algorithms, SIAM J. Computing Vol. 3, No. 4, Dec 1974.
- [7] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *Transactions on Visualization and Computer Graphics* In Transactions on Visualization and Computer Graphics, Vol. 13, No. 6. (2007), pp. 1129-1136.
- [8] Color Brewer <http://www.colorbrewer2.org>