ABSTRACT
Analysis of massive track datasets is a challenging problem, especially when examining n-way relations inherent in social networks. In this paper, we use the Mitsubishi track database to examine the usefulness of three types of interaction features observable in tracklet networks. We explore ways in which social network information can be extracted and visualized using a statistical sampling of these features from a very large track dataset, with very little ground truth or outside knowledge. Special attention is given to methods that are likely to scale well beyond the size of the Mitsubishi dataset.

1. INTRODUCTION
Our objective in this paper is to explore ways in which information about social networks can be extracted from very large datasets. Social networks are relatively long-term relationships that may arise through common interests, including work projects, hobbies, or simply common destinations in a physical environment. For very large data sets, it is unrealistic to rely on exhaustive analysis to uncover these relationships. We therefore focus on statistical methods requiring only modest computing resources to find groupings of interest in the data. In addition, we use these methods in a blind fashion, relying on only a minimum of domain knowledge (i.e., the fact that this is an office area, and the semantic attachments found on the floor plan).

Social network analysis is an area that is receiving increased interest. As sensors proliferate, an increasing volume of data is generated that, implicitly at least, contains information about social groupings among sensed individuals [1]. Knowing these groups can be important for a variety of reasons. For example, the DARPA CALO (Cognitive Assistant that Learns and Observes) project [2] exploits knowledge of social groups to anticipate user needs and scheduling conflicts. Physical security and safety applications can use knowledge of social groups to identify unusual, dangerous or threatening behavior.

The MERL (Mitsubishi Electric Research Lab) dataset [3] is an excellent example of the result of daily use of large numbers of sensors in an office environment. This is a dataset of activations of motion sensors distributed through the MERL environment, recorded during the course of 1 year. The sensors are mounted in the ceilings of the 7th and 8th floor of Mitsubishi Electric Research Laboratories in Cambridge MA. Although sensors are distributed in the stair areas, we do not assume that tracks are coherently represented across the two floors. Hence, most of our analysis is restricted to the 8th floor. The sensors generate a high volume of data each day, resulting in a dataset that is difficult to process exhaustively in a short amount of time. The situation is further complicated when one wishes to examine n-way social networks inherent in such a dataset. The data structures of interest, as defined in the MERL dataset, are tracklets and tracklet graphs. Tracklets are temporally ordered chains of adjacent sensor activations, and typically represent a person or group of people traveling in a common path. Tracklet graphs are partially ordered graphs of tracklets that describe the possible trajectories of one or more movers in the environment. The tracklet graph represents all possible worldliness for groups of people starting at one set of sensors and ending at another. This makes explicit the ambiguities that arise when tracking multiple nearby individuals. No appearance information is available from the sensors.

![Figure 1: Rendering the MERL dataset. Top: timeline view with tracklets shown as green bars; Bottom: spatially registered view with tracklets shown in red.](image)
with merges and splits every time the coarsely sampled paths cross, it is not always possible to determine the actual starting and terminating points for a specific trip through the building. In spite of this, there are multiple features of the tracklet networks that can, with time and many samples, build a statistical picture of the social organization.

We explored three features that roughly correspond to three different aspects of social behavior: visiting another person, attending meetings with another person, and traveling with another person. The features are represented in the form of three matrices: (1) a visit matrix, the number of visits between each pair of points; (2) a meeting matrix, the number of times two points were destinations of trips from the same meeting room at roughly the same time; and (3) a travel-with matrix, the number of times two points were destinations of trips that split from a single, shared tracklet. Using information theoretic measures and graph cuts, we examine the usefulness of these features, the soundness of our sampling process and aspects of social organization in the MERL environment.

1.1 The Sentient Environment System

An existing system was adapted for use with the MERL dataset. SRI’s Sentient Environment system is a multi-sensor stereo-based tracking system, built on top of the FREEDIUS geospatial IU workbench [4], that is currently used for experiments in object and event recognition. Acquired tracks are archived into a database for later visualization and analysis. Although the track data models are somewhat different, it was straightforward to import the MERL dataset into a database and modify the Sentient system to ingest and visualize sensor placement and tracklets. The Sentient system was written largely in Common Lisp. Track objects and their sources were implemented as classes that reflect the data models used for representation within the track sources (a database in this case). Class specialization allows different data models to be represented and rendered in a common framework. The two most relevant rendering styles are temporal, using a timeline of events that each typically have spatial extent, and spatial, using (for example) the floor plan map to highlight sensor cells as they are activated. Figure 1 shows a view of the GUI that permits interaction with the MERL database. At the top is a timeline view showing tracklets as green bars within a time interval. The bottom view shows the same tracklets as spatial objects overlaid on the MERL floor plan. All objects are mouse-sensitive. The Sentient system is equipped to use SQL databases, so the MERL dataset was imported to a MySQL database for visualization and analysis. Tracklets are mouse sensitive, allowing queries of sensor ID, timestamp, spatial position, and other related properties of the observations.

2. Sampling and Accumulators

The size of the MERL dataset makes sampling an attractive way to analyze global properties of the dataset. We approach the sampling issue by constructing class hierarchies of accumulators and sampler classes that can be used in combination to gather statistics of interest. This approach is similar in spirit to that used by Stauffer and Grimson [5]. Stochastic models of activity allow us to represent distributions that correspond to “normal” activity (assuming stationary processes). The hierarchy of timeline sampler classes represents different distributions of time as a random variable. Timeline accumulators and spatial accumulators allow statistics to be gathered as functions of time or space. Accumulator objects are one or two-dimensional histograms of some quantity, depending on the class of the accumulator. Each sampler iteration draws a small number of samples from the dataset and changes an accumulator appropriately. Accumulator histograms can be normalized to obtain frequencies, and they can be visualized as projected heat maps or graphs.

2.1 Temporal Sampling and Accumulation

Social networks and personal attributes are assumed to be stationary properties of the dataset, or within some time interval in the dataset. These attributes can only be discovered when the individuals involved are active. We therefore use a timeline sampler that is biased toward times of high activity. A bootstrapping approach is used to generate the desired distribution. A timeline sampler with a uniform distribution is used to draw time samples. A timeline accumulator is also constructed whose histogram maps onto a duration of one week, with a bin size of one minute. Using the timeline sampler, sensor hits are accumulated modulo one week, so that the accumulator learns a model of activity for the typical week at MERL. The accumulation process implicitly assumes a small, fixed query constraint interval for the SQL database. There is a tradeoff in interval size vs. computation time. Longer intervals will produce more samples, but will incur longer computation times per interval. For the experiments described in this paper, an interval of 10 seconds was empirically chosen to provide a reasonable tradeoff between per-interval sample count and per-interval computation speed.

Figure 2 shows the results of a sampling run that accumulated statistics over 1.5 million sensor hits. This figure clearly shows the diurnal progression of activity during business days (0 is midnight Sunday), with the expected reduction in activity during nights and weekends. The distribution shown in Figure 2 is then used to construct a timeline sampler. Let \( p(t) \) be the activity frequency at time \( t \). We compute the cumulative probability of activity over one week:

\[
C(t) = \sum_{j \leq t} p(t)
\]

and use this function in combination with a uniform sampling of weeks to generate a random timestamp whose distribution modulo one week is that shown in Figure 2. This biases the sampler toward times of higher activity.

In practice, sampling was done in parallel, in independent processes running on several CPUs. Although sampling runs were typically on the order of 20000 per process, this was usually far beyond the point of convergence based on the entropy measures discussed in Section 2.3.

\[ \text{Footnote: Though negligible for isolated queries, per-query latencies arising from database access and from computational overhead could significantly increase the time required to gather sufficient samples.} \]
Using the activity-biased timeline sampler, we can accumulate statistics over the set of sensors. One of the simplest spatial accumulators creates a histogram of activity for all sensors. Spatial histograms can be rendered as a kind of heat map as seen in Figure 3, where the color blue represents relatively low frequencies of sensor activity, progressing to red, indicating high frequency. The kitchen (seen in orange) appears to be a major hub of activity. All spatial accumulators used in this paper are rendered as a spatial heat map overlaid on the floor plan.

2.2 Spatial Sampling and Social Relations

Social relations in groups are characterized in part by the pairwise interactions of members of each group. We expect people who work together to meet and interact, either by visiting offices or by attending meetings. Another potentially observable behavior is walking together (while going to lunch, for example). This might reveal social interactions that are not necessarily work-related, but can be correlated to groups with a common purpose.

To extract properties of individuals and their relationships, we implemented accumulators that gathered pairwise statistics over the dataset using the activity-biased temporal sampler. These accumulators contain NxN histograms (where N is the number of sensors) that count the number of times a particular relationship was found across pairs of sensors. We experimented with three different classes of accumulator: 1) the visit accumulator, 2) The “travel-with” accumulator, and 3) the “meet-with” accumulator. Pairwise accumulators are visualized here as projections of the histogram matrix onto a particular sensor’s row (in the case of destination maps) or column (in the case of source maps).

2.2.1 The Visit Accumulator

A trip is defined as a tuple consisting of one source sensor and all possible terminal sensors that are connected to the source through a tracklet graph. The trip represents the possible destinations for a single source sensor in a tracklet graph. During accumulation, each trip adds 1/N to each of the destination bins, where N is the number of destinations.

The visit matrix A represents the number of visits made directionally between pairs of sensors, weighted by the number of alternatives found in each case. After sampling converges, each element $A_{ij}$ is a measure of the frequency of travel from sensor $i$ to sensor $j$. Hence, this matrix is asymmetric, although round trips are so common that it is almost symmetric, as is obvious from the raw matrix shown in Figure 4. In this figure, clusters of associated sensors are visible, the major component of which is due to the separation of the set across two floors.

Rows or columns of the visit matrix can be selected to display the source or destination set for a given sensor. If a particular sensor row or column is selected from the matrix, the selection taken as a histogram contains the frequency with which that sensor is a source or destination. This can reveal common sources or destinations for putative occupants of nearby rooms. Figure 5 shows the destination map obtained by selecting sensor 277 from the visit matrix. Sensors colored in red are more likely to have been trip destinations for sensor 277. To the extent that sensors coincide with offices, one can in principle assign likelihoods to various properties, e.g., the sex of the office occupant (as characterized by which rest room they visit). The visit matrix attempts to capture relations that indicate where individuals typically go and perhaps with whom they usually communicate. In this case, the individuals starting at sensor 277 typically visit other sensors only in the immediate area. They also frequently visit the kitchen, elevators, rest rooms, and stairwells. They visit the other wings relatively rarely.
2.2.2 The Travel-With Accumulator

We sampled a second statistic, called the “travel-with” matrix, that represented the count of paired trips. This relationship is symmetric, and the accumulator is incremented whenever two terminal sensors had a common ancestor tracklet. This indicates that two movers were traveling together for some time interval. Our assumption in this case is that pairs of individuals who work or socialize together will travel together more frequently. Figure 6 shows the travel-with frequencies for sensor 277. Note that high-frequency sensors are more spread out in this case, especially along the hallways near sensor 277 (highlighted by the arrow). This may be indicative of chance hallway encounters, especially if, as Figure 3 suggests, the hallway to the kitchen is relatively busy. On the other hand, sensor 277 also appears to travel with one or more of the occupants of two other large offices near the star. One possibility is that these are managers who frequently visit and move with each other. Alternately, the activity may arise from administrative assistants in adjoining offices.

2.2.3 The Meet-With Accumulator

In this accumulator, bins are incremented whenever they correspond to a pair of terminal sensors of trips that started in the same meeting room (as defined by the labels on the floor plan). In addition, the trips must have terminated within 15 minutes of each other. This attempts to capture situations where a group of individuals leaves a meeting (whether scheduled or impromptu). Figure 7 shows one selection of the accumulator for the 8th floor meeting room, at sensor 329 (indicated by the arrow). This pattern suggests that the individual(s) associated with sensor 329 are more likely to meet with neighbors and those diagonally opposite the machine room, but are less likely to meet with occupants of the intervening hallways. This may indicate that the occupants of offices near these sensors share common technical interests, or projects.

The use of projections and heat maps allows us to quickly visualize the properties of individual sensors, but more global properties of the dataset are often desirable. To that end, we need methods that reveal more about the global organization inherent in the collected distributions.

2.3 Entropy

Entropy estimates can be used to assess convergence of the samplers and to gauge the relative organization of computed distributions. We use the naïve Shannon entropy estimator:

\[ H = -\sum_{i=1}^{m} p_i \log p_i \]

where \( p_i \) is the frequency corresponding to bin \( i \) in our accumulator.

Entropy computation is prone to sampling error, and several estimators have been proposed to address this problem [6,7,8]. Fortunately, massive datasets affords us the luxury of gathering enough samples to provide good entropy estimates. In the case of the MERL dataset, the number of bins in our accumulators is typically 213 for simple spatial accumulators, or \( 213 \times 213 = 45369 \)
for 2-way interaction accumulators. Using the Miller-Madow entropy correction term [6] for $m$ bins and $N$ samples:

$$ e(m,N) = \frac{m-1}{2N} $$

we can estimate the sampling error as a function of $N$. For 2-way interactions, even a modest set of 500000 samples results in a correction of only 0.05 bit (out of a maximum entropy of around 11 bits). The naïve estimator almost surely converges to the true entropy in the limit as the number of samples increases. Figure 8 illustrates this for three instances of the pairwise accumulators discussed in the previous section. For the visit (blue) and travel-with (green) accumulators, the convergence of entropy to asymptote is clear and generally smooth. A purely random distribution exhibits the maximum entropy of 10.8 bits for this sensor suite. The travel-with accumulator converges to approximately 9.2 bits of entropy, while the meet-with accumulators range between 8.8 and 9.1 bits of entropy. Valid meet-with events appear to be less common and require more sampling runs for convergence. When they do occur, however, a large number of samples are typically available to contribute to the accumulator, providing discrete adjustments to the computed entropy. This may explain the slower staircase progression seen in Figure 8. Coincident travel could often arise by chance in busy hallways, biasing the entropy upward. In particular, Figure 3 shows that the hallways connecting to the kitchen are particularly busy, and this may be a prime source of random coincident travel. Finally, the visit matrix converges to a value of around 7.8 bits and appears to exhibit the most organized distribution of the three classes of accumulator.

![Figure 8: Convergence of entropies for matrix accumulators.](image)

All accumulators appear to be more organized than chance. Although these samplers were run well past 10000 iterations, in two of the three cases, convergence was achieved at around 1500 iterations. On a dual-core 3.2 GHz Pentium workstation, this usually takes 5-10 minutes. If parallelized operation is required (for longer sampling runs in larger datasets, or when the per-sample computation is expensive), entropy estimates can be compared across processes to help determine convergence. One expects only small variations in entropy across the independent samplers when convergence is achieved.

Our analysis assumes a stationary dataset. Events can occur that significantly change the interaction patterns in the dataset. For example, an office renovation might require movement of individuals to new offices. If we assume that such events are relatively rare, it should be possible to perform piecewise entropy estimation over separate intervals in the dataset, to segment periods between significant changes in interaction patterns.

### 2.3.1 Mutual Information

We compared our pairwise statistics with each other to better understand how independent the statistics were. Mutual information is defined as follows:

$$ I(X,Y) = H(X) + H(Y) - H(X,Y) $$

where $H(X,Y)$ is the joint entropy across two variables. One interpretation of mutual information is that it represents the degree to which samples from one distribution allow us to predict samples drawn from the other. In some sense, it represents the information redundancy of the distributions. Computing $I(X,Y)$ for one instance of visit and travel-with accumulators yields a mutual information of around 11 bits, indicating a high degree of redundancy. Similar values are seen when comparing visit and meet-with accumulators.

### 2.3.2 Per-Sensor Entropies

By computing the entropies of accumulator projections, we can get a measure of the organization of an individual’s visit patterns. Figure 9 shows each sensor color-coded according to the entropy of its source distribution. For each destination sensor, this illustrates the relative organization of the source sensors whose tracklets make trips to that sensor. Colors toward the red end of the spectrum show sensors that have a diffuse source profile; they tend to be destinations for many other sensors. The stairs, rest rooms, and elevators predictably show high source entropies. Sensors that are green or blue have more organized source patterns. To distinguish whether these patterns are simply undersampled (as opposed to highly specific), we can examine the source marginals as shown in Figure 10.

![Figure 9: Trip source entropies color-coded for each sensor. Red indicates diffuse source patterns, while blue indicates organized source patterns.](image)

Sensor 398 has been highlighted in Figure 10 to show an example of one sensor that has relatively low source entropy, but at the same time has a modest histogram value (700 weighted accumulator hits, which typically implies well over 1000 trips). The relatively low entropy value is probably due more to an organized source pattern than to undersampling.
Figure 10: Trip source marginals, showing the relative frequency that a given sensor is a destination. Sensor 398 is highlighted with an arrow.

Figure 11 shows the source map for sensor 398. Sensors shown in blue are rarely sources for this sensor. Sensors shown in red are most likely to be sources for sensor 398. High-frequency sources for this sensor predictably include the stairwell, rest rooms, elevators, and kitchen. A few offices along the nearby hallway are also significant sources for this sensor. Sensor 398 straddles an office and a lab. The source pattern suggests that it is capturing the activity of one or more individuals that form a fairly tight social group, perhaps focused on a limited number of specific projects, or on special purpose hardware found in the nearby lab. This group does not interact with the 7th floor.

2.4 Segmentation of Groups

From the information contained in the visit matrix, we are interested in discovering social groups among the inhabitants of the environment. To do this, we first extract a subset of sensors that are most likely to correspond to locations of personal offices on the 8th floor. A total of 65 such sensors are extracted, and from these, we construct a symmetrized visit matrix \( A' \) among these sensors (called the office visit matrix).

We hypothesize that the office visit matrix contains information about the level of social contact between the office inhabitants, and thus can provide interesting information about social networking, group organization etc. If this is true, groups can be found by segmenting the set of offices into multiple clusters such that visits among each cluster appear much more often than visits across the clusters.

Formally, this can be posed as graph partitioning problem, and a family of methods based on “graph cuts” can be used to yield the clusters. We chose the normalized-cut algorithm [9] since it directly attempts to find clusters that maximize visits among the clusters and at the same time minimize visits across the clusters. The objective function to be minimized when partitioning the offices into two groups \( X \) and \( Y \) is:

\[
Ncut(X,Y) = \frac{cut(X,Y)}{vol(X)} + \frac{cut(X,Y)}{vol(Y)}
\]

where \( cut(X,Y) \) represents the total visits between two groups, \( vol(X) \) represents the total visits to/from members of group \( X \). Despite being a NP-hard problem, the normalized cut has an efficient approximate solution derived from the generalized eigenvectors of the Laplacian matrix of the visit matrix/graph, and has been widely applied in the context of image segmentation.

Figure 12 shows the result of partitioning the office sensors into two groups. We redraw the visit matrix after grouping offices in the same groups together in Figure 12 (b), and the effect of the grouping can be seen clearly. It is difficult for us to interpret the floor plan showing the two clusters without any knowledge about the office inhabitants. However, we can hypothesize that there appears to be two broad organizational/social groups occupying this floor. Figure 12 (c) shows the floor plan of the two groups in red and green.

Figure 12: (a) Office visit matrix, (b) Office visit matrix after clustered into 2 groups, (c) Floor plan showing the two clusters of “office sensors”.

If we look at the red group, Figure 12 (b) shows that there is one sensor cell that acts as a “hub” for the group, i.e. it is very well associated with other members in the red group, and at the same time disconnected from members of the other group. The location of this “red hub” (sensor id 442) is shown in Figure 12 (c).
Outliers in Figure 12(b) correspond to cross-group visits, and two such cross-groups can be seen. The first cross group links the red sensor cells (id 279, 280) and blue sensor cells (id 271,272) in the top right corner of level 8th floor plan, forming a sub-group among themselves. This agrees with the description of the environment indicating this area is used by managerial and administrative people. The second cross-group corresponds to a group of office sensors in the bottom left of the 8th floor (id 421,422,417). This could correspond to a small social and/or work-related group among the near-by offices and lab.

2.5 Individual Properties

Hubs and cross-groups identify potentially important individuals within the organization. The visit maps for the corresponding sensors reveal interesting properties of the putative individuals involved. Figure 13 shows the destination map for sensor 442, the hub of the larger cluster, while Figure 14 shows the same sensor’s source map. The visit relations for this sensor are asymmetric: this sensor is predominantly a destination rather than a source. Much of the visit activity comes from what appears to be a nearby bank of cubicles. The individual at sensor 442 often uses the 8th floor meeting room and is probably (although not certainly) male, based on the rest room frequencies. His role remains a mystery, but perhaps he oversees or supports the personnel who are housed in the nearby cubicles.

The same kind of analysis can be applied to the cross-group sensors. Figure 15 shows source map for sensor 272, which appears to cover the entry to a cubicle near the large offices on the 8th floor. The individual associated with this sensor is probably female, judging by the pattern of rest room frequencies. Most importantly, the visit patterns are concentrated around the large offices but are spread across members of the two major groups distributed there. One explanation for the cross-group pattern is that this person supports management at a high level, where one would expect more work-related cross-group interaction than at lower levels of the organization. This is further supported by the nontrivial pattern of 7th floor sources.

Finally, Figure 16 shows the source map for one of the cross-group-2 sensors isolated in Figure 12. The interactions for the sensors in this cluster are well localized. It is a relatively small group that appears to use the nearby lab, but does not interact as much with other groups on the 8th floor.
3. Conclusions
We present tools for extracting information about social networks and individuals in the MERL dataset. Our analysis looks for invariant, global properties of the group under surveillance. Two-way co-occurrences, spectral analysis, and entropy measures can all contribute to understanding the nature of social interactions in the groups. It is important to acknowledge that our analysis is blind, in that we rely only on limited ground truth (like the set of sensors near offices). We could conceivably improve the accuracy of our analysis by including sensor-office correspondence likelihoods (e.g., the degree to which a given sensor is associated with a specific office). Even so, we are able to retrieve some specific information about individuals and the nature of their interactions within their respective groups.

To improve tractability, especially for massive collections, our approach to analysis relies on statistical sampling over the dataset, driven primarily by the notion that true is the best filter for managing dataset size and avoiding excessive computation. Our samplers typically ran for only a short time before convergence, suggesting that this approach can scale to datasets much larger than the current MERL collection. In fact, entropy-based analysis will generally improve as dataset size increases (although estimators exist that specifically address small sample sizes). Our results capture typical interactions, and can serve as a baseline for detecting anomalous behavior. Although pairwise statistics are gathered by randomly sampling time intervals, the entropy convergence rate depends on gathering enough samples for all bins of the NxN affinity matrices described here, and hence is likely to be O(N²).

It is also important to acknowledge other approaches to mining data within large datasets. Sampling will miss possibly important specific events. An alternative approach to analysis of massive datasets is to use context to focus attention on highly constrained subsets of the dataset. When a dataset has sufficient semantic context associated with it (identities of specific movers, for example), it is possible to construct highly constrained queries that will reduce the volume of data that needs to be analyzed.

In the absence of ground truth, it is difficult to know how accurate our assessments are. Certain clusters make sense, however. In particular, one expects that in an office domain, groups working on common projects will for the most part be in the same area, but as new people are brought into (or hired on) the project, their offices will not necessarily be near the existing project group. The floor plan itself provides valuable information, such as the locations of rest rooms, cubicles, and large offices. This provides context that helps us make sense of the analysis, and serves as a sanity check for the results.

Although this paper assumes a certain correspondence between offices and sensors, it may be possible to exploit the same machinery in the analysis of less constrained indoor video track data. SRI’s Sentient Environment system generates track data that can be discretized in a manner analogous to the MERL data. In this case, though, there is no clear association between individuals and sensors. However, it may be possible to segment tracks with common spatiotemporal properties to segment activity into groups corresponding to activity type (e.g., organized activity associated with a meeting, or birthday celebrations). We are currently experimenting with the approaches described here on a 1-terabyte collection of indoor video tracking data from the Sentient Environment system.

4. REFERENCES