1 Motivation

The Twitter online social networking site is a relatively new phenomenon that nevertheless attracts many attempts at visualization. The concept of Twitter is simple and elegant: a fusion of microblogging and status updates. Users post what they think and do, and see the posts of their friends. Links between people on Twitter are bidirectional, which means that while I may be interested in you, you may not even be aware of me. One of the motivations of this work is to use the tools of social network analysis to determine the strength of ties between Twitter users.

The network is asymmetrical in another way. Twitter users tend to use the service in different ways. Some users never post a single update of their own, but use the site as an aggregator to keep in touch with their friends. Some users are only interested in recording their own thoughts, and may even disable public viewing of their tweets. Some users struggle to increase their own follower count, whether for social capital or some other reason. A basic metric we can apply here is, does the user have more followers or more friends? The main motivation of this visualization is to be able to see at a glance what kind of behavior is exhibited by users related to yourself in the network. Who is a producer of information, and who is a consumer? In the terminology of marine biology, who is a source and who is a sink?

2 Audience

The design of the visualization is meant primarily to provide extra information that is not immediately obvious by reading through user pages on Twitter. On the other hand, it should be possible to use the visualization for at least one of Twitter’s primary purposes, that is, to read peoples’ tweets. The audience of this visualization would mainly be interested in seeing the structure and overall properties of the network in relation to themselves, instead of using the visualization as their primary interface.
3 Related Work

For the analysis of tie strength, this program relied on research which predicts tie strength based on readily available social data on Facebook [5]. Many of the variables used by Facebook are not available on Twitter, but two of the most important variables did translate and were used by the visualization’s algorithm. Previous visualizations of Twitter have focused on network structure, browsing, statistics, search, and filtering. Here are some of the notable examples.

The “Twitter Friends Network Browser” is a web-based Twitter browser. Beginning with a root user, you can click on related users to expand their friends. The profile image and latest tweet of each user is shown. A series of links can be navigated quickly, based on people the user knows or interesting tweets to follow up on. However, no work is done to try to determine the strength of the ties between users, and no other statistics are visible in this view. However, it does demonstrate excellent usability and design.

Twittervision is a Google Maps mashup which places users on a map by locating them either based on biographical information or coordinates and places extracted from the tweets themselves. This visualization is an example of mapping coordinate space to physical space.

Tweetstats.com is a site that allows you to visualize a number of statistics for a Twitter account. Number of tweets per day, reply statistics, number of friends or followers over time, and other statistics are all plotted. Some of the same statistics are useful for visualizing activity in the network. A strength of this visualization is the ability to view changing statistics over time, something which my program cannot do.

TweetPad is a visualization of incoming activity from one’s friends on Twitter. It graphically represents the post application (web, SMS, etc.), the number of words and characters of the post, and other statistics. It allows the user to interact with the data in new ways. The interface is entirely mouse-driven, which makes it easy and fun to use. This visualization used similar technologies (Processing) and interface techniques to this work.

4 Design

My visualization’s design is roughly broken into two parts: to display statistical information, and to display real-time Twitter information. The right-hand side of the screen is able to display the last few tweets from a selected user, as well as show their profile picture. In addition, it is possible to see biographical information about a user by hovering over them in the main window. The window itself is able to be dragged around with the mouse, which allows the user to pan around a large graph. The interface elements are all drawn with transparency, so that the main graph structure is never entirely invisible or occluded.

The central element of the graph is the node, which represents Twitter users. Beginning at the center of the graph is the user’s avatar, represented by a circle.
The circles in this visualization convey several statistics at once. The total number of friends and followers of a user is directly related to the area of the entire circle. Each circle is randomly assigned a color to distinguish it from others. Inside this main circle is a smaller white circle, whose area corresponds to just the number of friends that user has. Friends are people whom the user is following Twitter. Therefore, the larger the inner circle with respect to the outer, the fewer followers the user has. Therefore, the more white you see in a circle, the less that user is acting as a “source.”

Another visualized statistic is frequency of posts. This number is calculated based on the length of time the user has been on Twitter and their total number of updates. The circles onscreen seem to “pulse” with a frequency that is directly related to their post frequency; more prolific posters beat faster. It is easy to compare the activity level of users by comparing the beating of their circles.

Next, whenever a friend/follower relationship exists between two users, a line is drawn between them indicating a tie exists. The width of the line is proportional to the strength of the tie, as determined by a simple algorithm based on the number and time since communications between the users. The links are drawn semi-transparent, to make crossing ties less confusing.

Finally, the visualization attempts to indicate who is paying attention to what updates. The inner circle of a user’s node, which represents the consumption side of their identity, can be filled with the color of the nodes they pay attention to. When a circle pulses, small bursts of their color travel outwards.
along the links, and when they reach a destination (a follower node), they flash the node’s interior color to their own. That color fades fairly rapidly, which should graphically indicate that the information is “consumed.”

5 Implementation

All the code for this project was written in the Processing language. The scene elements were simply drawn with graphical primitives like circles and rectangles. For access to twitter, the program included the twitter4j library, a Java library which wraps the public Twitter API.

The only computationally and resource intense part of the program was determining tie strength. For this, each user in the graph must be queried on Twitter for their last 100 tweets. Fortunately, Twitter was kind enough to lift the rate limit for this project. After the tweets have been pulled, each is scanned for references to other users of the form “@username.” When a new reference is found, the program records the number of days since that reference was made. It then calculates the total number of times the user has communicated with the referenced user. Finally, the strength of the tie is calculated as a linear combination of the number of references (a positive influence) and the number of days since the latest reference (a negative influence), plus a constant.

6 Conclusion

The visualization was very successful at indicating the overall structure and activity of a portion of the network related to the central user. It was immediately clear who was producing most of the content on the network, and who produced almost none. The structure of the graph, combined with the flashing of the circles, conveyed a lot of information at once. The movement of bursts and pulsing of circles especially lent itself to a vivid, almost living look. Navigating and manipulating the space was fairly easy, and the sidebar and hover panes were not intrusive. If anything, the level of visual activity may have been too high, as sometimes it was distracting when focusing on a particular part of the graph. Also, because of the high resource consumption, it was not possible to dynamically expand the graph, which would have been nice.
References