Crowd Textures as Proximity Graphs

Claudio Martella and Maarten van Steen, VU University Amsterdam
Aart van Halteren, VU University Amsterdam and Philips Research
Claudine Conrado, Thales Research and Technology
Jie Li, Technical University Delft

ABSTRACT

We are only starting to understand how people behave when they are part of a crowd. This article presents a novel approach to the study and management of crowds. The approach comprises a device to be worn by individuals, an infrastructure to collect the information from the devices, a set of algorithms for recognizing crowd dynamics, and a set of feedback strategies to intervene in the crowd. A fundamental element of our approach is to consider crowds in terms of their *texture*. The crowd texture is represented through the *proximity graph*, a data structure that captures the spatial closeness relationship between individuals over time. We address its properties and limitations, a system architecture to measure and process it, and a few examples of insights that can be obtained from analyzing it.

INTRODUCTION

We may be witnessing the dawn of a radical change in the social sciences. With the availability of a vast amount of online data through online social networks, and the usage of wearable sensing and computing devices, scientists no longer need to rely only on self-reported data on social behavior. The availability of these digital traces now form an important asset for the field of computational social science [1], which investigates complex social systems through quantitative modeling. Research in this field has so far focused mainly on the analysis of online social networks and patterns emerging from face-to-face interactions [2]. More recently, attention is also being given to the study of crowds and crowd dynamics.

The term crowd dynamics is used to refer to patterns of crowd movement, and more precisely to "the coordinated movement of a large number of individuals to which a semantically relevant meaning can be attributed, depending on the respective application" [3]. Examples include a queue of people, the formation of unidirectional lanes in bidirectional pedestrian flows, the intersection of these lanes, or a group of people at a specific location. We use the phrase *texture of a crowd* to express the spatio-temporal relationships resulting from the interdependencies in the social fabric of a group of people. At this stage we may not yet fully understand what potential insights can be derived from crowd textures, but in any case it enables the study of emergent spatio-temporal and social behavior of people in a crowd. For example, it allows one to question to what extent a group is dispersed in a crowd.

Discovering and investigating the texture of a crowd is at the heart of research on crowd management. As a prerequisite, it is essential to adequately represent texture. A common approach is to simply place cameras and collect their images and videos. There are a few drawbacks: the computational cost of video analysis limits the scale at which experiments can be run, cameras can be affected by complications such as occlusion and incomplete coverage, and privacy issues can emerge when the footage is recorded from real-world events.

As an alternative, on-body sensors can be used. Such sensors can collect rich information about the individual behavior of each subject. We believe a better understanding of crowd dynamics can be achieved by sensing from *within* the crowd instead of from an external observation point, as the sensing is based directly on the individuals forming the crowd. An example of this approach can be found in [3], where accelerometers are used to recognize groups of people walking together.

A crowd is more than just a sum of individuals and collective behavior results from a continuous interaction and mutual influence between each individual and those nearby. The literature presents a vast number of examples of such behaviors as exhibited by animals such as swarms of insects, flocks of birds, and schools of fish, and there is evidence of herding behavior in humans as well [4]. Such networks of influence are fundamental for the emergence of collective behavior, and are based on both the spatial relationship between the individuals and their social relationships.

In this article, we address the representation of the texture of a crowd through a *(dynamic) proximity graph*.[1] The proximity graph provides a

computational representation of a crowd over time, allows analysis of the crowd texture it represents, and provides a way to compute the effects of interventions into the crowd. Modeling relationships between individuals through a graph is not new. Social graphs represent social relationships between individuals through edges. Also, on-body sensors have been used to actually measure social graphs [2, 5]. Finally, for a few years various groups have been dedicated to gathering data on the mobility of people.

However, using a spatio-temporal graph to represent the texture of a crowd has not been done before, and we are not aware of any attempts to do so on the scale of (tens of) thousands of people. Besides its intended scale, the novelty of the proposed approach lies in the content of the proximity graph and the semantic interpretation of an edge. At a specific moment in time, an edge merely represents that two individuals were close to each other. Measurements over prolonged periods will reveal social groups (as we discuss in this article), spatial structures (lanes, clogging, etc.), but also the changes in the texture that result from targeted interventions (e.g., displaying announcements on a large public display). The main contribution of this article is introducing the concept of crowd textures and their representation by proximity graphs.

The remainder of this article is organized as follows. We connect the local spatio-temporal nature of crowds and crowd dynamics to the concept of crowd texture. We then introduce the proximity graph as a representation of crowd textures, along with analytic examples. We describe a system architecture for an instrument to extract a series of proximity graphs from a crowd, analyze it, and communicate feedback to the crowd. We present an analysis of the proximity graphs we collected during a real-world experiment through a wearable device. We conclude by discussing possible extensions to the presented work.

Crowds and Crowd Dynamics

A generally accepted definition of a crowd is that it is a sizable number of individuals gathered together at a specific location with a sufficient density distribution for a measurable amount of time and for a specific purpose. Moreover, the individuals in a crowd generally act in a coherent manner, sharing a social identity and common goals, interests, and behavior, in spite of coming together in a typically unfamiliar situation.

Crowd behavioral patterns emerge from individual human interactions, which typically have a strong local character: individuals in a crowd influence one another, and this influence is stronger between nearby individuals. In fact, several models of crowd dynamics — which give rise to large-scale emergent behavioral patterns — do take into account local interactions between individuals, as well as the related measures of crowd density. For example, the social force model and its extension for panicking pedestrians [6] considers physical forces between individuals (as well as forces between individuals and the environment). The forces depend on the dynamic spatial relationships among close individuals (more specifically, their distance). This model can be simulated to show most of the coordinated and uncoordinated behavioral patterns mentioned above.

From these continuous interactions, reflected in the crowd texture, behavioral patterns in crowd dynamics can clearly emerge. Therefore, by measuring the crowd texture, substantial information about the underlying crowd dynamics can be gathered. However, in order to be useful, this information needs to be put in the context in which the crowd exists. The contexts can be diverse, such as a football stadium, a train station, a music festival, or a scientific conference. A pattern such as clogging, for instance, may occur in all these diverse contexts as it depends on environmental constraints such as the existence of narrow passages. In a music festival or stadium, clogging may occur at the entrance or exit of the festival site or stadium. Finally, in a train station, clogging may occur at the entrance to the station hall or entrance to a train, or at the access to stairs/escalators. The same measured crowd texture will clearly mean different things in different situations, with corresponding different risk level assessments and different feedback strategies to intervene in the crowd.

Crowd behavioral patterns emerge from individual human interactions, which typically have a strongly local character: individuals in a crowd influence one another, and this influence is stronger between nearby individuals.

The Proximity Graph

A static proximity graph is a representation of the texture of a crowd at a specific moment in time. In its basic form, each vertex corresponds to an individual. Two vertices are joined by an edge if the two individuals they represent happen to be in physical proximity, within a chosen distance. Each edge represents only the Boolean relationship without capturing the actual distance. Time is generally discretized into small slots. If and only if two individuals were *detected* to be in each other's proximity during a time slot would their associated vertices be joined by an edge for that slot. A proximity graph is therefore *dynamic*; we speak of a proximity graph at time t as a (*proximity graph*) *snapshot* at t. Note that a proximity graph is naturally represented by a time series of proximity-graph snapshots. Although a proximity graph contains spatial information, it does not rely on absolute positioning data. It is the global representation of the texture of a crowd constructed from the local perspective of individuals within that crowd. A proximity graph incorporates information to support the modeling of crowd dynamics where the individual behavior is defined only on relative neighborhood information. Basic models of flocking behavior are of this kind, as they are controlled by the following simple rules [7]

1 Separation: Avoid crowding neighbors (short-range repulsion).
2 Alignment: Steer toward the general direction of neighbors.
3 Cohesion: Steer toward the general position of neighbors (short-range attraction).

Each of the three rules can be applied by a member of the crowd based only on local information (its neighbors' states), and no global view or absolute spatial reference points are necessary. This type of modeling has been successfully applied to human herding behavior as well [4]. The above set of rules has been extended in dif-

[1] *Our definition of proximity graph should not be confused with that of a relative neighborhood graph, although the two share some properties.*

A consequence of the relative nature of the proximity graph is that movement cannot be attributed to vertices. An edge "breaks" when at least one of the two individuals involved in the relationship moves away. As the result of these actions being the same, determining which one occurred is not possible.

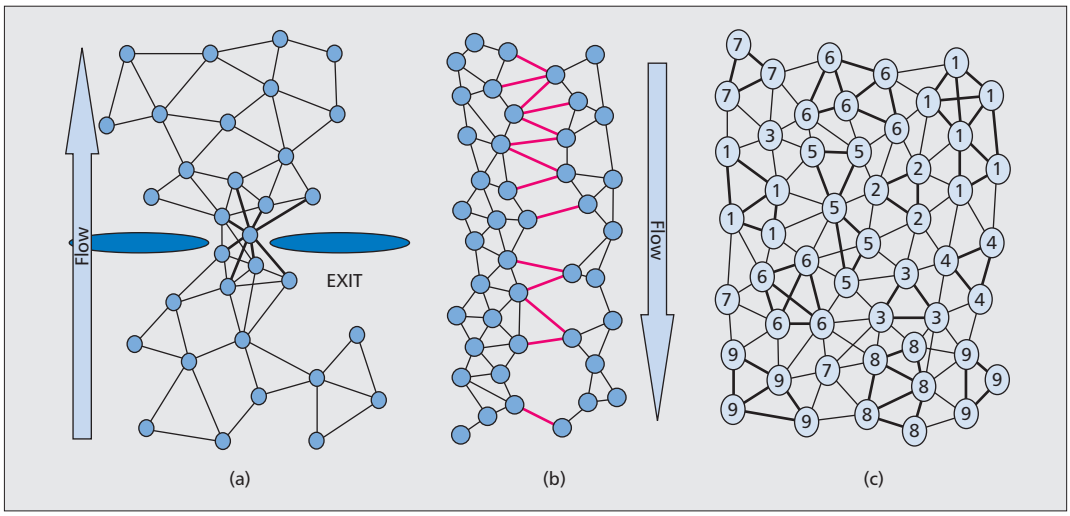

Figure 1. *Examples of analyses to perform on the proximity graph: a) detection of congestion; b) detection of flows; c) detection of social groups.*

ferent ways since its introduction to incorporate emotions, leadership, and so on.

Analyzing the Proximity Graphs

We now introduce a few examples of categories of analysis that could be performed on proximity graphs, which correspond to the contexts mentioned in the previous section.

Figure 1a presents a scenario where a crowd is exiting a stadium. As each individual approaches the exit, the local density increases given the physical bottleneck imposed by the gate. In the proximity graph, the local density is represented for each vertex by its degree, the number of incident edges (depicted in the figure for a particular vertex with thicker lines). Recognizing congestion requires measuring the gradient of the average density of a crowd as it approaches the exit.

Figure 1b presents a scenario at a train station where a pedestrian flow is forming on a platform. There are two groups of individuals: on the left there is a stationary group waiting for a train, on the right there is another group that has just stepped out of another train and is heading toward the exit. The detection of the pedestrian groups is based on the transience of edges. In a time interval, the transience of edges and their repetition determine the degree to which neighborhoods remain the same. Intra-group edges present relative stability. On the contrary, inter-group edges, depicted in red in the figure, are characterized by a short-lived nature. By filtering out these edges, it is possible to detect the two connected components representing the two groups.

Figure 1c presents a scenario at a music festival. This type of event is usually attended by groups of socially related individuals who tend to stick together during its course. Nonetheless, groups might occasionally split (e.g., to reach the bar or a restroom, or due to the density in front of the stage). In the figure, we picture a moment when a crowd stands in front of a stage. The edges represent the current proximity between the individuals, and the thickness represents the accumulated time spent close to each other over the whole period. Edges representing past proximity have not been drawn. In fact, they are not valid to describe the current crowd texture, but are used to compute the accumulated time. The detection of social groups requires preserving only the edges that present a long-lasting time interval when compared to the others. Each vertex is annotated with a group label to which it has been assigned. Note that vertices which are far apart can still belong to the same group as some of its members might have currently and temporarily split.

Awareness of Context

Up to this point, we have considered vertices corresponding only to individuals. However, the proximity graph can be extended to represent other types of objects as well, such as a door, an automatic teller machine (ATM), or a food stand. This type of extension allows for semantic and contextual interpretation of the observed behavior. In fact, different interpretations can be applied to the same crowd texture, depending on the context in which the observation took place. Figure 2 presents an example of how the same proximity graph can be interpreted differently. Knowing it was collected in front of an ATM, the left-most graph can be interpreted as a queue. Instead, the same graph can represent an orchestra, knowing that the context was a stage (in this case the right-most vertex in the graph corresponds to the conductor, while the other vertices correspond to the musicians). Location awareness allows for the disambiguation of the meaning a proximity graph can assume.

The concept of awareness can be generalized through annotations. An annotation to a vertex is a key-value pair that describes a specific property of the object represented by that vertex (e.g., gender, age, exit number, gate capacity, GPS coordinate, staff role). Annotations contribute to awareness as they allow the observed texture to be pictured more precisely and support informed decision making. For example, through an annotated graph it is possible to rec-

ognize at what specific exit a congestion has formed, if children are involved, or whether stewards are already around the area for support. Also, static vertices with GPS coordinate annotations can act as absolute positioning reference points when necessary.

A consequence of the relative nature of the proximity graph is that movement cannot be attributed to vertices. An edge "breaks" when at least one of the two individuals involved in the relationship moves away. As a result of these actions being the same, determining which one occurred is not possible. Going back to the platform example in Fig. 1b, recognizing the short-lived edges, depicted in red, allows for the detection of relative movement between the two groups. Again, it is not possible to determine for each group whether or not it is moving. Extending the graph with static vertices spread along the platform provides a solution to the problem. In fact, the vertices belonging to the moving groups would create short-lived edges with these static vertices allowing inference of movement.

The temporal information encoded in the proximity graph allows for more than the description of crowd dynamics. As proximity can be a by-product of social interaction, socially related individuals tend to spend more time close to each other than strangers. This principle is at the basis of the example depicted in Fig. 1c. For example, in [8] the authors were able to analyze the social ties within organizations by looking at low-frequency proximity information. In general, knowing how much time two or more individuals have spent close to each other, with the possible addition of contextual information about location, enables the inference of the type of social relation incurring between them.

Temporal information also enables the inference of consequentiality. Imagine the festival scenario depicted in Fig. 1c. Suppose we observe individual A conversing first with B and later with C. If, later on, we note A, B, and C conversing together, we could imagine that B and C were introduced by A. In the same way, we could predict that an individual who was observed in proximity to a counter, where tickets for drinks are sold, will eventually show up at the bar. Similarly, the temporal dimension of face-to-face interactions has also been investigated to study the spreading patterns of epidemic diseases in different types of social events, such as a conference or a museum [5].

An Instrument

We identify the following requirements for an instrument to measure and analyze the proximity graphs presented thus far:

1. It is composed of a device that is *wearable* by an individual.
2. It is *detectable*, within a chosen distance range to measure proximity.
3. The information measured by the devices is *extracted* and *collected* through an infrastructure.
4. The collected proximity graph is *analyzed* by a processing system looking for patterns.
5. *Feedback* is transmitted from the system to the individuals.

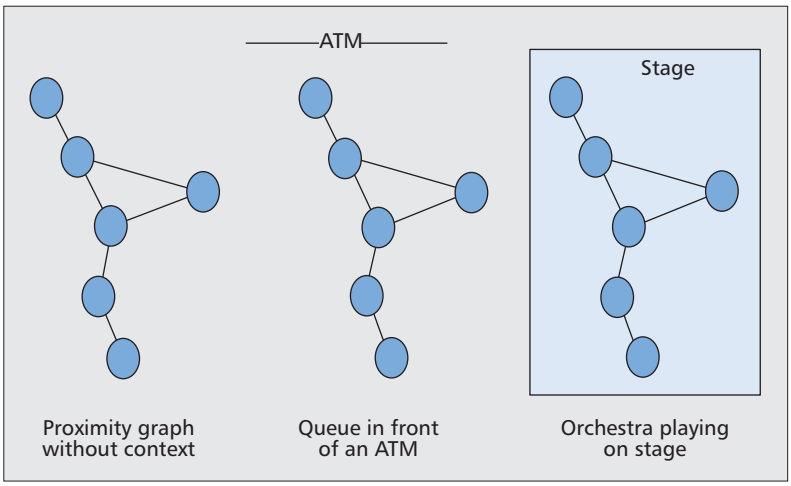

Figure 2. *Example of the dependency of the proximity graph on contextual information.*

One notable way to implement our sensing device is an application running on a mobile phone. Mobile phones are widely diffused across the population, and already ship with both high computing power (e.g. dual-core processors) and a broad range of sensors (accelerometers, microphones, cameras, etc.). On the other hand, with the decrease in price and size of wearable technology, another way of implementing our device is through a smart chip card, such as those currently used for public transport tickets or badges in workplaces (the same technology can easily be integrated into a festival bracelet as well). Figure 3 shows the device we currently use for our experiments with the proximity graph. We return to its description shortly.

Sensing Technologies

As far as detectability is concerned, multiple technologies have been investigated in the past. Infrared, ultrasound, and radio-frequency sensors, such as RFIDs and Bluetooth, have been utilized to track face-to-face interactions and co-location. The general approach to detection requires the assignment of a unique identifier to each device. The ID is periodically transmitted to nearby devices over the communication medium, and the reception of such a message constitutes a detection.

Various aspects influence the functionality of proximity detection and determine the vertices' neighborhoods. The range, direction, and angle of the transmission cone bias the type of interaction being recorded. For example, face-to-face interaction is tracked through a transmission range within about 2 and 4 m, a frontal direction, and a narrow cone of about 20° (e.g., via an infrared sensor). Conversely, co-location is measured through long-range omnidirectional transmission (e.g., via Bluetooth). The theory of Proxemics guides the choice of transmission range depending on the type of social behavior one wants to measure. Measuring a crowd texture poses a set of constraints on the detection strategy: omnidirectionality to maximize the recall of nearby devices, short-range transmission to detect only nearby devices, and high frequency to grasp instantaneous changes to the texture.

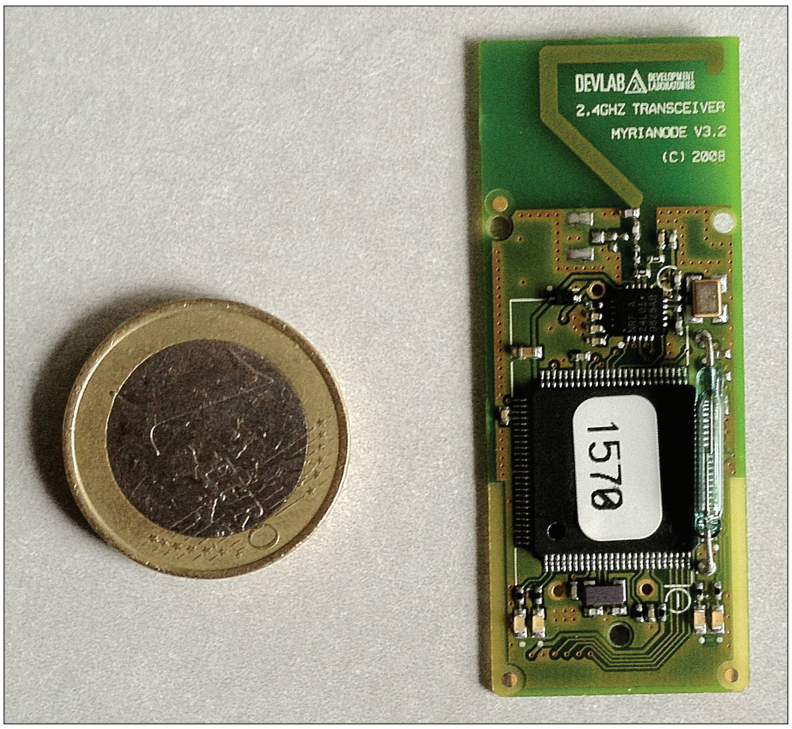

Figure 3. *Experimental device used for capturing the proximity graph*

System Architecture

As neighborhood information needs to be extracted and collected from the devices, they need to be connected to a network that allows them to reach a central repository. The mobility and high density that characterize a crowded environment make centralized networks, such as cellular networks and WiFi, unsuitable for this utilization. In contrast, decentralized ad hoc wireless networks provide the flexibility to design problem-specific protocols that guarantee higher scalability. Typically, such networks make use of special devices, usually called sinks, that receive data from on-body devices to subsequently store that data at a central repository. Sinks are spread around the event location to achieve high coverage. The on-body devices can reach the sinks by using a high transmission range or through dissemination protocols (e.g., gossiping and routing). The sinks bridge the ad hoc wireless network with the network where the central repository and processing systems are situated. The central repository and processing system can be deployed to one or multiple servers at the event location or in a cloud service.

We connect the devices through an ad hoc wireless network, as it matches the given constraints while enabling both transmission of detection messages and exchange of information between the devices. Also, it allows for communication with the external network for processing and feedback.

While data streams from the devices to the central repository, the global view of the proximity graph is constantly updated. Periodically, the processing system analyzes the proximity graph, examining the crowd texture for known patterns. Recognition of crowd dynamics is a classification problem that requires statistical analysis of the metrics of the graph, at either the vertex, group, or global level. The temporal information in the proximity graph is exploited by analyzing snapshots, defined by time intervals whose duration depends on the analysis being performed. Graph algorithms are potentially computationally expensive, introducing additional latency between the measurement and the recognition of patterns. This requires a system with sufficient computational power to analyze a proximity graph within the required time constraints.

Once the information is extracted from the proximity graph, it can be presented for feedback. The feedback can be sent to the crowd managers or directly to the crowd, through either the devices or fixed infrastructure (screens, speakers, etc.). While feedback can reach the control room and the infrastructure over the reliable network, devices can be reached via the ad hoc wireless network. In the same way information is extracted and collected from the devices, dissemination protocols allow feedback to reach specific individuals, groups, or the whole crowd. The intervention strategy defines the information, destinations, and the techniques used to give feedback to the crowd.

A Processing Chain

We present now a processing chain that incorporates all the components presented above into a loop. The processing loop is presented in Fig. 4a. Each individual wears a device with a proximity sensor capable of detecting the other devices within a chosen distance range. The devices share a communication medium that enables transfer of information. The loop starts with the measurement of the individual's neighborhood through the device's sensor. Once the neighborhoods are measured, the next step consists of the collection of the neighborhoods from the devices to a central repository to compose the global view of the proximity graph. Afterward, the proximity graph can be analyzed by a processing system to recognize crowd dynamics. Possibly, feedback is computed and sent to the managers or crowd. At this point, the new state of the crowd can be measured, and the loop can start again.

Figure 5 shows a possible future instantiation. People at a train station are assumed to have proximity sensors (e.g., embedded into their smartphones), allowing for the detection of a proximity graph. The analysis of the situation in the train and on the platform may be used to subsequently inform people where to board or leave the train.

Note that centralization is not strictly necessary to the instrument. As individual behavior depends on local context represented by the individuals and the objects in close proximity, the analysis algorithm can often be expressed based on a vertex-centric local view of the graph. Following this approach, nearby devices can exchange their neighborhoods right after the measurement step, and analyze their local view of the graph autonomously, without relying on a global view of the graph contained in the central repository. This autonomy decreases the interval between the moment the state of the crowd is measured and the moment feedback can be generated. This alternative is presented in Fig. 4b.

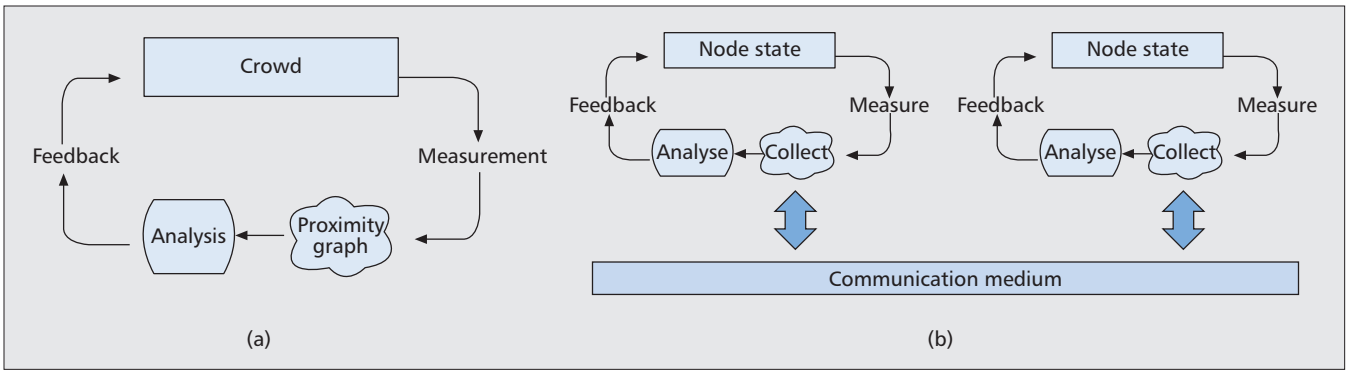

Figure 4. *Block diagrams for the processing chain: a) centralized processing chain; b) decentralized processing chain.*

However, a third possibility is available: a hybrid system where the devices locally aggregate and process proximity information, and only these aggregated views are later collected and processed centrally. This latter approach provides both decreased latency in feedback generation and central monitoring of the crowd. Moreover, it minimizes the overhead of data extraction from the devices, as only aggregated information is transmitted.

Real-World Experiment

To give a flavor of what can be achieved following our approach, in this section we describe an example application based on a real-world experiment we conducted at an information and communications technology (ICT) conference. The conference was divided into seven tracks, each focusing on a specific ICT topic: High-Performance Computing, Software Engineering, Security, and so on. Of the 250 individuals attending, 139 were wearing one of our devices as a name tag throughout a whole day. We asked each of the participants for their main track of interest, as a hint to community membership (note that the groups were not balanced; e.g., one had only four individuals participating in the experiment). Table 1 shows the distribution of the participants across the tracks. The device, depicted in Fig. 3, has an Atmel ATXMega128 CPU with 8 kbytes of RAM, 128 kbytes of flash memory, and a Nordic nRF24L01+ wireless radio. To communicate, the devices create an ad hoc wireless network through an energy-efficient medium access control (MAC) protocol designed for mobile social networks [9]. Through this network, every second each device transmits its ID to the devices nearby, within some 2–3 m distance, allowing for its detection.

The devices log detections on the on-board storage unit along with their timestamps. At the end of the event we downloaded these logs from the devices for offline analyses. The devices were also broadcasting a second type of transmission. Between two short-range transmissions, a long-range transmission was broadcast up to about 20 m. This second broadcast contained the list of IDs received by each device during the previous second, hence comprising only short-range detections. We captured these long-range transmissions through the sinks we installed in the main hall of the event location. We used this data to visualize at the event the evolution of the proximity graphs in real time. Long-range transmissions were not used to detect proximity.

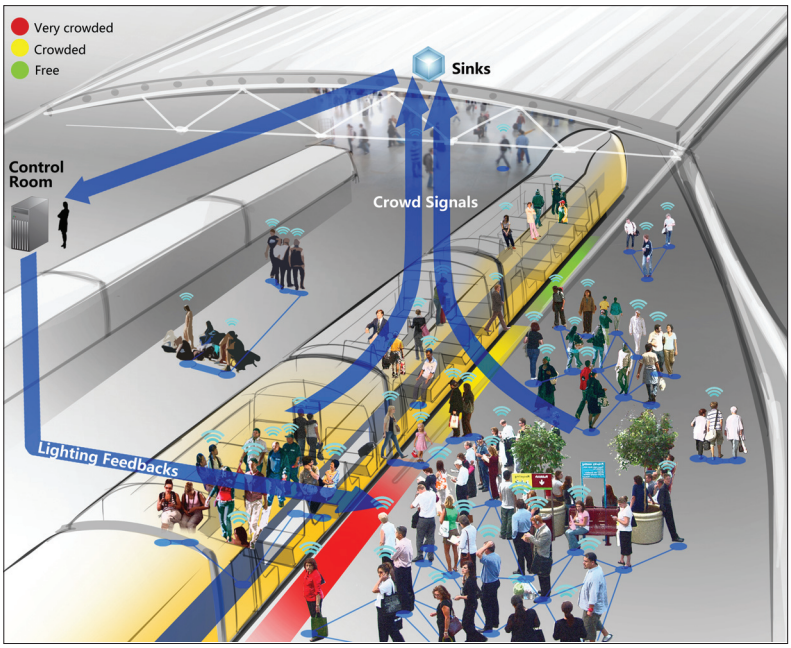

Figure 5. *How to sense and use the texture of a crowd at a train station. The density inside the train and on the platform is used to guide the individuals toward the less crowded coaches.*

Although we recorded proximity information during the whole conference, we concentrate here on the two hours between 12:00 and 14:00, when the poster session and the lunch break took place. During this time, all the participants gathered in the main hall. Our goal was to investigate to what extent during this time the participants stayed close to people with whom they shared interests, as indicated by their main track of interest. To perform this analysis, we aggregated the series of proximity-graph snapshots into a single undirected static graph, for which we decided to join two vertices by an edge if the two corresponding individuals had been in physical proximity for at least 600 seconds during the two hours. Each edge has a weight that accounts for the total number of seconds the two individuals have spent in physical proximity. On the largest connected component (LCC) of this graph, we ran a state-of-the-art community detection algorithm [10].

The mutual influence between individuals is an interesting aspect of social behavior, as it can be used to guide a crowd by targeting a subset of the individuals through feedback information about the current state. Although over the last few years we have started to better understand crowd dynamics, less is known about how to influence a crowd.

	Track 1	Track 2	Track 3	Track 4	Track 5	Track 6	Track 7
Participants	27	9	19	4	26	10	27
In LCC	15	7	13	3	20	4	22
Correct classifications	0.53	1.0	0.62	0.0	0.74	0.0	0.77

Table 1. *Sample statistics and results of the analysis. An additional 17 participants were part of the organization and were not labeled with one of the main tracks.*

The analysis we performed is analogous to the detection of social groups presented earlier. The community detection algorithm assigns vertices to communities trying to maximize modularity. Graphs with high modularity tend to have dense connections between vertices within communities and sparse connections between vertices across different communities. Intuitively, it groups together vertices that are interconnected and have spent a long time together. Figure 6 shows the results of the analysis. Vertices are colored according to the community to which they have been assigned by the algorithm and labeled according to the main topic of interest. The clustering tends to assign vertices with the same label to the same community, supporting our hypothesis and showing the validity of the data extracted through the instrument. Note that the algorithm does not make use of the noted information about interests, but only of the topology of the graph. One should not consider the main track of interest as ground truth. In fact, many of the individuals indicated their interest as one out of more possible ones. Moreover, as the participants came from a number of universities and departments, they tended to socialize also according to different criteria, for example, with people with whom they shared affiliation. Finally, the nature of poster sessions and banquets stimulate people to spend time in proximity to people belonging to different communities. We leave a deeper and more sophisticated analysis of the experiment for future work.

The unreliability of the wireless communication and the devices causes misdetections, meaning that detections are missed or erroneously added. Typical examples of these causes are collisions, interference of other sources of radio frequencies (e.g., WiFi spots), shielding of the human body, and data corruption due to faults in the device. For this reason, to extract the proximity graphs used in our analysis we preprocessed the logs with a density-based clustering algorithm. The algorithms exploits the "bursty" nature of the collected data to reconstruct part of the missed detections and filter out noise. Our simulations show that this filtering phase greatly increases the sensitivity of the instrument [11].

Discussion

The local nature of individual behavior in crowds also affects the analysis of crowd dynamics through other modalities. In [3], the authors propose a processing chain for the recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering. Subjects wear on-body sensors and move collectively. Activity patterns are then extracted from the sensors for individual behavior recognition (e.g., from accelerometer data). Afterward, a pairwise correlation of this information between each pair of individuals is computed, forming what they call a disparity matrix. Finally, this information is transformed into a graph by performing multidimensional scaling. In the obtained graph, each vertex corresponds to a subject. The neighboring nodes have similar behavioral characteristics and are thus more likely to participate in the same dynamics. By performing a graph clustering algorithm, they are able to predict that the subjects corresponding to the vertices belonging to the same cluster participated in the same group. By exploiting the information contained in the proximity graph, only the behavioral data between neighboring vertices could be compared, reducing the cost of an expensive step of the processing chain. The proximity graph is a representation of the crowd texture that can be used to either directly recognize crowd dynamics or support recognition through other modalities.

The mutual influence between individuals is an interesting aspect of social behavior, as it can be used to guide a crowd by targeting a subset of the individuals through feedback information about the current state. Although over the last few years we have started to better understand crowd dynamics, less is known about how to influence a crowd. The proximity graph is a representation of a crowd and allows interventions toward a desired behavior to be computed. Consider the following (admittedly still speculative) examples of simple interventions on crowds as presented previously in their scenarios.

The example in Fig. 1a could represent a bottleneck at the entrance of a stadium. This is a typical situation where people get pushed, and in the most dramatic conditions also trampled. This behavior often finds its origins in the absence of information. The individuals at the back of the crowd cannot see the high density at the entrance or the presence of congestion, and may start to push. A screen on top of the gate, visible to all the individuals, could depict with colors the density in the front.

The example in Fig. 1b could represent a platform in a train station. A common situation in such a scenario is that people getting out of the train tend to head toward the closest exit. Uneven usage of exits could be avoided by feeding back information about less crowded exits.

Finally, the example in Fig. 1c could represent groups of visitors to a festival. Such events

are usually visited by groups of friends. It is common that people lose contact with some members of their group. Once the groups are detected, each member of the same group can be guided toward the same exit so that they can find each other there.

These are just a few examples of the many possibilities that emerge once the texture of a crowd is captured in a proximity graph. Our approach has great potential to accelerate the emerging field of computational social science. In particular, the capability of sensing the crowd from within, without any requirement for location information or centralization, respects the privacy of the individuals in the crowd. Moreover, it allows timely insights about the state of the crowd to be computed, and communicate feedback to ensure safety and comfort.

Acknowledgments

This publication was supported by the Dutch national program COMMIT. The authors would also like to thank Matthew Dobson and Marco Cattani for their help with the deployment of the real-world experiment presented in this work.

References

[1] D. Lazer *et al.*, "Social Science: Computational Social Science," *Science*, vol. 323, no. 5915, Feb. 2009, pp. 721–23.

[2] T. Choudhury and A. Pentland, "Sensing and Modeling Human Networks using the Sociometer," *Proc. 7th IEEE Int'l. Symp. Wearable Computers*, IEEE Computer Society, 2003, p. 216.

[3] D. Roggen *et al.*, "Recognition of Crowd Behavior from Mobile Sensors with Pattern Analysis and Graph Clustering Methods," *Networks and Heterogeneous Media*, 2011.

[4] J. R. G. Dyer *et al.*, "Leadership, Consensus Decision Making and Collective Behaviour in Humans," *Philosophical Trans. Royal Society B: Biological Sciences*, vol. 364, no. 1518, Mar. 2009, pp. 781–89.

[5] L. Isella *et al.*, "What's in A Crowd? Analysis of Face-to-Face Behavioral Networks," *CoRR*, abs/1006.1260, 2010.

[6] D. Helbing and A. Johansson, "Pedestrian, Crowd and Evacuation Dynamics," R. A. Meyers, Ed., *Encyclopedia of Complexity and Systems Science*, Springer, 2009, pp. 6476–95.

[7] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," *SIGGRAPH Comp. Graph.*, vol. 21, no. 4, Aug. 1987, pp. 25–34.

[8] D. Olguín et al., "Sensible Organizations: Technology and Methodology for Automatically Measuring Organizational Behavior," *IEEE Trans. Systems, Man, and Cybernetics — Part B: CY-Bernetics*, 2009, pp. 43–55.

[9] M. Dobson, S. Voulgaris, and M. van Steen, "Merging Ultra-Low Duty Cycle Networks," *Proc. 2011 IEEE/IFIP 41st Int'l. Conf. Dependable Systems & Networks*, Washington, DC, 2011, IEEE Computer Society, pp. 538–49.

[10] J. Reichardt and S. Bornholdt, "Statistical Mechanics of Community Detection," *Phys Rev E 9 Stat Nonlin Soft Matter Phys*, 74(1 Pt 2), July 2006.

[11] C. Martella *et al.*, "From Proximity Sensing to Spatio-Temporal Social Graphs," Technical Report EWiDS-WP3-130410-01, VU University Amsterdam, Apr. 2013.

Biographies

Claudio Martella (claudio.martella@vu.nl) is a Ph.D. candidate in the Large-Scale Distributed Systems group of VU University Amsterdam (VUA). His research is focused on complex networks, graph processing, and large-scale distributed systems. He is currently working on modeling collective behavior through spatio-temporal graphs, by means of wearable devices and ad hoc wireless sensor networks. The main use case scenario is crowd management.

Aart van Halteren is a senior scientist with Philips Research and an associate professor at VUA. He explores the boundaries of large-scale distributed techno-social systems for sensing, modeling, and influencing human behavior. His work finds application in the consumer space in the area of healthy eating and adherence to a personalized physical activity program. In the healthcare space he delivers solutions for therapy adherence of chronic patients.

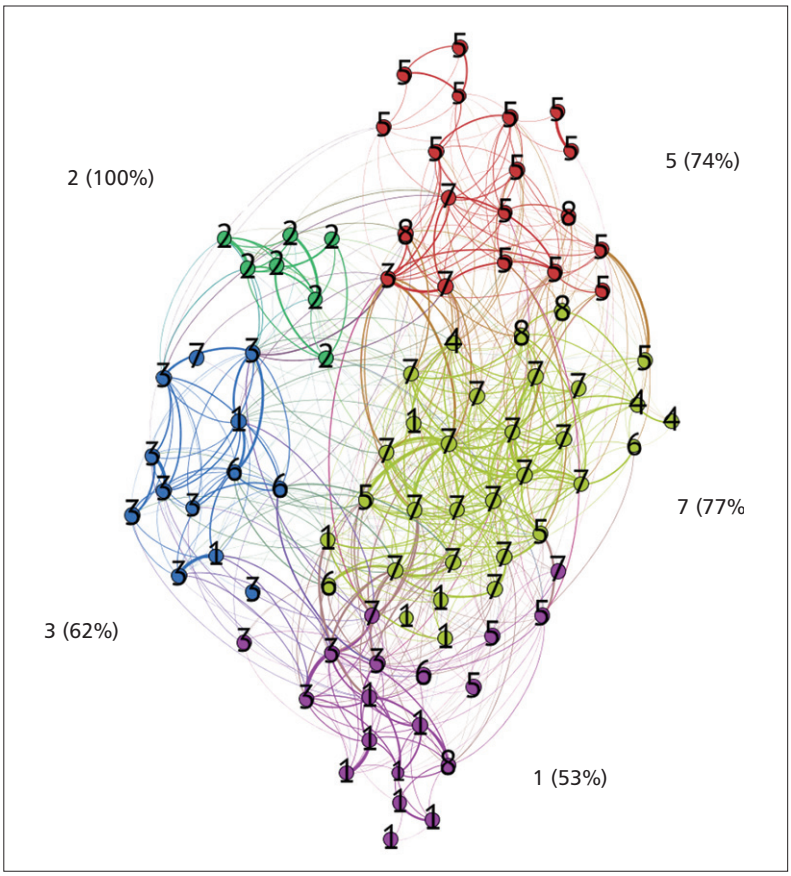

Figure 6. *Community detection at an ICT conference. Colors indicate the detected communities, labels indicate the main topic of interest of the individuals, and edge thickness indicates the total amount of time the individuals have spent in physical proximity. The vertices have been positioned through a spring embedding algorithm. The algorithm takes into account only the topology of the graph; hence, the layout is not related to location information within the main hall. We have labeled each community with the label that appears most frequently in it, and the value expresses the percentage of vertices with the label appearing in their community.*

Maarten van Steen is a full professor at VUA and currently chair of the Department of Computer Sciences. He has taught modules and courses covering various areas of computer systems and complex networks to academics and professionals. He has co-authored two textbooks on networked computer systems. In his recent research, he has been exploring gossip-based solutions to achieve decentralized autonomous systems, partly focusing on very large wireless sensor networks and pervasive computing.

Claudine Conrado obtained her Ph.D. degree in 1993 in theoretical physics from the Niels Bohr Institute, Denmark. After working as a research associate at Imperial College London, United Kingdom, she joined Philips Research, The Netherlands, to work on adaptive algorithms for home systems and privacy enhancing technologies. Currently she works at Thales Research and Technology, The Netherlands, where her areas of interest include decision making in real-world applications and distributed reasoning.

Jie Li is a Ph.D. candidate at the Human Information Communication Design Section of Delft University of Technology. Her research is focused on designing for crowd well being. She is currently working in the COMMIT EWiDS project, investigating the relation between end users (e.g., crowd managers and crowd members) and the design of extreme wireless distributed systems.