# Vizster: Visualizing Online Social Networks

Jeffrey Heer
Computer Science Division
University of California, Berkeley

danah boyd
School of Information Management and Systems
University of California, Berkeley

## ABSTRACT

Recent years have witnessed the dramatic popularity of online social networking services, in which millions of members publicly articulate mutual "friendship" relations. Guided by ethnographic research of these online communities, we have designed and implemented a visualization system for playful end-user exploration and navigation of large-scale online social networks. Our design builds upon familiar node-link network layouts to contribute techniques for exploring connectivity in large graph structures, supporting visual search and analysis, and automatically identifying and visualizing community structures. Both public installation and controlled studies of the system provide evidence of the system's usability, capacity for facilitating discovery, and potential for fun and engaged social activity.

**CR Categories and Subject Descriptors:** H.5 [User Interfaces] Graphical User Interfaces, I.3 [Methodology and Techniques] Interaction Techniques, K.8 [Personal Computing]
**Keywords:** social networks, visualization, graphs, community, data mining, exploration, play

## 1    INTRODUCTION

The advent of the Internet has given rise to many forms of online sociality, including e-mail, Usenet, instant messaging, blogging, and online dating services. In 2003, another form of online community acquired stunning popularity: online social networking services. In addition to descriptive personal profiles, members of such communities publicly articulate mutual "friendship" links with other members, creating a browseable network of social relations. Although pre-dated by other services such as sixdegrees.com in 1997, social networking services emerged as a veritable phenomenon with the dramatic rise of friendster.com, which rapidly amassed millions of users beginning in the spring of 2003. Friendster's success was closely followed by a number of other popular services, including Tribe.net and orkut.com, quickly earning these services the acronym "YASNS" or "Yet Another Social Networking Service."

Though the users of these systems have constructed massive graph structures of social connectivity, typical web interfaces to these social networks remain relatively impoverished, showing only the network connections of single individuals in a linear list on a web page. Articulated connections between one's own "friends" in these systems are obscured, and can be unearthed only by paging through each friend's profile page. Higher level constructs, such as shared patterns of interests and community structures can be even harder to discern. This has problematic implications for members' ability to efficiently explore their online community and gauge both the scale and the individuals to which their self-reported personal information is exposed.

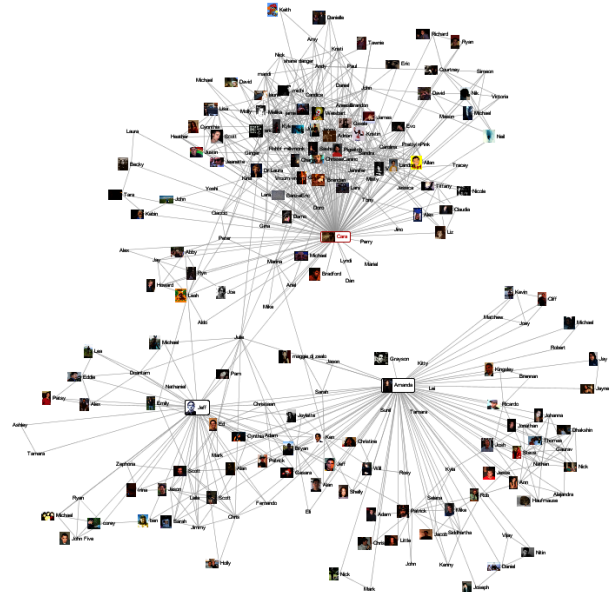In this paper we present the design of Vizster, a visualization

E-Mail: jheer@cs.berkeley.edu, dmb@sims.berkeley.edu

**Figure 1: Vizster depicting three intersecting social networks**

system for exploring such online social networks. Vizster builds on ethnographic research of online social networking services and previous work in social network visualization to provide a system by which members of such online communities can explore their articulated social network in the playful manner that they desire. Our system is motivated by both social and technological concerns. On the social side, we are interested in better facilitating the discovery of people, connections, and communities to promote increased awareness of community structure and information exposure, while maintaining a fun and socially engaged online space. From the perspective of information visualization, we are interested in crafting techniques for the situated exploration and visual analysis of large graph structures by a specific end-user population.

Our design builds upon familiar node-link network layouts to provide novel techniques for exploring inter-connectivity in large graph structures, supporting visual search and analysis of both network and profile data, customizing spring-embedded layouts, and automatically identifying and visualizing community structures. We have used the resulting system to visualize a 1.5 million member crawl of the popular Friendster service, collected in Winter 2003.

We first summarize related work in social network visualization and share relevant findings from an ethnographic investigation of the Friendster service. We then describe the design of the Vizster visualization in detail, followed by observations from a public installation and informal studies of the system's usage.

## 2    RELATED WORK

Visualization of social networks has a rich history, particularly within the social sciences, where node-link depictions of social relations have been employed as an analytical tool since at least

the 1930s. Linton Freeman documents the history of social network visualization within sociological research, providing examples of the ways in which spatial position, color, size, and shape can all be used to encode information [8]. For example, networks can be arranged on a map to represent the geographic distribution of a population. Alternatively, algorithmically generated layouts have useful spatial properties: a force-directed layout can be quite effective for spatially grouping connected communities, while a radial layout intuitively portrays network distances from a central actor. Color, size, and shape have been used to encode both topological and non-topological properties such as centrality, categorization, and gender.

In recent years, such approaches have been effectively used in the analysis of domains such as e-mail communication [7], early online social networks [1], and co-authorship networks in scientific publications [15]. There are a number of systems for generating such social network visualizations and performing statistical analyses for the purpose of sociological research, including UCINet [19], JUNG [12], and GUESS [2].

In addition to sociological research, there have been numerous social visualization projects intended for end-users rather than outside researchers. Though often utilizing techniques seen in analytical domains, these systems present users with visualizations of their own online social world; they are non-anonymous and often perceived with a sense of social place [10] different from more detached analytical environments. Nardi *et al.*'s ContactMap [14] uses spatial grouping and color to redundantly code for various community groups within a visualization of a user's e-mail contacts. TouchGraph [18] uses a force-directed layout to present a network visualization of users of the LiveJournal online community, allowing personal networks to be expanded or contracted by user interaction. BuddyZoo [5] analyzes users' instant messaging (IM) buddy lists to present a static network visualization of their IM contacts. boyd and Potter's Social Network Fragments [20] visualizes personal e-mail archives in a zoomable network view and includes temporal filtering to visualize contacts at various time periods. They found their system to be particularly well suited as a memory aid for past contexts and contacts.

Despite this wealth of social network visualization, we believe there is still a need for new designs and techniques, especially as articulated social networks become increasingly common in web services for signifying various types of relationships. Though limited network views have proven manageable and useful, means for better exploring extended network contexts and providing useful cues for exploration are possible. Visualization of profile attributes unique to online social networks is needed, and techniques for incorporating analytical tools within the simplified domain of end-user visualization may prove useful. In crafting Vizster, we aspired to create a visualization system that builds upon the successes of existing work, introducing an increased potential for exploration while maintaining a sense of playful social place. To do so, we turned to ethnographic investigations of the Friendster community to guide our design.

## 3 FRIENDSTER DESCRIPTION

Friendster was designed to be an online dating site, complete with profiles, demographic and interest driven search, and a private messaging system. What made Friendster unique was its articulated social networking component and testimonial feature. Users were asked to declare "friends" on the system whose pictures would also appear on the profile when the friends confirmed the relationship. Friends could write testimonials that would also appear on the profile. Both the friends and testimonials

were intended to signal additional information about the person's character for those interested in dating the person. Yet, when the early adopters began to use the service, they did not view it as a dating service, but a site where they could gather and communicate with their friends, surf for entertaining profiles and explore public displays of identity and relationships [6].

Early adopters fell primarily into one of three identity groups: gay men, techies/bloggers, and Burners (people who attended the Burning Man art festival in Nevada every September). Each group had a different presentation aesthetic and their profiles conveyed the aspects of their identity most associated with the communities through which they joined the service. For example, Burners usually used their "Playa names" (the nickname they used at the festival), uploaded pictures of themselves at the festival and talked about their music tastes. Many early adopters had densely connected social groups, which aided Friendster's spread. As Friendster became more popular, the user population diversified. That said, the vast majority of users throughout 2003 were in their 20s, college-educated, politically liberal, and living in urban centers. Teenagers did join the service, reporting ages of 69 or 61 and 71 (16 and 17 inverted) because the site forbade users under 18. By marking their age this way, teen users were able to easily find each other. While the service began in the San Francisco Bay Area, its popularity quickly spread to Europe and to Singapore, Malaysia and the Philippines where it took off dramatically. New users primarily came through invitations from active users; thus, the user base represents the extended networks of early adopters.

When users joined the site, they were required to develop a profile that asked for demographic information (age, location, hometown) and a description of tastes (dating preferences, music, books, movies). Users typically created the profile based on the norms of the friends who invited them. As such, the network structure was also apparent in the profile norms of different groups. People regularly updated their profiles, adding new photos, changing tastes dependent on their friends and writing testimonials for their friends in order to encourage reciprocity.

Through ethnographic analysis, boyd [4] uncovered common social practices endemic to Friendster amongst early adopters. While dating did occur, most joined the service because all of their friends joined. Users began using the service by searching for friends and old friends, adding them to their network where appropriate. Because Friendster limited the visibility of profiles to four degrees (friends' friends' friends' friends), users were often motivated to connect to as many people as possible to expand their visibility. Friendster's design decision to list the most highly connected people on the front page further activated the competitive tendency to collect as many people as possible.

In addition to personal profiles, early adopters began creating fake profiles – "Fakesters" – to represent anything from famous media figures like Angelina Jolie to cities like New York, concepts like Love to everyday objects like Salt. Fakesters like Brown University helped connect people who attended that school while Fakesters like Homer Simpson helped connect fans of the TV character. The Fakesters increased the density of the network because they both helped friends find each other and connected new people through shared interests. The profiles created for Fakesters were often quite creative and motivated many users to surf for interesting profiles in a treasure-hunt fashion.

While most users surfed the network by clicking on the friends of other profiles, there was also a search tool available, which helped in finding friends by name and aided in finding potential dates or hookups. Users could also click on interests and find anyone with that interest marked.

While the majority of people used Friendster for social exploration and play, questionable practices were also present. Some users used Friendster to distribute drugs while others used the service to construct fraudulent profiles for public roasting through testimonials. Another problem that emerged was the colliding of networks normally kept apart – not everyone was prepared to expose their networks to both friends and colleagues.

## 4    VISUALIZATION DESIGN

Our goal with Vizster was to build a visualization system that end-users of social networking services could use to facilitate discovery and increased awareness of their online community. We wanted to support the exploratory and playful aspects of Friendster while also giving users easier access to search and group patterns. While users regularly explored the network on Friendster, the linear format limited such explorations. This led us to develop richer network views and exploratory tools, while maintaining a local orientation. We also learned that the use of imagery was indispensable for identifying people and establishing a presentation of self, and so must play a central role in the visualization. In addition to helping support the current practices, we wanted to make sure that Vizster did not eliminate the data that helped users get a sense of people through their profiles. This includes the use of re-appropriated profile fields (*e.g.*, inverting ages to identify teenagers) for coded communication within a sub-population. For this reason, we realized that we must make searchable profile data very present and accessible in the visualization. These goals position Vizster differently from traditional social network visualizations used as analysis tools by social science researchers. The following description includes the implications this approach has had for our design decisions, both in terms of presentation and the level of technical sophistication exposed by the visualization.

Vizster presents social networks using a familiar node-link representation, where nodes represent members of the system and links represent the articulated "friendship" links between them (Figures 1,2). In this view, network members are presented using both their self-provided name and, if available, a representative photograph or image. The networks are presented as egocentric networks: networks consisting of an individual and their immediate friends. Users can expand the display by selecting nodes to make visible others' immediate friends as well. To the right of the network display is a panel presenting a person's profile. As discussed later, the profile panel also provides direct manipulation searches over profile text.

In pursuing this design, we chose to violate Shneiderman's mantra of "overview first, zoom and filter, then details-on-demand" [17], instead opting for a philosophy of "start with what you know, then grow." An overview of the full network is inappropriate in this personal context, as the sheer scale obscures useful landmarks. Users of this system are familiar with their friends, some friends of friends, and various "celebrities." Given a lack of *a priori* knowledge of the user's familiarity with their extended network, starting from an egocentric perspective not only carries less perceptual and computational burden, but guarantees the presence of readily identifiable landmarks for
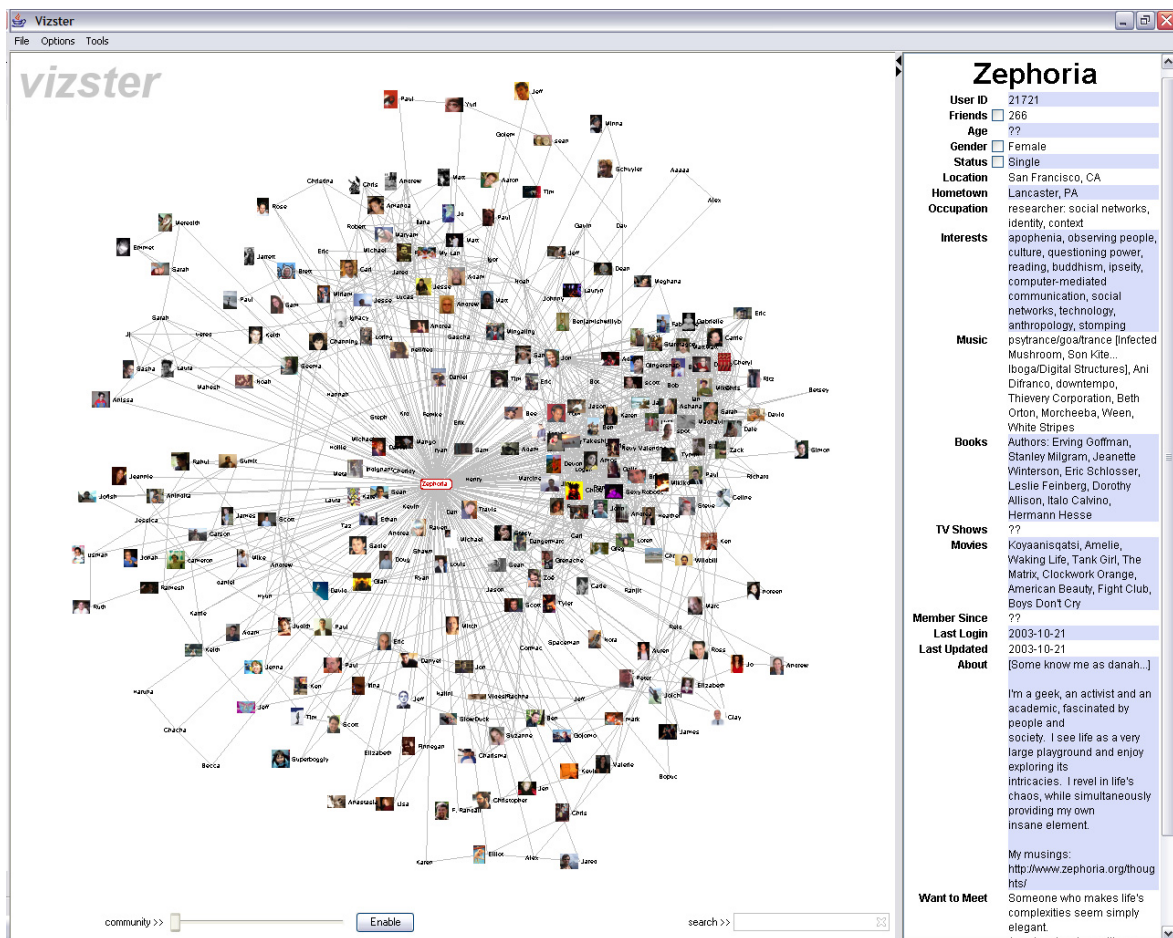


Figure 2: Screen shot of the Vizster visualization system. The left side presents a network display with controls for community analysis and keyword search. The right side consists of a panel displaying a selected member's profile information. Words in the profile panel that occur in more than one profile will highlight on mouse-over; clicking these words will initiate searches for those terms. The checkboxes in the profile panel will initiate an "X-ray" view of that particular profile dimension (see Figures 7-9).

orienting the user: one's self and their immediate friends. From this base, users can selectively expand the network to explore their greater community.

## 4.1    Layout

Network layout is computed using a spring-embedding (force-directed) algorithm, in which nodes repel each other and edges act as springs. This particular layout has the advantage of grouping users into identifiable communities based upon increased connectivity. The average connectivity of these social networks, with users commonly having fifty to a hundred articulated friendships, renders radial views illegible for anything beyond 1st order networks. Though adjacency matrix-based views have shown promise for visualizing larger networks (at least among very specialized, technical users [9]), their unfamiliarity to our prototypical users and their single degree of freedom for spatially organizing members (thus limiting community groupings) make them unsuited for this particular context.

The layout is computed in real-time, providing an animated layout akin to that of the Visual Thesaurus [21] and TouchGraph [18]. The Barnes-Hut algorithm [3] is used to efficiently compute n-body (anti-gravity) calculations and numerical integration routines are used to smoothly update on-screen locations. This approach avoids interactive delays induced by layout computation and allows the user to participate in the layout process by interactively dragging nodes to tease apart communities. Although items eventually settle into permanent locations, we allow the force simulation to continue running, causing items to maintain some subtle movement. In our observations of usage, this movement has not sparked undue distraction or interference with user actions, but gives the visualization a living or "breathing" feel, connoting social energy and playfulness. This approach also helps the user understand that the locations are relative to the present data instead of exact relationship strength.

To optimize the layout, we parameterize the tension of the individual spring-edges by node connectivity. Nodes with lower connectivity are given higher tension, causing singly-connected nodes to remain close to expanded nodes and connected communities to assume higher "orbits." This additionally causes nodes with lower connectivity to more closely "tag-along" with their friends, reinforcing their limited connections. Meanwhile, nodes with higher connectivity still group into communities, but with a reduced density to increase readability.

## 4.2    Basic Interaction

Basic interaction is done with simple mouse operations. Clicking a node causes the corresponding member profile to appear in the profile panel to the right of the network view. Dragging a node moves it around in the space; upon release it is again subject to the ongoing force-directed layout. Double-clicking a node causes it to either expand or contract, depending on its current state. Upon expansion, the egocentric network for that member is added to the visualization. The expanded node assumes a fixed position (i.e., it is freed from the layout), subject to re-positioning by the user. Double-clicking an expanded node causes that member's egocentric network to disappear and the node again becomes subject to the layout. This mechanism allows users to selectively grow or hide aspects of the network, increasing the scope of the visualization while maintaining the context of already expanded members. The force-directed layout adjusts to these changes, with friends shared between expanded members moving to an intermediate position.

## 4.3    Exploration: Connectivity Highlighting

To aid network understanding, Vizster highlights nodes based on connectivity in the larger network context. When the mouse hovers over a node, it causes that person, that person's friends, and visible friends-of-friends to highlight. A graded color scale moving from red-orange to orange to yellow—redundantly coding through both intensity and hue—is used to visualize network distance. All other members in the graph are correspondingly desaturated, with images reverting to grayscale, to provide a figure-ground separation between highlighted and non-highlighted nodes (Figure 3). Moving the mouse off the node causes the highlighting to disappear, but with a half-second hysteresis applied to prevent annoying visual "bouncing" as consecutive nodes are visited by the mouse pointer. Clicking and holding a node for 1 second will make the current highlighting persistent (indicated by a brief flashing of the highlighted nodes), after which clicking the background or another node will resume the normal highlighting interaction.

Though readily understandable, egocentric views suffer from a lack of greater network context. To provide cues to the greater network topology, connectivity is computed using the complete backing graph rather than the visualized graph. As a result, people connected to a selected person through intermediate friends in the larger network highlight even if those intermediate nodes are not currently visible (Figure 4). These connectivity queries, however, do not pass through expanded nodes (e.g., "Amanda" in Figure 4), as they are obviously intermediate to currently visible nodes, and the goal is to help provide cues to currently obscured connections.

## 4.4    Exploration: Linkage Views

As expanding a member's egocentric network can significantly change the display, in response to user feedback we have also incorporated linkage views that enable the user to visualize intermediate nodes in a lightweight manner. A user can click a node to select it, mouse over another node of interest, and then tap the space bar to visualize all connections between the selected node and the current mouse target. This causes any intermediate nodes not in the currently expanded networks to also be displayed, providing a complete view of the two-hop connectivity between the nodes (Figure 5).

## 4.5    Navigation

The network visualization employs a mix of both manual and automated panning and zooming for navigating the space. Panning is performed by dragging the background of the display with the left mouse button down. The display automatically pans when a new node is expanded, centering on the newly expanded network. Manual zooming is performed by holding down the right mouse button and moving the mouse up or down. Simply clicking the right mouse button causes the display to automatically pan and zoom such that the entire visualized network fits within the display. The rendering components update to draw higher resolution photos when zoomed-in to double the normal scale.
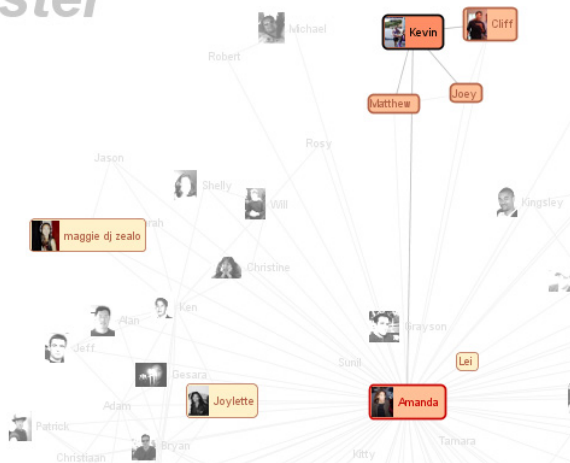
**Figure 3: Connectivity Highlighting. The highlighted person "Kevin" is connected to "maggie dj zealo", "Joylette", and "Lei" (all shown in yellow) by friends not currently visible.**
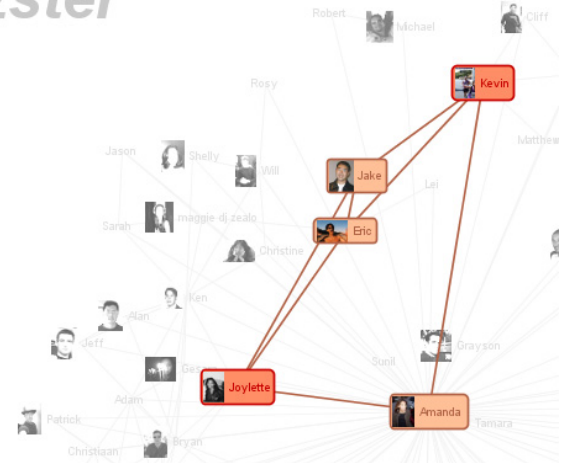


**Figure 4: Linkage Views. Clicking on "Kevin" and then tapping the space bar while hovering over "Joylette" (or vice versa) causes all intermediate nodes to be visualized.**

In some cases it is desirable to see a particular region while still maintaining the context of a zoomed-out network. To support this, we use geometric distortion within the context of the force-directed layout to provide a focus+context display (Figure 5). When the display is zoomed-out such that node text is no longer legible, users can hold down the left mouse button on a node for a second to "inflate" the currently highlighted subset of nodes. The nodes balloon to become more readily visible and their mass in the force-directed layout is correspondingly increased to cause the nodes to push further apart and prevent occlusion.

### 4.6 Search

Vizster also supports keyword search of the visualized network. As items become visible, their profile attributes are added to an in-memory search index. A search box allows users to type in search queries. Search hits are indicated by "auras" around matching nodes (Figure 6), with non-matching nodes again de-emphasized through desaturation. The search auras are presented in a dark purplish color, intended to leverage blue-yellow color opposition to balance against the hues of the highlighting. This contrast was inspired by the graceful balance of Mark Rothko's painting *No. 14*.

As users mouse over the network with search auras present, highlighting is performed as usual, desaturating the text and images of non-highlighted search hits. The search auras themselves remain visible, however, to balance competing search and connectivity queries. Search auras for highlighted nodes also grow slightly larger to maintain visibility.

Additionally, users can issue search queries from the profile panel in a direct manipulation fashion. As the mouse pointer moves over specific words, the word will highlight if there are at least 2 matching search hits for that term. Clicking the word will then execute a search, with the text of the query appearing in the editable search box. Double clicking a word will initiate a search for the surrounding phrase (profile entries often have multiple items, typically comma-delimited). Manually selecting text by mouse drag similarly executes a search for the selected text.

### 4.7 Attribute Visualization using "X-Ray" mode

In addition to generating search queries, the profile panel also includes controls to further visualize the individual profile attributes of members. By clicking on a check box next to an
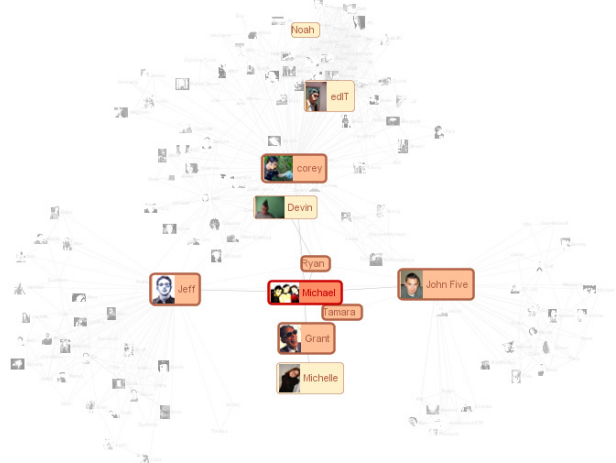


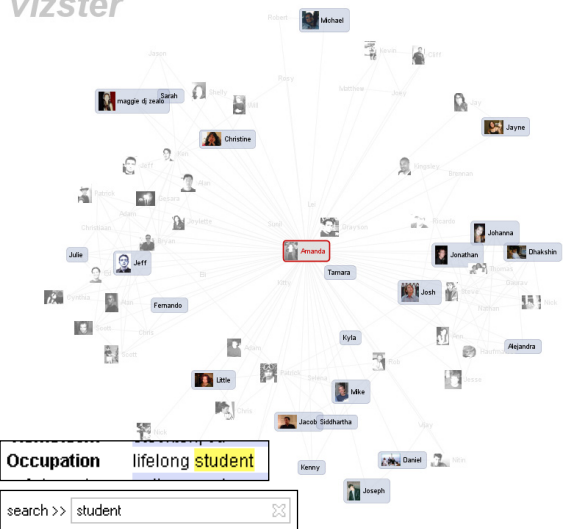**Figure 5: Focus+Context views by inflating highlighted nodes.**



**Figure 6: Visualized search results for the query "student". Searches are executed by clicking text in the profile panel or typing in the search box.**

attribute within the profile panel, Vizster will enter an "X-ray" mode visualizing attribute values such as age, number of friends, gender, relationship status, or time since last login. The background of the display turns black to clearly indicate the modal state, member photos are removed, and node color is used to visualize the currently selected attribute. Additionally, a color legend appears in the upper-right corner of the display. In this state, all the other operations discussed still function as normal. Highlighting and search still desaturate non-matching items, though node colors no longer change under highlighting.

Figure 7 shows Vizster in X-ray mode visualizing the number of friends each person has in the service. Figure 8 shows the use of X-ray mode to visualize each member's gender. Figure 9 displays the results of including a mouse-over highlight, and search results for the query "student." Community groupings (discussed below) are shown, but consistent with the X-ray metaphor are reduced to simple outlines.

## 4.8 Visualizing Community Structure

Although the force-directed layout does an admirable job of clustering the network into communities based on linkage, Vizster also supports explicit visualization of community structures. Inferred community groupings are visibly represented as "blobs" surrounding their community members, taking advantage of low spatial frequencies to make community structures apparent (Figure 10). These blobs are computed by taking the coordinates of constituent members, calculating the convex hull, and then using quadratic splines to provide curved borders. As members within a community are moved about, the blob boundaries adjust accordingly for a dynamic, playful character. The blobs are translucently rendered, with colors determined by hue sampling. This semi-transparency allows the blobs to act as Venn diagrams if members are assigned to multiple communities. To further improve the layout, extra-community edges are given weaker spring forces to support the spatial separation of communities.

As manual creation of large communities can prove somewhat tedious, Vizster includes tools for the automatic determination of community structure. For this task, we have used Newman's community identification algorithm [16], which identifies group structures based solely on link analysis. The algorithm employs hierarchical agglomerative clustering, first placing each node in its own community, then greedily merging groups based upon a metric that attempts to maximize within cluster linkage while minimizing between cluster linkage.

Users can instigate automated community analysis by clicking the community analysis button at the bottom-left of the display. The community structure algorithm is then run on the adjacency matrix of the currently visible network. Currently expanded nodes are omitted from the input matrix to avoid unduly collapsing communities—the goal is to show community relations *surrounding* the expanded nodes. The results of the community analysis are then visualized using communities identified at the optimal value of the clustering metric.

Our design hides the complexity of the algorithm from users, letting them explore these data mining results through simple widgets and visual analysis. Like most data mining methods, the community analysis is not perfect, as it may identify communities at higher or lower granularities than those desired by the user or make assignments otherwise perceived as problematic. To help combat this, a community slider is provided to explore the various states of the clustering (Figures 10 and 11). Moving the slider to the far left reverts the display to the initial state of the clustering, while moving the slider progressively to the right reveals each merge performed by the algorithm. Thus the slider allows the user
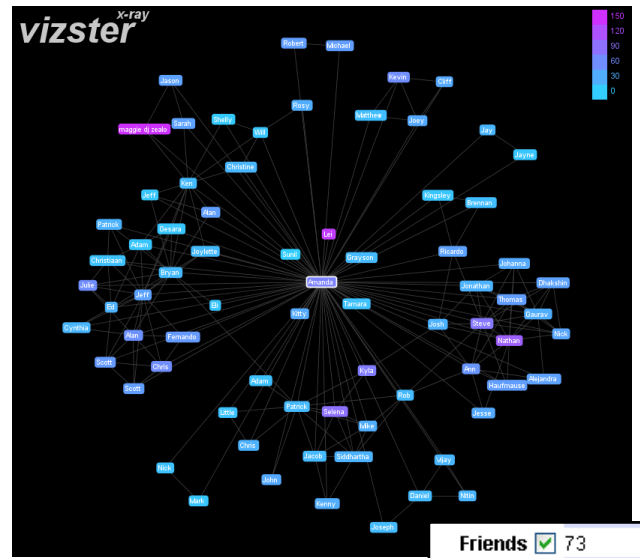


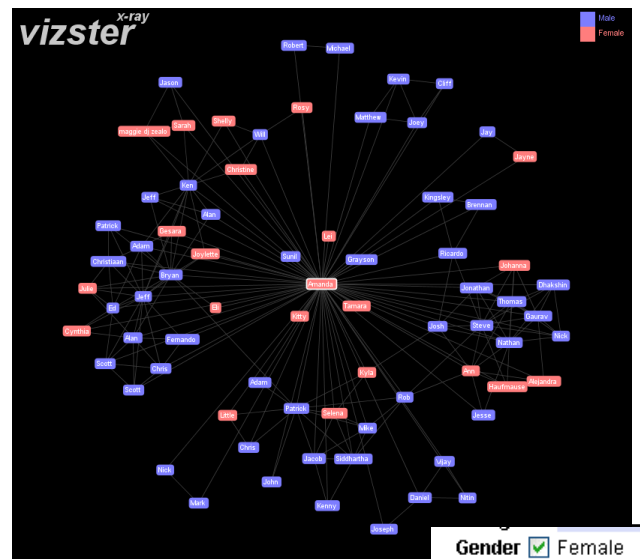Figure 7: X-ray mode visualizing the number of friends.
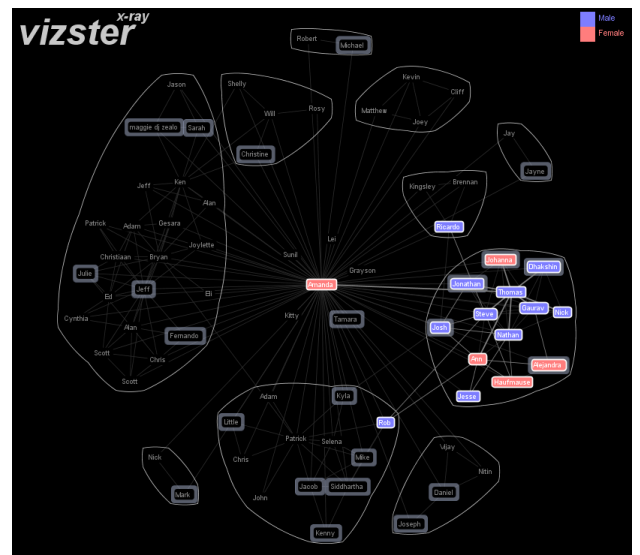


Figure 8: X-ray mode visualizing gender.



Figure 9: X-ray Mode visualizing genders, search hits, mouse-over highlight and community structures.

to interactively explore clustering states by moving through progressive slices of the computed cluster tree. In practice, we have observed users employ the slider until the communities "look right" to them. The current community visualization persists when additional nodes are expanded or contracted; when the visible network changes, an update button is added to the control panel, allowing users to re-run the community analysis.

The current algorithm assigns members to unique communities. In future work, it may be useful to generalize the underlying algorithm to support soft or "fuzzy" clustering techniques permitting multiple membership in communities. We also plan to introduce manual editing so that users may further adjust communities as desired.

### 4.9    Summary

In summary, the Vizster design constitutes a visual environment for the exploration and analysis of online social networks, including both topological and profile data. The scale of displayed information and layout were chosen to support observed behavior and capabilities, and allow users to expand visualized networks while maintaining landmarks. Interactive highlighting is used to explore friendship relations and unearth "hidden" connections in the larger network structure. Panning, zooming, and distortion techniques are provided to help users navigate visualized networks. Interactive search and attribute visualization ("X-ray" mode) enable visual exploration of member profile data. Finally, visual community analysis is provided to help users construct and explore higher-level structures of their online communities.

### 5    IMPLEMENTATION NOTES

Vizster was written in Java using the prefuse visualization toolkit [11]. The toolkit's filters, configurable physics simulation, image loading and caching, renderers, and interactive controls greatly facilitated the implementation of our design. We also wrote extensions for database connectivity and to perform connectivity highlighting and linkage views. To support generalized keyword search, we also integrated the Lucene [13] search engine into the prefuse framework. Network and profile data for the visualization were collected using a custom web crawler and stored in a backing MySQL database. To support highlighting connectivity queries and faster expansion response times, $2^{nd}$ order networks are loaded from the database upon expansion of nodes, though only the $1^{st}$ order networks are immediately visualized. The only caveat is the increased memory requirements of the application. Source code (but not collected data) for the application is available from http://prefuse.sourceforge.net.

### 6    USAGE OBSERVATION

To evaluate and further guide the design of our visualization, we observed usage in two environments: a public installation at a large party and an informal laboratory setting. While we were certainly interested in gauging Vizster's utility and usability, we were also interested in larger patterns of discovery—finding unknown people, connections, communities of relevance—and in people's social and affective reactions to the visualization.

Our first observation of usage was conducted around an interactive installation at a 500-person all-night event in San Francisco. Many of the party-goers included early adopters of the Friendster system, especially those affiliated with the Burning Man festival and tech culture. The installation consisted of an interactive kiosk and a projection of the visualization onto a large screen. Throughout the night, we observed usage of the system by over a hundred users, noting the reactions of users and onlookers.
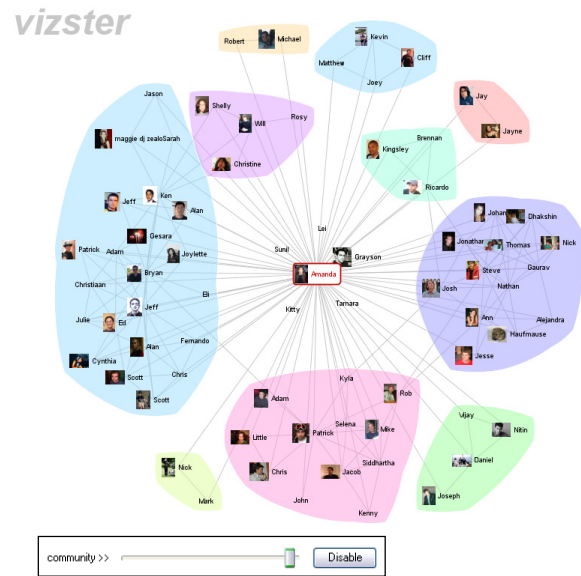


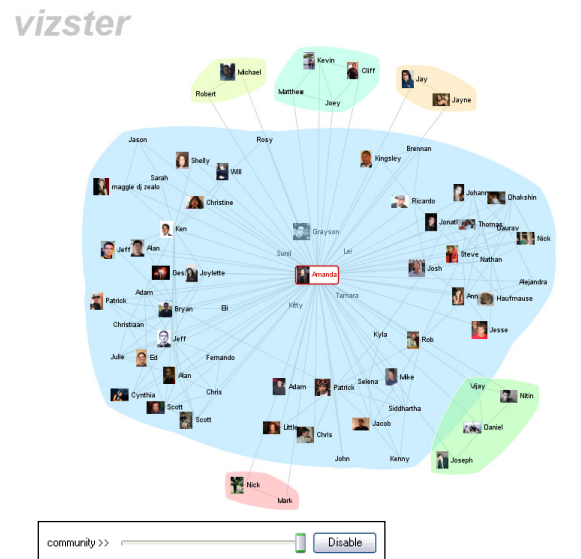Figure 10: Community structure visualization using algorithmically determined optimum.



Figure 11: Community structure visualization after the community slider has been dragged to the right.

We also observed usage in an informal laboratory setting. Participants consisted of 5 males and 1 female, all Friendster members in their early-to-late twenties. To maintain ecological validity, we did not provide users with any tutorials nor did we give them pre-specified tasks. Instead, we simply asked them to play with the system, explore it as they saw fit, and talk-aloud about their experiences as they did so. After 15 minutes of exploration, we then provided users with a one page reference to the various controls, to ensure that we could gauge user reaction to features they might not have explored on their own. After 10 more minutes of observation, we then interviewed participants about their experience and opinions.

Usage, especially within the party setting, was routinely coupled with some form of social play—for example, games to see if one could hunt down specific people they knew. Onlookers tended to participate as well, engendering a shared experience around the visualization. Most users and observers began retelling

stories about people, their relationships, and shared experiences. Users would also play with the various interactive features, often without pursuing a specific task. In particular, users enjoyed zooming navigation and experimenting with the real-time layout, whether dragging items around to tease clusters apart or dragging items to build up spring tensions and then gleefully watch the recoil of the network upon release.

Through experimentation, most party-goers and study participants quickly made sense of the connectivity highlighting. Often times, a lone node would highlight far away from the current locus of interaction, eliciting comments such as "Why did that light up?" This would then trigger exploration, with users further expanding the network to discover the intermediary friends that were not yet visible. Users (or others also watching the visualization) would then answer their own question: "Oh, it's telling you that they are connected in some way."

Similarly, users in both observation contexts enjoyed experimenting with the community analysis. Users would play with the slider to get a feel for how the feature worked, and then settle with a configuration that "looked right" to them. Nearly everyone then began retelling stories about the various communities and their members. Prototypical comments included "Look, it found all the people from Harvard" and "It was smart enough to know not to group them together, they hate each other." In general, people tended to impart a higher level of sophistication to the community analysis than is implemented; some remain convinced that it compares members' profile data. Though they often parameterized the analysis using the slider, users seemed to discount their own input into this semi-automatic process.

Users easily discovered the X-ray mode, often commenting on the results of their ad-hoc analyses (*e.g.*, "What!? She's not single!"). It proved particularly powerful when coupled with searches, as users naturally specified conjunctive queries by running search queries on top of the X-ray state. As we expected, however, users in both contexts showed a strong preference for imagery for navigating the networks, for example saying "bring the pictures back" when they wanted to hunt for someone else.

The lab context also provided valuable feedback on various features. In particular, linkage views were introduced in response to early user requests for a means of quickly unearthing connections without expanding another network. However, in later studies we found that both linkage views and the ability to inflate nodes were not discovered through experimentation, though were appreciated after consulting the reference. One participant suggested adding a mechanism to opportunistically suggest features to users—an approach with a successful track record in video games.

In all, we found that the visualization was used by users to both explore and play with their networks—expanding the network to quite large depths, performing visual analyses, and exploring community structures while simultaneously engaging in social narratives. Although more formal comparative and longitudinal studies would be required to more fully assess the utility of the individual features introduced, the engaged yet playful exploration and discovery we observed helps validate a number of our design decisions. As one study participant put it, "Friendster gives you your two hours of fun, and this doubles it."

## 7    CONCLUSION

In this paper we have described the design of Vizster, a system for end-user visual exploration of online social networks. Grounded in an ethnographic investigation of the Friendster social networking service, the resulting Vizster system is an experiment in applying the fruits of information visualization within an end-

user context more characteristic of play than analysis. As such, Vizster contributes a number of network analysis techniques designed to both reveal information and maintain a playful social environment. These features include connectivity highlighting and linkage views for users to explore greater network contexts, X-ray mode and profile search for the visual analysis of member profile data, and community analysis for investigating higher-level community structures. Observations of usage found that users find these features both useful and enjoyable for exploring their personal online networks. In future work, we are interested in exploring the extension of these techniques to more traditional analytical domains, where more rigorous evaluation should be applied to evaluate their benefits and trade-offs. As forms of online sociality continue to spread and diversify, we also anticipate the need for further design approaches balancing utility and experience, enabling information visualization technologies to take on an increasingly important role in both mediating and making sense of our shared social landscape.

## REFERERNCES

[1]   Adamic, A., O. Buyukkokten, E. Adar. A Social Network Caught in the Web. *First Monday*, 8(6), 2003.

[2]   Adar, E. GUESS: The Graph Exploration System. http://www.hpl.hp.com/research/idl/projects/graphs/index.html

[3]   Barnes, J., P. Hut. A Hierarchical O(N log N) Force Calculation Algorithm. *Nature*, 324, 4 December 1986.

[4]   boyd, d. Friendster and Publicly Articulated Social Networks. *CHI 2004*, Vienna, Austria, 1279-1282.

[5]   BuddyZoo. http://buddyzoo.com

[6]   Donath, J., d. boyd. Public displays of connection. *BT Technology Journal*, 22(4), October 2004, 71-82.

[7]   Fisher, D., P. Dourish. Social and temporal structures in everyday collaboration. *CHI 2004*, Vienna, Austria, 551-558.

[8]   Freeman, L. Visualizing Social Networks. *Journal of Social Structure*, 1, 2000.

[9]   Ghoniem, M., J.-D. Fekete, P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *InfoVis 2004*, Austin, TX, 17-24.

[10] Harrison, S., P. Dourish. Re-place-ing Space: The Roles of Place and Space in Collaborative Systems. *CSCW 1996*, Boston, MA, 67-76.

[11] Heer, J., S.K. Card, J.A. Landay. prefuse: A Toolkit for Interactive Information Visualization. *CHI 2005*, Portland, OR, 421-430.

[12] JUNG: Java Universal Network/Graph Framework. http://jung.sf.net/

[13] Lucene Search Engine. http://lucene.apache.org

[14] Nardi, B., S. Whittaker, E. Isaacs, M. Creech, J. Johnson, J. Hainsworth. ContactMap: Integrating Communication and Information Through Visualizing Personal Social Networks. *Communications of the ACM*, April 2002.

[15] Newman, M.E.J. Co-authorship networks and patterns of scientific collaboration. *Proc. Natl. Acad. Sciences*, 101, 2004, 5200-5205.

[16] Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 2004, 066133.

[17] Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *1996 IEEE Conference on Visual Languages*, Boulder, CO, 336-343.

[18] TouchGraph Live Journal Browser. TouchGraph, LLC. http://www.touchgraph.com/TG_LJ_Browser.html

[19] UCINET: Social Network Analysis Software. http://www.analytictech.com/ucinet.htm

[20] Viegas, F., d. boyd, D. Nguyen, J. Potter, and J. Donath. Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. *HICSS-37*, Big Island, HI, 2002.

[21] Visual Thesaurus. http://www.visualthesaurus.com