TIE: Temporal Interaction Explorer for Co-presence Communities

Daniel J. Boston, and Cristian Borcea
Computer Science Department
New Jersey Institute of Technology
Newark, NJ, USA
E-mail: djb38@njit.edu, and borcea@cs.njit.edu

***Abstract*—The widespread adoption of smart phones allows for the seamless capture of social interactions on a scale that was once impossible. Co-presence, collected using Bluetooth on the phones, faithfully represents such real-world social interactions. This social information can be transformed into communities, which can be leveraged into applications such as recommender systems and collaborative tools. However, correctly identifying communities is difficult.**

This paper presents TIE, a visualization tool that enables effective review of detected communities. With TIE, we can visualize the social interaction of a set of people over time. Also, TIE can overlay detected community events in a usable way over the underlying social interactions. Further, it allows us to investigate specific social interaction events and see how well detected communities match those events. Lastly, it enables the comparison of different sets of detected communities by interactively switching between overlays. TIE has proven useful in evaluating our community detection algorithms and has been invaluable in identifying strengths and weaknesses of these algorithms. Beyond our needs, TIE is usable for other data sets that can be reduced to temporal interaction events such as multiplayer game communities, SMS interactions, and paper co-authorship.

***Keywords*-visualization of co-presence communities; smart phones; time-series events**

I. Introduction

The widespread adoption of smart phones allows for the seamless capture of physical world social interactions in the form of co-presence events. These phones come from different vendors and have a wide variety of interfaces and capabilities, but most support some form of short-range radio such as Bluetooth, and Internet-capable radios like 3G or WiFi. The ubiquity of these interfaces enables large-scale collection of co-presence events, and thus potential social interactions, with an immediacy and correctness that was once impossible.

Using a short-range radio such as Bluetooth offers many advantages, as its maximum effective range of ten meters ensures any interactions captured indicate co-presence. Its small power footprint is especially attractive for situations where social information capture is desired, but device size and weight must be restricted. As well, it helps overcome some privacy implications of GPS or other location-specific interfaces. Bluetooth does not indicate absolute position, it just captures co-presence events in the immediate vicinity. This is an attractive consideration for those concerned with revealing their actual location. Bluetooth is an established method for capturing co-presence data, as exemplified by the Reality Mining [1] project, Mtibaa et al. [2] and more recently by Kostakos et al. [3] and the Mobius [4], [5] project.

These co-presence events can be used to discover communities of mutually co-present individuals. Such results are interesting from a sociological perspective, but also from a social computing point of view. This type of physical community (as opposed to on-line community) can be used in many socially-aware applications such as recommender systems for social events, tools for collaborative work, or opportunistic data forwarding in ad hoc networks.

A variety of techniques exist to identify these communities or groups, ranging from investigating the data by hand to using complex algorithms. Manually identifying communities is prohibitively time consuming; yet algorithms often produce results that are difficult to verify or that are inadequate. This is due to the inherent difficulties of finding groups from co-presence events, especially with the larger problem of communities that meet over time.

Various techniques exist for verifying collected social information. Experience sampling methods [6] allow investigators to query participants about their neighbors during the day. This can be useful, but surveys are often ignored or may disrupt the community. End-of-day journaling provides a way to avoid frequent disruptions, but can suffer from forgotten events or interactions. Surveys taken after the completion of the entire study are also commonly used, but suffer amplified effects similar to the prior method. One other complication is that the mobile phones can discover strangers often socialized with—known as familiar strangers [7]—which a survey cannot confirm as their identity is unknown to the participant.

The use of such validation techniques is demonstrated by Mardenfeld, et al [8], where an after-study survey revealed the GDC algorithm finds measurably better groups than another popular community detection algorithm, K-Clique. This survey does not help evaluate alternative group detection algorithms, illustrating the need for some other way to evaluate these results.

In surveying existing methods for social information review, a number of shortcomings related to our problem domain were identified [1]. Some tools clearly show communities, but hide the underlying co-presence events used to detect those communities. As well, the nature of the community over time is hidden, as most of the social information tools lack temporal representation. While the tools display the end result of the group detection algorithms and may allow direct comparison between methods, comparison against the social information used in the detection process is impossible. An investigator cannot verify that a detected community adequately expresses the membership of a group over time.

This paper presents Temporal Interaction Explorer (TIE), a tool designed to investigate co-presence and related community events. We began our design from a simple observation about the nature of the social information we collected. As it was derived mechanically, it is devoid of the memory bias that might be introduced from human journaling or other methods of record. Nearby communities will be present in the record, as will familiar strangers [7]. Based on this observation, the raw co-presence record represents sufficient ground truth for detected community verification, if a tool was developed to efficiently explore it.

TIE leverages the social information from which communities are discovered to enable easy verification of detected groups. At the core of TIE is the ability to explore the raw co-presence events independent of community information. Communities are easily explored through a configurable overlay system. Multiple overlays can be pre-configured, allowing comparison of different detected community sets against the same source data. TIE also enables deeper exploration of specific co-presence events as recorded by particular people; these reported records and the aggregate opinion of all contributing records on member participation in an event are featured, with the community overlay appearing in a consistent fashion across all visualizations. These interactive and powerful capabilities make TIE a valuable tool in evaluating the results of community detection algorithms.

In our own research, TIE has proven useful in evaluating our community detection algorithms. We have used it to find algorithm output errors that had otherwise evaded discovery, and to identify strengths and weaknesses of detected communities using the event exploration and overlay features of TIE. It has also enabled us to compare the results of slightly different detection methods.

Besides the communities of people that are the focus of our research, TIE could be useful in the more general case of data reducible to interactions over time between entities. It can be applied to SMS or email interactions to explore ‘communities’ of correspondents. Website visits could also be a form of interaction, as websites become more focused on enabling user interaction. Online games are a fast emerging medium that often involve significant social interaction; there are a wealth of communities to discover and validate within this venue. Alternatively, TIE could be helpful in ornithological surveys of bird migration and flock habits, or other animal community studies such as ZebraNet [9] using data collected from tiny wireless sensors.

The rest of the paper is organized as follows. In Section II, we provide background information for our research, including data collection and community detection. We also discuss similarities and differences of our tool from existing social and temporal information exploration tools, as well as related research on collecting and utilizing social information. Section III presents the format of data TIE expects. In Section IV, TIE’s design and implementation are presented and discussed extensively. Section V describes use-case experiences we have had using TIE, including specific ways it has helped during the development of new community detection methodologies. Finally, Section VI presents conclusions and a few remarks on the future of TIE.

II. Background and Related Work

Before describing TIE in more detail, we will discuss the nature of our work, including its social and temporal aspects. This will serve as a background to the design and implementation of TIE and help facilitate understanding of our research area.

A. Background

The incorporation of social information into online and mobile applications or services has become mainstream in the past several years. Capturing information about social groups formed through face-to-face interactions can enable new and improved types of applications, like recommender systems, collaborative tools, or socially aware content sharing.

One way to identify this type of social group is to leverage information collected automatically from smart phones carried by mobile users, such as location or co-presence. Correspondingly, a group can be defined as a collection of users who spend a significant amount of time together and meet for a significant number of times. Discovering such groups, however, is difficult: (i) group members do not necessarily attend all group meetings, (ii) guests or people who pass by the meeting location can appear to be part of groups, (iii) group members spend different amounts of time at meetings, (iv) the collected data is incomplete due to sampling frequency and mobility, and (v) users may collect different data for the same group meeting.

Our GPI algorithm [10] used location traces to identify, with high accuracy and low false positives, groups and their associated places. However, GPI requires a localization

[1] These tools and their shortcomings are addressed in section II-B, Related Work

system on every mobile device which may not be available (especially indoors), may have questionable accuracy, requires significant battery power, and exposes significant privacy concerns.

With this in mind we designed a new algorithm, Group Discovery using Co-location Traces (GDC) [8]. It uses co-presence traces to identify groups. A co-presence trace for a user is a set of records of other users that are within a certain proximity at the same time. GDC leverages the Bluetooth discovery protocol to collect these traces. We also define a user's co-presence trace as their perspective, as two users will ultimately produce a slightly different, although similar, perspective on their overall co-presence, even when nearby.

To validate GDC, we collected one month of Bluetooth co-presence data for a set of 141 students at NJIT. The study took place on our medium size urban campus, and the subjects were representative of the various majors offered on campus; 75% were undergraduates and 25% were graduates. Also, 28% were women and 72% were men.

Mobile phones were distributed to students, and a program quietly recorded the Bluetooth addresses of nearby devices using the Bluetooth discovery protocol. Discovery queries occurred on each device at a random interval between 1 and 3 minutes. The randomness was introduced to minimize the potential delays due to wireless collisions (which could lead to losing records). Each discovery query took approximately 20-30 seconds to finish, and the local records were uploaded to a server periodically. Note that this randomness, as well as other environmental concerns (such as radio interference), leads to the differences in perspective on co-presence mentioned previously.

With these co-presence records, we used GDC to discover a set of groups. These groups entail specific meeting events, and are generated from the co-presence perspectives contributed by the set of users. We presented the detected communities to participants for validation, and our results received positive feedback.

GDC ran on data collected at a central location. We modified the design of GDC to run distributed over mobile phones, which store co-location traces and calculate groups locally. Phones exchange data directly with other phones, and a localized GDC runs on available collected records. This alternative formulation is called Distributed GDC (DGDC).

DGDC benefits users with increased privacy; although users share records, in general this is not private information as the users physically saw each other. However, local record exchange results in each user developing her own perspective on communities. This can lead to significant difficulties in evaluating DGDC results, as not all communities are fully represented by all perspectives. Also, too much time has passed to re-survey the original participants about the new results. These challenges helped motivate the creation of TIE, and guided our design and implementation.

B. *Related Work*

Besides our own work in community detection, there are other well-known graph algorithms, such as K-Clique [11] and WNA [12], that can be employed to detect communities based on co-presence. They work by putting an edge in the graph between any two users who have spent a certain amount of time together. Unfortunately, since they use only pair-wise information, there is no guarantee that their detected communities actually spent any time together. Furthermore, the actual meeting times (events) of the communities are lost.

The data we collected to evaluate GDC and DGDC does not represent the only co-presence data available. The Reality Mining project [1] collected some nine months of co-presence data on nearly 100 subjects using Bluetooth. They augmented their analysis of resulting communities by incorporating other social information such as location and phone call logs. A different study by Mtibaa, et al. [2] collected Bluetooth co-presence of 28 subjects over the course of a conference. They compared the resulting social information against a declared list of social contacts, in contrast to our interest in community discovery.

Another approach is introduced with Cityware [3], where several thousand users voluntarily contribute Bluetooth meeting information by running an application on their mobile phones. Although for the purposes of the study in [3] they were only interested in the presence or absence of an encounter, their collection methodology (Bluetooth) warrants interest.

Beyond human interactions, the ZebraNet [9] project involved using radio and GPS sensors in a sparse sensor network to accurately track the movement, interactions, and migrations of zebra in Kenya. There is interest in the biological research field concerning the social behaviors of animals, and as such presents an interesting opportunity for community detection algorithms and validation techniques.

A significant body of research has occurred concerning tools and techniques for visualizing social information, generally as a graph or network. Vizster [13] is one such tool, designed to allow interactive visualization of social networks both large and small. For social information lacking a temporal component, it is an effective tool; beyond this, it falters.

Graph interaction libraries such as igraph [14] and JUNG [15] also provide powerful tools for visualizing and analyzing social information when in the context of a network or graph, but again lack temporal expressiveness without significant manipulation.

General frameworks for visualization such as Protovis [16] and Prefuse [17] allow the creation of excellent interactive data exploration environments. However, in our case creating visualizations from programmatic primitives proved simpler than using the interfaces of these frameworks and finding ways to fit our data to their more general models.

We also believe that it is better to have a tool tailored for reviewing co-presence communities given the rising interest in social networking and social computing.

More closely related to TIE are tools for temporal data visualization. A number of taxonomies exist [18], [19] regarding the breadth of techniques and approaches for temporal representation of information. While these taxonomies give good general guidance in terms of best practices and current practices, they do not address our specific domain, that of visualizing co-presence social interaction information over time.

An important distinction should be made here. Some visualization tools exist that enable exploration of spatio-temporal data, such as TimeMap [20], city-wide crime investigation [21], and geovisualization mashups [22]. Yet these tools and the data they visualize are fixed in space. Our data lacks spatial characteristics; it involves relationships between people and people, not between people and spatial locations.

ActiviTree [23] provides an interesting approach to visualizing temporal information as a sequence of events, mining such sequences from underlying 'activity journal' data common to the social sciences and enabling exploration of significant events recorded in these journals. However, bridging the gap between ActiviTree's approach to its data model and the far less clearly event-associated data model of opportunistically recorded co-presence is difficult.

III. TIE Data

In discussing TIE, it would be remiss not to describe in more detail the data it is designed to visualize. There are two components to the data; the first is a set of co-presence events called the source data, and the second is a set of communities or groups (of users) and their events called the overlay data.

A. Source Data

Co-presence is simply defined as two entities being very near each other at a particular moment in time. A co-presence event is an expansion of this definition from a particular moment to a span of time, with a clear start time and clear stop time. Expanding from moments to spans is easily achieved. All of the co-presence event data used with the tool thus far was generated using a simple threshold approach; if successive co-presence moments of two entities occur within a particular time limit, these moments are joined into a continuous event. This is done over all co-present entities' records of moments to form a set of co-presence events.

These events can be recorded very simply as a comma separated value file (.csv), and this is the format read by TIE. Each co-presence event is recorded on a separate line, and each line is formatted as follows:

> Perspective ID, ID One, ID Two, POSIX Event Start, POSIX Event End

Perspective ID refers to the string label identity of the entity—person or device—contributing the event record. Generally, the entity contributing is also one of the entities involved in the event reported, although occasionally DGDC's record exchange results in a reported co-presence event where neither entity is the reporting entity. ID One is the string label identity of the first entity involved in the event. Similarly, ID Two is the string label identity of the second entity involved in the event. The POSIX Event Start is the Unix time encoded—expanded in this case to be the number of milliseconds since January 1, 1970—start to the event, and the POSIX Event End is the Unix time encoded end to the event.

One aspect of this definition for input data is that it can be expanded for any data source that fits the model described above. Numerous other applications are conceivable as the model needs only events shared between sets of two entities with definite starts and definite ends in time. In fact, this interaction dimension could be arbitrarily redefined to involve any one dimension progression.

B. Overlay Data

The second aspect of input data involves groups (in this context, equivalently called communities) of entities in overlays. Groups are defined as a set of entities that are mutually co-present for at least one event but possibly more. As such, their representation as data is somewhat more complicated, but still quite easy to describe. First, a set of entity IDs indicates the members of the group. Second, a set of start-end time pairs denotes the one or more events associated with the community.

These records can be represented in a comma separated value file (.csv) by a series of fields, with each complete group and all group time events together on a single line. Distinct communities should be represented on separate lines, with each line formatted as follows:

> Perspective ID, # of Entities (*n*), # of Events (*m*), Entity 1, ..., Entity *n*, Event Start 1, Event End 1, ..., Event Start *m*, Event End *m*

Perspective ID is often a moniker for the algorithm used to generate the groups, but it can refer to the entity who reported this community as relevant. The # of Entities field controls how many entities are members of the group. The # of Events field controls how many meetings, or events, this community entertained. The remaining fields should be numerous enough to fulfill the expectation of the '# of' fields.

Note that the relative simplicity of this encoding model means it is simple to construct overlays of any clustering or community data related to the underlying co-presence event data. For instance, with SMS and email, detected or known communities of correspondents could be overlayed for comparison against the underlying message record. Existing gaming communities—such as 'clans' or 'alliances'—could form an overlay to see how well the self-formed communities of players correspond to their actual interactions. Alternatively, known flocks of birds or algorithmically reconstructed flocks for ornithological data could be set as an overlay.

IV. Design and Implementation

While designing TIE, we had several considerations. Our first consideration was that the information we wanted to visualize involved at its core pairs of people interacting over time. Whatever tool we built needed to involve people and their temporal interactions. Next, we had to consider the structure of communities formed from these people and their interactions. We would need to find a solution to intelligently display communities alongside the interaction timelines that were leveraged to find them. Finally, we needed a way to compare sets of communities discovered by different detection methods for the tool to be useful.

This led us to three major objectives and visualization goals in designing TIE. They are as follows:

1) Visualize the complete co-presence level interactions of a set of people
2) Intelligently overlay distinct sets of communities and their discovered interactions to allow visual investigation and comparison
3) Inspect more closely particular co-presence events (social interactions reported by multiple people)
    a) View all contributing records of a co-presence event and show the people who contribute social information about an event
    b) View event record inverse by showing how many people agree on a person's presence at the event

These design objectives guided the implementation of TIE and the data which it visualizes. All coding was done in Java, and all graphical interfaces were built using Swing (with some elements from AWT). What follows is a discussion of how each objective is met through the design and implementation of three visualizations and through a set of simple user interactions.

A. Visualizations

The design for an effective visualization of co-presence interactions is non-trivial due to its nature. For every user, there are at least three considerations: when interactions occur, who the interaction occurs with, and if the interaction is mutually perceived. Mutual perception is not assured in all cases, as it is possible for one person's recording to detect some individual, while that individual does not detect the person in return.

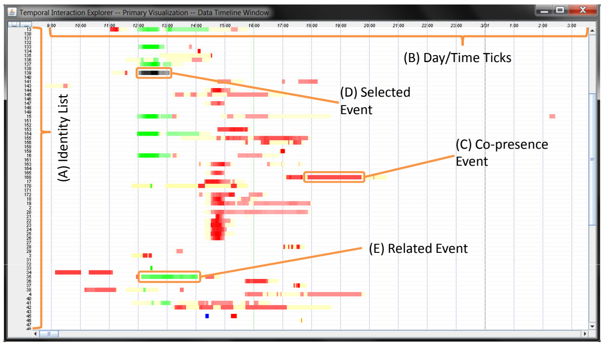

Figure 1. This figure shows how TIE represents social information contributed by specific entities in the primary visualization. (A) the contributor identity labels. (B) the ticks that fix events in time. (C) a particular co-presence event. (D) a selected event perspective, colored in shades of gray. (E) a related event perspective to the one highlighted by (D).

To clarify, as we discuss our visualization constructions, it is worthwhile defining exactly what an event and a perspective are in this context.

Event is a span of time where three or more people are co-present

Perspective is the recorded perception by a single person of who was nearby during an event

Further, two people who were present at an event may have different perspectives on the event, differences that may include when it began, when it ended, and who participated. This is why only by viewing and investigating all perspectives of people present at an event can a true understanding of the event be formed, and is also at the heart of why discovering communities from events is challenging.

In the following, we present the three main TIE visualizations. The primary visualization shows all perspectives of all people over all events at once, and addresses the first objective of showing the complete co-presence record. The isolation visualization accomplishes the first part of the third objective by enabling investigation of all the perspectives that contribute to a particular event. The intensity visualization shows, instead of contributed perspectives, the aggregate opinion on a particular person's presence at an event over all moments of the event. It shows how recorded presence opinion can change over the course of an event. This achieves the second part of the third objective. Finally, the second objective is achieved by a community overlay feature that consistently appears across all three visualizations. The relationship between communities and the event perspectives that generated them can be investigated with the same detail as the events and perspectives already described.

1) Primary: The primary visualization of TIE shows all perspectives on when all interactions occur and how many people were involved in each interaction according to each perspective. This is a subset of the three objectives, and can be achieved in a compressed three dimension space. The vertical axis is used to stack perspectives, the horizontal axis shows events in time, and color intensity of points within perspective timelines indicates the number of involved people at that moment. With this construction, we achieve our first objective of visualizing the complete co-presence level interactions of a set of people.

Figure 1 illustrates the implementation of this design. Each contributing user is represented as a separate row. Each row visualizes the timeline of co-presence events recorded by an entity, known as a perspective. All rows show the same period of time, with events inside the timeline given a specific color. A two-person co-presence event is colored light yellow, to illustrate it only. Events observed involving three or more people are colored by shades of red—indicated as (C) in Figure 1. Full red indicates the maximum number of people observed at any one time within that perspective, while lightest red indicates three person co-presence. In this way, all perspectives of all entities are clearly shown, with individual perspectives separated vertically and a unified timeline of events color-coded horizontally.

The primary visualization's construction gives insight into who the most active users are, which events are repeated in time, and what times of day or week are more active than others. This kind of exploration into overall co-presence data enables deeper understanding of recorded interactions.

To review, this first window is named the primary visualization; it displays all perspectives on events together for consideration. On selection of an event, two new windows are generated. First is the isolation visualization, which displays perspectives related to the selected event and all surrounding events of contributing users. Second is the intensity visualization, which displays an aggregation of people's perspective-based opinion on the presence of each person at every moment in time (Figure 2 shows these visualizations together).

2) Isolation: The isolation visualization is a focused subset of the primary visualization. Data is represented the same way visually; additionally, the isolation visualization displays the identities with whom interaction occurs via a 'tooltip' overlay, accessible moment-by-moment on a per perspective basis. The perspective subset is hinged around a single event from the primary visualization, and all perspectives related to that event—both in time and in presence—will be included in the isolation visualization. This achieves the first part of the third objective, which is to show the perspectives of the people who contribute social information about an event.

In the implementation, a particular co-presence event is investigated by clicking on the event in the primary visualization. Figure 2 illustrates the result of selecting an event. Beyond generating two new visualization windows, the primary visualization is also modified slightly to indicate which events from the overall timeline are involved in the isolation and intensity visualizations. Specifically, the selected event is re-colored in shades of gray, as shown by label (D) in Figure 1. The selected event is used to find three sets of entities related to this event.

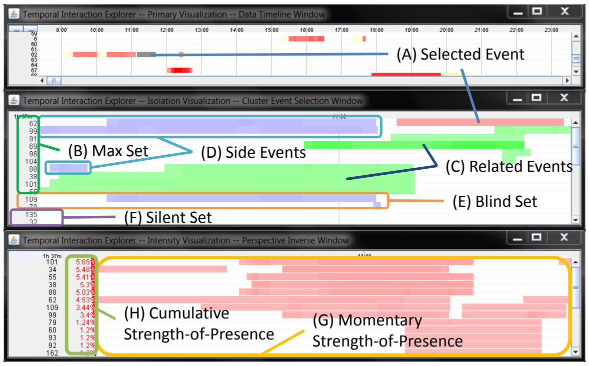
Temporal Interaction Explorer -- Primary Visualization -- Data Timeline Window
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
(A) Selected Event
Temporal Interaction Explorer -- Isolation Visualization -- Cluster Event Selection Window
(B) Max Set
(D) Side Events
(C) Related Events
(E) Blind Set
(F) Silent Set
Temporal Interaction Explorer -- Intensity Visualization -- Perspective Inverse Window
1h 37m
(H) Cumulative
Strength-of-Presence
(G) Momentary
Strength-of-Presence

Figure 2. This figure illustrates the various classifications of event perspectives and how they are represented in TIE. (A) a selected event perspective. (B) all the identities that contribute perspectives related to the selected event. (C) related events, shaded in green. (D) events part of contributing perspectives but not related to the selected event. (E) perspectives with no related events but were observed in related events. (F) observed identities that offer no perspective on the selected event. (G) the moment-by-moment aggregate opinion of all perspectives on a particular identity's observed presence. (H) the summation of all momentary opinions as a percentage scaled against total viewed time.

The first set is known as the *max set*, and it includes all those whose perspectives include events that are related to the selected event (see (B) in Figure 2). We define a related event as one that shares at least one observed identity with the selected event and overlaps it in time. Those related events recorded by users in the *max set* are colored in shades of green in both the primary and isolation visualizations. Reference Figure 1's label (E) and Figure 2's label (C) to see how they appear.

Once all related events and the perspectives that contributed them are identified, a complete time range can be denoted. Other events within this extent that are not strictly related but are within the perspectives of the *max set* are also displayed in the isolation visualization. To set them apart, they are colored in shades of blue, as shown in Figure 2's label (D).

A second set is known as the *blind set*. These are entities who were recorded in *max set* events, but who do not themselves contribute any events to the *max set*. They do however, have recorded events within the extent; these are colored in shades of blue in both the isolation and primary visualizations (see Figure 2, label (E)).

The third set is known as the *silent set*. These are users who were seen, but neither contribute to the *max set*, nor have any recorded events within the extent, as demonstrated in Figure 2's label (F).

Perspective timelines are sorted by contributing event similarity to the selected event, where similarity is a measure of observed identity set overlap and time overlap. Perspectives that do not contribute are arbitrary ordered subsequent to those that do.

The isolation visualization makes it simple to see how perspectives overlap in creating a unified perspective on a particular event. It also highlights the differences in perspectives that make identifying shared or global perspectives difficult. In evaluating DGDC, it allowed us to identify when a community has been discovered but lacks mutual confirmation from other perspectives. The moment-by-moment 'tooltip' overlay shows that the identities seen in one perspective appear as members of the *blind* or *silent sets*, demonstrating any communities detected from this event are flawed (as seen in Figure 6).

3) Intensity: The intensity visualization, instead of reproducing what is reported by each perspective, shows the level of agreement of those perspectives. The vertical axis is used to stack event participants, rather than perspectives as previously. The horizontal axis still represents time, but the color intensity of points within that timeline indicate the number of contributing perspectives who agreed on the presence of the participant at that moment. The identifiers of the perspectives that contributed to this agreement are visible through an interactive, moment-by-moment 'tooltip' overlay. The remaining goal of the third objective is achieved by this design, to show how many people agree on a person's presence at an event.

This final visualization is dynamic and interactive, allowing a tuned set of event perspectives within a specific time frame to alone form the display of aggregate opinion. This intensity of agreement, called 'Strength-of-Presence' and shown as label (G) in Figure 2, on an entity's presence at a particular point in time is shown in shades of red. If all contributing perspectives agree on an entity's presence, it will appear as full red. If only one perspective records a particular entity as present, it will be lightest red. Finally, the aggregated cumulative strength-of-presence of an entity's presence over the selected time frame is shown as a percentage bar and number surrounding each identity label (highlighted by label (H) in Figure 2).

B. Overlay

The second objective is achieved through a consistent overlay displayed in each visualization. When enabled, the overlay will show community event information as color coded bars drawn in the timelines of each participant who is a member of the community. Since each participant may be a member of multiple communities whose events overlap, these color coded bars are vertically stacked within each participant's timeline. This is done in a consistent fashion throughout all three visualizations, so that individual communities can be clearly identified and compared.

In the isolation and intensity visualizations, the overlay functionality shows where in the timeline and for which perspectives community events were identified. This allows simple comparison of the overlay communities to the raw co-presence perspective data. These features and highlights enable the isolation and intensity displays to be very useful in comparing ground truth data with detected groups.

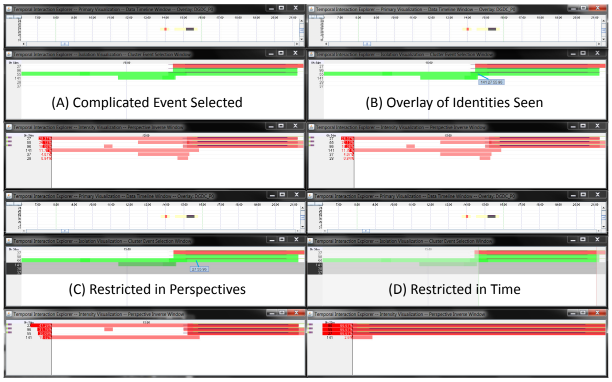
(A) Complicated Event Selected
(B) Overlay of Identities Seen
(C) Restricted in Perspectives
(D) Restricted in Time

Figure 3. This figure demonstrates a sequence of actions. Quadrant (A) shows the three visualizations after an event is selected. (B) illustrates the 'tooltip' overlay of who was seen at the moment in the perspective underneath the mouse cursor. (C) shows the effect of middle-clicking a moment in a perspective, which restricts the isolation and intensity perspectives to who was seen in that moment. (D) demonstrates the impact of selecting a time span in the isolation window. The three mutually co-present individuals and the event they share are confirmed in the intensity visualization.

With this relatively simple, but clear and uncluttered design, all three major objectives of TIE are accomplished.

C. Shared Features

All three visualizations share certain features for consistency. First, there is a border on the far-left of each visualization that lists the identity attached to each perspective time-line (see Figure 1, label (A)). Second, there is a top border of ticks indicating significant marks in time, labeled in the figure as (B) (it is automatically adjusted for zoom level in the primary visualization to prevent crowding when zoomed out). Third, the time-line perspective space of each visualization has scroll bars which appear if necessary to allow all visualization data to be accessible. Lastly, in the middle of each perspective time-line is a very light gray line to assist investigators in visually matching events within a timeline to the contributing identity.

In addition, the isolation and intensity visualizations have in the top-left corner a text label indicating the length of time represented by the time-line perspective space. It is displayed as hours and minutes; seconds precision is not included due to space considerations. Note that for the primary and isolation visualizations, selections are indicated by applying a gray mask to all unselected data and inverting the colors on unselected identities in the identity list (see Figure 4).

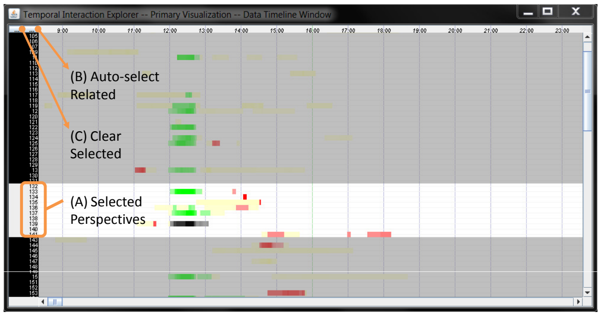

Figure 4. This figure shows a small set of perspectives selected in the primary visualization. All unselected perspectives are grayed out, and only event perspectives that involve the selected identities are drawn.

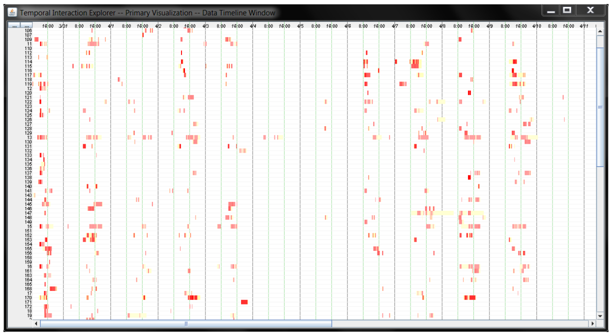
Temporal Interaction Explorer -- Primary Visualization -- Data Timeline Window

Figure 5. In this figure, a zoomed-out perspective on the data set is shown. The relatively sparse nature of our dataset can be seen from this view.

D. Advanced Interactions

The tools to control the intensity visualization set of contributing perspectives and time frame are quite simple. Initially, all perspectives and the entire time-extent as shown in the isolation visualization are included in the intensity visualization, as shown by quadrant (A) in Figure 3. Left-clicking on an identity in the identity list of the isolation perspective restricts the contributing perspectives to that entity alone. Clicking additional identities includes their perspectives in the visualization. Alternatively, middle-clicking on a particular perspective's event restricts the intensity visualization to the entities seen in that event, demonstrated by quadrant (C). Hovering the mouse over the perspective event shows which identities were recorded by that perspective in that event, shown in (B). Left-clicking in the timeline sets the starting extent of the intensity visualization to that point in time. Right-clicking similarly sets the end of the extent. In this way, a few clicks allows the intensity visualization to be restricted to just a particular set of perspectives over a carefully selected time-frame, as shown in quadrant (D).

Such control allows a much more dramatic confirmation of groups contributed by overlays, by selecting the entities whose perspectives involve the overlay group and by restricting the time-line to match the overlay group. This confirmation of mutual agreement on entity presence is a powerful resource in investigating overlay groups, which are also shown on the intensity visualization. Such confirmation is demonstrated in quadrant (D), where three overlay communities (one for each identity) are clearly validated by the restricted intensity visualization.

There is a simple set of controls for manipulating the display in the primary visualization. Left-clicking the mouse on a red-colored event marks that as the selected event, and the isolation and intensity visualization displays are reset as described previously. Left-clicking on an identity from the list redraws the primary visualization to show only events that involve the selected entity. Clicking on additional identities expands the restriction to events that involve any of the selected entities, as can be seen in Figure 4.

There are two buttons located in the top left of the primary visualization that manipulate the selection. The left button clears the current selection (identified as (C) in Figure 4). The right button adds to the selection all identities whose events involve the entities that are already part of the selection (identified as (B) in Figure 4). Note that clicking an identity that is already part of the selection removes it from the selection.

These controls for manipulating selected identities allow an investigator to explore the interactions between perspectives on a data-set scale. For instance, one could click on a particular identity, and as all other identity information is visually suppressed, only the selected identity and those other identities who directly interacted with the selected identity appear, giving insight into the overall interaction behavior of that individual.

Right-clicking the mouse in the main display cycles between configured overlays. This allows very simple comparison between group detection algorithms, as well as the capacity to investigate particular overlays by themselves. As these overlays pervade each visualization, switching between overlays allows comparison at each level of co-presence and community interaction.

For mice with a middle button, clicking it on the main display cycles the zoom. By default each horizontal pixel represents a minute; each click modifies this ratio. The second zoom level is one pixel represents two minutes; the third level is one pixel for five minutes, and so on. By cycling through the zoom levels, it is possible to see the entire span of time-line data in one screen. This gives TIE users unparalleled ability to analyze the whole dataset at a glance, as demonstrated in Figure 5.

V. Evaluation

In our own research, TIE has proven useful in evaluating our community detection algorithms. Using TIE, we rapidly identified a significant algorithm implementation problem that went undetected for quite some time. A coding error in one version of DGDC resulted in groups that spanned several

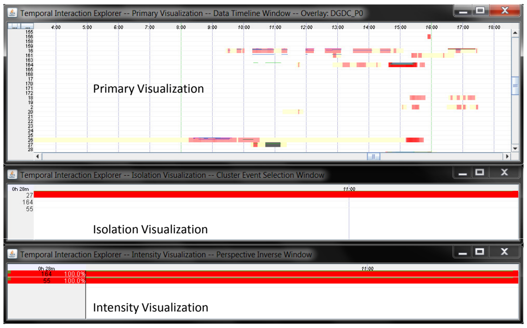
Temporal Interaction Explorer -- Primary Visualization -- Data Timeline Window -- Overlay: DGDC_P0
Primary Visualization
Temporal Interaction Explorer -- Isolation Visualization -- Cluster Event Selection Window
Isolation Visualization
Temporal Interaction Explorer -- Intensity Visualization -- Perspective Inverse Window
100.0%
100.0%
Intensity Visualization

Figure 6. In this figure a selected event perspective reports three people seen, but two are silent as shown in the isolation visualization. The intensity visualization reinforces this, as the person claiming the event is absent from this visualization.

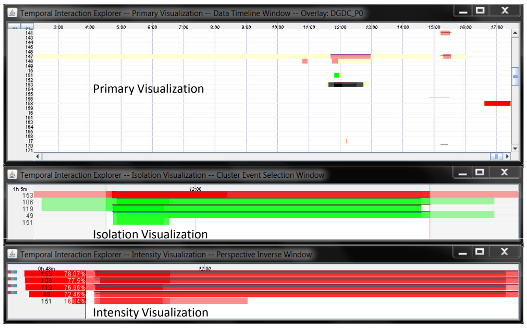
Temporal Interaction Explorer -- Primary Visualization -- Data Timeline Window -- Overlay: DGDC_P0
Primary Visualization
Temporal Interaction Explorer -- Isolation Visualization -- Cluster Event Selection Window
Isolation Visualization
Temporal Interaction Explorer -- Intensity Visualization -- Perspective Inverse Window
Intensity Visualization

Figure 7. This figure demonstrates a straightforward confirmation of mutual co-presence. Four people contribute significant perspective information in the isolation visualization, and their perspectives confirm each other's mutual presence as demonstrated by the high cumulative strength-of-presence in the intensity visualization.

weeks of continuous time. These spurious groups appeared as color coded bars that spanned the entire visualization timeline without break when the overlay was active. As typical overlay communities represent constrained events (often less than three hours in length), events spanning weeks of time made identifying the error's presence very simple. Prior to TIE, this small number of affected groups remained undetected in spite of numeric analysis and manual inspection of the raw comma separated data. With TIE, setting the distributed algorithm results as an overlay allowed us to identify the existence of an error, which we quickly correctly.

Validating the results of DGDC provided further insights. In DGDC, each person generates groups based on information they have collected. These groups are not confirmed by other people, which can result in groups that are only observed in a single perspective. As TIE distributes overlay group events over all the identities claimed to be in the group, the absence of confirming perspective information can be seen when only one perspective is shown but three identities are listed in the isolation visualization. The other two identities are silent, lacking any confirming perspective information. This situation can be seen in Figure 6.

In some cases, confirmation of results is clear from quickly referencing the data shown in the isolation and intensity visualizations against an overlay, as shown in Figure 7. Here, all four perspectives claim to see the three other identities involved in the group, and the group from the overlay correspondingly appears in each perspective, meaning the algorithm correctly identified this group event. The one person appearing to 'pass by' was not included in any overlay group here, as one might expect of a proper community detection algorithm.

In other cases, confirmation is not so clear. As shown in Figure 8, multiple overlay groups cover what appears to be a single event in the source data, but an event that is not mutually confirmed by all claimed participants (as demonstrated in the intensity visualization). Two people were unable to see each other but saw all other members of the group. Likely they are one community, but somewhat distant from each other during detection.

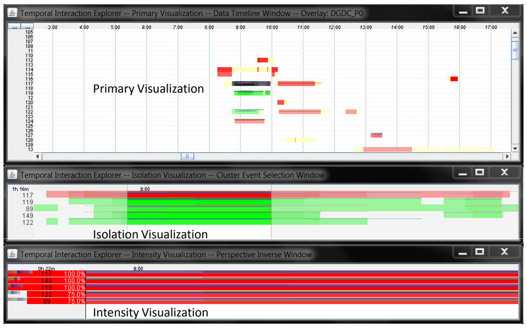
Temporal Interaction Explorer -- Primary Visualization -- Data Timeline Window -- Overlay: DGDC_P0
Primary Visualization
Temporal Interaction Explorer -- Isolation Visualization -- Cluster Event Selection Window
Isolation Visualization
Temporal Interaction Explorer -- Intensity Visualization -- Perspective Inverse Window
100.0%
100.0%
100.0%
75.0%
75.0%
Intensity Visualization

Figure 8. Unlike Figure 7, the event seen in this figure involves five contributing perspectives but only three of the five perspectives mutually confirm each other's presence. Two people, probably at opposing edges of the group, did not detect each other. This is reflected by their 75% cumulative strength-of-presence in the intensity visualization.

Generally, the intensity visualization is helpful in determining the overall opinion about the presence of a particular entity at a particular event. This is useful for identifying which users may have been just at the edge of co-presence detection, or perhaps were near the middle of two groups of entities unable to clearly detect each other.

The combination of tools within TIE gives an investigator many resources for evaluating the strengths and weaknesses of particular community detection methods and implementations, and forming improvements or tweaking thresholds while designing new or improved methods.

VI. Conclusion

Due to people mobility, unpredictable meeting patterns, and co-presence recording issues, detecting co-presence based communities with high accuracy is difficult. Currently, there is no simple way to verify the accuracy of algorithms for co-presence community detection or to compare the results of such algorithms.

The Temporal Interaction Explorer (TIE) is a powerful companion in the search for clear confirmation of community detection results. Using the three distinctly purposed visualizations, an investigator can quickly examine how well a particular group or set of groups compares with the ground truth of actual recorded co-presence data. This allows existing methods to be efficiently evaluated for strengths and weaknesses, and allows new methods to undergo equivalent scrutiny without falling to the ambiguity of a lack of participant confirmation.

TIE, while useful in its current incarnation, has the potential for many future improvements. Adding more tools to manipulate the overlay within the application is one goal, as this would help reduce the potential for error introduced by manually manipulating the overlay input files to restrict which portions of the overlay to display. As well, enabling session saving, results marking, and dynamic loading/unloading of overlays and source data will greatly enhance the usability of the tool.

Our hope is that this tool becomes useful in more contexts than just our field of interest. Social scientists, ornithologists, and other researchers working with data that fits the models described could benefit from this tool and the ease it lends to dataset exploration and results confirmation.

References

[1] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[2] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A. Pietilainen, and C. Diot, "Are you moved by your social network application?" in *Proceedings of the first ACM Workshop on Online Social Networks*, 2008, pp. 67–72.

[3] V. Kostakos and J. Venkatanathan, "Making friends in life and online," in *IEEE International Conference on Social Computing*, Aug 2010, pp. 587–594.

[4] S. Pan, D. Boston, and C. Borcea, "Analysis of fusing online and co-presence social networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, Mar 2011, pp. 496–501.

[5] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: User-Controlled P2P Social Data Management for Socially-Aware Applications," in *Proceedings of the 11th ACM/IFIP/USENIX International Middleware Conference(Middleware 2010)*, Dec 2010, pp. 212–231.

[6] J. A. Schmidt, J. M. Hektner, and M. Csikszentmihalyi, *Experience Sampling Method*. Sage Publications, 2006.

[7] E. Paulos and E. Goodman, "The familiar stranger: anxiety, comfort, and play in public places," in *Proc. of the SIGCHI conference on human factors in computing systems (CHI '04)*, Apr 2004, pp. 223–230.

[8] S. Mardenfeld, D. Boston, S. Pan, Q. Jones, A. Iamntichi, and C. Borcea, "GDC: Group Discovery Using Co-location Traces," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, Dec 2010, pp. 641–648.

[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," in *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, ser. ASPLOS-X. ACM, 2002, pp. 96–107.

[10] A. Gupta, S. Paul, Q. Jones, and C. Borcea, "Automatic identification of informal social groups and places for geo-social recommendations," *International Journal of Mobile Network Design and Innovation*, vol. 2, no. 3, pp. 159–171, Feb 2007.

[11] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun 2005.

[12] M. Newman, "Detecting community structure in networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, Mar 2004.

[13] J. Heer and D. Boyd, "Vizster: Visualizing online social networks," in *IEEE Information Visualization (InfoVis)*, 2005, pp. 32–39.

[14] G. Csárdi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006.

[15] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y. Boey, "Analysis and visualization of network data using JUNG," *Journal of Statistical Software*, vol. 10, pp. 1–35, 2005.

[16] M. Bostock and J. Heer, "Protovis: A graphical toolkit for visualization," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2009. [Online]. Available: http://vis.stanford.edu/papers/protovis

[17] J. Heer, S. K. Card, and J. A. Landay, "prefuse: a toolkit for interactive information visualization," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '05. ACM, 2005, pp. 421–430.

[18] E. Wohlfart, W. Aigner, A. Bertone, and S. Miksch, "Comparing information visualization tools focusing on the temporal dimensions," in *Proceedings of the 2008 12th International Conference Information Visualisation*, 2008, pp. 69–74.

[19] C. Daassi, L. Nigay, and M. Fauvet, "A taxonomy of temporal data visualization techniques," pp. 41–63, 2005.

[20] I. Johnson and A. Wilson, "The timemap project: Developing time-based gis display for cultural data," *Journal of GIS in Archaeology*, pp. 123–135, 2003.

[21] S. K. Lodha and A. K. Verma, "Spatio-temporal visualization of urban crimes on a gis grid," in *Proceedings of the 8th ACM international symposium on Advances in geographic information systems*, ser. GIS '00. ACM, 2000, pp. 174–179.

[22] J. Wood, J. Dykes, A. Slingsby, and K. Clarke, "Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup." *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1176–1183, 2007.

[23] K. Vrotsou, J. Johansson, and M. Cooper, "Activitree: Interactive visual exploration of sequences in event-based data using graph similarity," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 945–952, November 2009.