Self-Organized Togetherness in a Crowd Group

Kashif Zia
Institut für Pervasive Computing
Johannes Kepler University,
Linz, Austria
+ 43 732 2468 5351
kashif.zia@researchstudio.at

Sajjad A. Madani
CS Department, COMSATS Institute
of Information Technology,
University Road, Abbotabad, Pakistan
+ 92 343 9630077
madani@ciit.net.pk

Alois Ferscha
Institut für Pervasive Computing
Johannes Kepler University,
Linz, Austria
+ 43 732 2468 8555
ferscha@soft.uni-linz.ac.at

ABSTRACT
Self-organization is a computing paradigm in which participating entities proceed to execute a global goal strictly based on local information. In population dynamics, the sense of togetherness (due to social bindings or a common confinement) experienced by a group of individuals (i.e. 'crowd group') is an interesting phenomenon to explore in the context of self-organization. Given a mechanism supporting spatial awareness, many settings require individuals belonging to a group, not only, to stay together (togetherness), but also to account for personal goals (dispersion). Self-organization can help individuals within such a group to stay together and having dispersed at the same time, togetherness being the primary requirement. In this paper, we discussed the parameters defining togetherness and dispersion within a spatially aware crowd group. In this context, the factors affecting the interplay between togetherness and dispersion were examined and maximum tolerable limits of dispersion were tailored in diverse settings of collaboration range, number of individuals willing to diverge and density of a crowd group. Simulation results provide insight into the interplay between these parameters, hence resolving operational dependencies.

Keywords
Self-organization, Space awareness, Connectivity vs. Coverage, Crowd Dynamics.

1. INTRODUCTION
Given that each individual in a crowd group is aware of required spatial information, self-organizing such a group within a large population presents many interesting application scenarios. One of these application scenarios is the notion of 'togetherness vs. dispersion' within a crowd group. For example, in a dense urban pedestrian environment, individuals part of one of the crowd group and heading towards a common destination, are required not only to stay together but also to avoid collisions with nearby pedestrians [1]. Self-organization can play an important role in achieving a balance between these essentially conflicting requirements.

Most crowd simulation systems rely on human behavior studies based on sense of sight and hearing for modeling population dynamics. Although this is a natural choice in case of human activities under 'normal' circumstances, it might reveal as a bold presumption in a really dense population due to threat of overseeing or overhearing in situations when human senses are exposed to an overload of visual and auditory stimulation. In such an environment, each individual in the target crowd group must be equipped with a digital device to sense the required spatial information. The device should also have a demanding interface to attract attention of the individual. We have already developed such a device in the form of LifeBelt [2]. LifeBelt can generate a variation of vibro-tactile stimuli (such as intensity, duration, frequency) to indicate the direction to move, speed of motion and a must-stop alert. The notifications generated are based on the relative spatial relations (distance and orientation) of all neighbors within desired communication range.

The basic purpose of this paper is to model the self-organizing capabilities of a crowd group within a dense population. The study is targeted to resolve the interplay between parameters defining togetherness and dispersion within a spatially aware crowd group. Considering a crowd group which has to be 'together' all the way to the destination, we concentrated on maximum tolerable limits of dispersion in diverse settings of collaboration range, number of individuals willing to diverge and density of individuals within the crowd group. It is important to note that the strange notion of a 'crowd group' is necessary to distinguish individuals forming a group within a crowd from a crowd (i.e. a population) itself. Since we deliberately avoided interaction of individuals with the individuals not part of target crowd group, for the rest of the paper, any reference to 'crowd' should be considered as a reference to 'crowd group'. The rest of the paper is arranged as following. In Section 2 we discussed the sensing capabilities of the digital device (LifeBelt). Section 3 after explaining the concepts of space based togetherness and self-organization, states the problem in hand and a solution outline. We simulated the scheme proposed in section 3 in Section 4 and analyzed the results. Section 5 discusses related work whereas section 6 concludes the paper.

2. SPATIALLY AWARE DEVICES
Given that a digital device (LifeBelt) is the only mean to describe an individual in a crowd, for the purpose of this paper, we have not emphasized on vibro-tactile notification aspects of the LifeBelt. Concentrating only on spatial sensing capabilities and transmission range, such a device is used for three important dynamics: (i) capability, (ii) social competence range, and (iii)

FIT'09, December 16–18, 2009, CIIT, Abbottabad, Pakistan.

interest delegation capability. By capability, we mean sensing and actuating capabilities of the device. Sensing capabilities are realized by sensors embedded into the device, particularly related with spatial awareness. Abstraction of space is possible through position, orientation and speed sensing [3]. As a proof of concept, we have only considered orientation sensing (direction of motion in this case) of one-hop neighborhood as minimum possible requirement for claiming a device to be spatially aware. Considering position and speed sensing of peer in the neighborhood would definitely improve the self-organized behavior. By social competence range, we mean the coordination (communication) range of the device. We consider proximity coordination (data sharing of one-hop neighborhood) as the minimum requirement to realize any useful application. By interest delegation capability, we mean the extant of the device to sacrifice its own goal to realize community goal. This factor depends on application requirements.

3. SELF ORGANIZING TOGETHERNESS

As indicated by authors in [4], self-organization is used in conjunction with related Self-X capabilities, like Self-healing, Self-configuration, Self-management, Self-optimization and Adaptation. According to the definition in [4], self-organization stands for *a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of a system. Moreover, the rules specifying interactions among the systems' components are executed using only local information, without reference to the global pattern.*

Self-organization, by virtue of its definition, advocates a case of 'togetherness' at global level. The global conditions justifying the notion of togetherness emerges with the passage of time with the help of decisions made strictly at local level. Togetherness of a goal of a crowd can be applied in many application domains. But scenarios related with spatial awareness are most challenging. In addition to the challenge, spatial awareness is the foundation on which many other application categories are built. Further, it suits to sensing (actuating), social competence and interest delegation capabilities of LifeBelt indicated in section 2. Fig. 1 indicates some of many possible spatial arrangements related to dynamics of moving objects (individuals in case of a crowd).

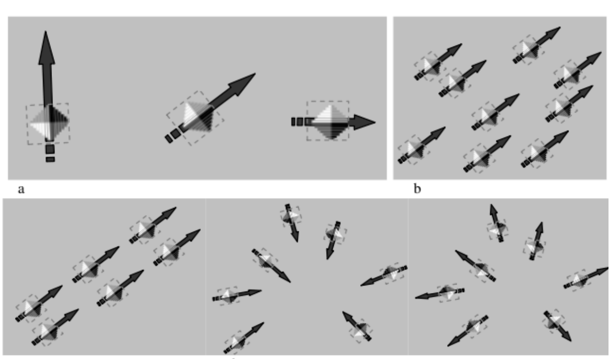

Figure. 1: Examples of Spatial Arrangements. (a) Object representation, (b-e) Possible arrangements of objects.

Each object shown in Fig. 2 (b-e) is based on the concept of an entity represented by its location, orientation (direction of motion) and speed of motion. In Fig. 2 (a), the location of the entity is represented by the diamond shaped center. The orientation is represented by the direction of arrow head passing through the entity. And speed of motion is represented by the length of the arrow. Although, as discussed in section 2, we did not consider speed and location of an object for the purpose of this paper, still, the entity representing these concepts is essential for clarity purpose. In Fig. 2 (b-e), we have drawn different spatial arrangements of objects based on orientation. All varieties consider same objects' speed. The location of an object is irrelevant (though it plays its role in calculating one-hop neighbor and next position of a moving object, but it does not have any role in decision making). We name these arrangements as: Oriented crowd (b), Line up crowd (c), Converging crowd (d), and Diverging crowd (e).

The diverging example (the focus of this paper) reminds us a famous problem of 'coverage vs. connectivity' mainly discussed in WSN community [5]. Keeping a network connected while allowing nodes to extend their coverage is a problem still in search of answers. Mapping this problem to real life applications provide it with many specific dimensions. Here we are investigating applications of this problem on behavior of crowds [6]. A crowd normally has a center of activity which is the densest part of it. Moving out towards peripherals, the crowd becomes progressively sparse and volatile, courtesy to less attentive audience. Application categories ranging from a museum tour and outdoor activity trips of kindergarten to disaster management require the crowd to stick together to prevent the danger of loss of crowd members. The crowd management strategy in this case should provide a mean to cope with the erratic behavior at the crowd periphery. It means that the volatile section of the crowd needs to be monitored. Functionally it means that we need to extend the coverage of the crowd as much as possible, still keeping it connected with the center of gravity. Using only orientation sensors and one hop neighbors as collaboration range, the interest delegation capability of an object allows the object to disperse as long as connectivity constraint allows it.

3.1 Problem Statement

The problem states that we want to keep the objects (individuals) in a crowd connected while ensuring maximum coverage. The intended algorithm should be truly distributed only relying on information locally available. After an object is disconnected from the crowd, the object should know about the state of disconnection and an optimal directional alert should be generated to re-connect the object with the crowd.

Before explaining the solution to the problem, we discuss two varieties of a crowd. To differentiate these two varieties, we take two real life examples. The excursion (particularly outdoor) tours of children in kindergarten require the teacher to keep children together. In a playground (park), children are left alone to explore around. It is assumed that a substantial number of kids would always stick to their teacher due to their young age nature, whereas, the others would wander around (volatile entities). This scenario defines a crowd with a 'static core'. A static core is the activity center of a crowd which is not moving. It is defined by a static group of entities having closely located neighbors. Contrary to the static, a crowd can have a core (closely related neighborhood) but with moving objects. A 'dynamic core' is essentially not an erratic arrangement of entities in which an entity moves in whatever direction it likes. The movement of the core is visualized as a collaborative effort keeping the closely related neighborhood intact. Still in the case of a dynamic core there can be outliers which deliberately deny moving with the core (volatile entities). An example of this kind of scenario is escape panic in crowded situations [7]. In such a situation, the escape route is the route followed by a substantial number of entities in the crowd. If for some reasons (local hurdles, for example), individual entities or small group of entities are not able to follow the route, it would be disastrous, if a situation of disconnection with the center of activity emerges. This unfortunate situation would deny the victims from possibility of a safe escape, after they get through the local hurdles they were facing. Irrespective of the type of the core, a 'volatile' object is the object which is willing to move away from the center of activity of the crowd.

In Fig. 2, we draw a scenario based on static core. Entities without directional arrows are not moving and are part of static core. The entities with arrows and a lighter shade diamond center are volatile. After acquiring their direction of motion (indicated by curved arrows with broken lines), the volatile entities would face a situation in which these would nearly be disconnected with the core. The volatile entities need to realize this situation and immediately invert their orientation as indicated by U-turn arrow.

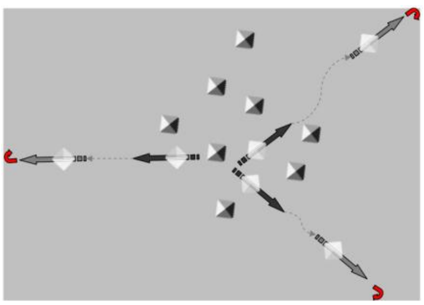

Figure. 2: Connectivity vs. Dispersion: A scenario based on static core.

In Fig. 3, we draw a scenario based on dynamic core. In Fig. 3 (a), all entities have directional arrows. The entities with lighter shade diamond center are volatile, while other are part of dynamic core moving in the same direction. Fig. 3 (b) shows the volatile entities nearly being disconnected with the core, after acquiring their direction of motion (indicated by curved arrows with broken lines). The volatile entities need to realize this situation and immediately acquire the orientation compatible with the orientation of the dynamic core.

3.2 Proposed Scheme

The proposed algorithm is based on Connection Index (CI), a measure of (count of) connections of each object in crowd. We will keep updating this index on each time stamp, thus justifying the requirements for mobile core. The intuition says that the value of CI depicts the nearness of an object towards the center of activity. We consider following initial assumptions:

- Constant speed of volatile objects along with core objects (in case of moving core)
- Modification of orientation in a single time stamp instead of modifying it iteratively

If an object has a direct connection with an object with greater CI than itself, it can be considered as a “safe” object. Otherwise it would invert its direction of motion in case of static core. In case of dynamic core, it would acquire a direction of motion equal to direction of motion of the core.

This seemingly simple strategy promises to be very robust. In fact, the dynamics generated by the interplay of parameters defining togetherness and dispersion require a keen study, as we have done it in next section.

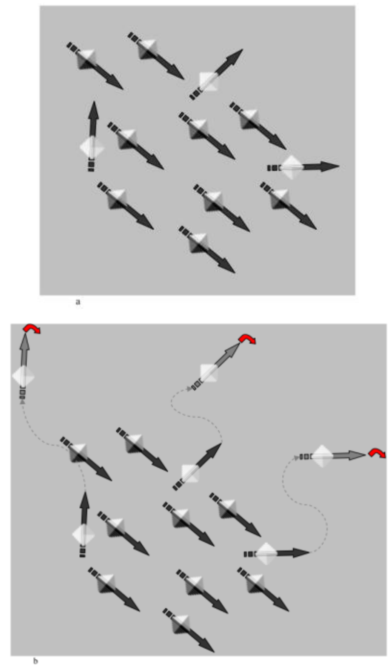

Figure. 3: Connectivity vs. Dispersion: A scenario based on dynamic core. (a) Initial arrangements of objects. (b) Arrangements when objects are nearly disconnected from the dynamic core.

4. SIMULATION

We performed separate simulation for static and dynamic core. Objects were randomly placed within a region of size 100 * 100. Having mapped onto the pixel information, the coordinates of the object and bounding region represent an arbitrary unit. In addition to its position of placement, each object is represented by following parameters:

- *Zone of Influence (ZoI):* Each object has a sensing range. The sensing range of an object overlapping with another object places sensed object in ZoI of the sensing object. This zone can be of any shape. For simplicity we consider a 2D circular region (classical sensing range) as the shape being considered. As ZoI expands, the neighborhood an object can interact with, expands. Possible values of ZoI considered for the simulation are 10 and 20 (arbitrary units of length based on pixel coordinates). Fig. 4 shows what it means to have ZoI equal to 10 and 20. The grid represents a total area of 100*100, with each square having an area of 10*10. Assuming a uniform distribution of 100 objects, we consider center of each of smaller square as the object presence. The square with hatched lines represents object of interest. The gray squares around the object of interest represent objects in ZoI of the said object, if ZoI is equal to 10. Gray plus black squares represent the objects in ZoI of the said object, if ZoI is equal to 20. The figure shows that for practical purposes, we do not need to consider ZoI greater than 20.

- *Orientation:* Orientation of an object is direction of motion of the object. Orientation does not have any significance for the objects which are not moving. But an object which is 'volatile' would only move in the specified orientation with a certain speed. The allocation of orientation to an object is randomly chosen from a set of possible orientations of 0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, and 315^o. A change in orientation of an object is performed in a single time stamp instead of modifying it step-by-step.
- *Speed:* It is speed of motion of an object in a specified orientation. We consider a constant speed of volatile objects along with core objects (in case of moving core).

In addition to bounding region which remains static throughout the simulation, following global parameters are considered:

- *Number of objects (N_{obj}):* Simulation was conducted for each of 25, 50 and 100 objects.
- *Percentage of volatile objects (PV_{obj}):* The percentage of objects moving away from the core. V_{obj} is related number of volatile objects with reference to N_{obj} . Percentage of volatile objects considered is 10%, 20%, 35%, 50%, and 75%.

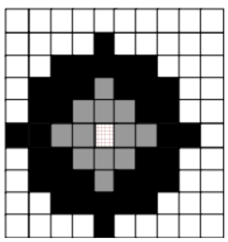

Figure. 4: The concept of ZoI.

We performed separate simulation for static and dynamic core. We calculated average (averaged 10 readings to neglect outliers) dispersion for ZoI * N_{obj} * PV_{obj} tuple, related to each possible value mentioned above.

4.1 Propose Algorithm

Following pseudo code describes overview of the algorithm at atomic level (for unique values of ZoI, N_{obj} and Pv_{obj}). Algorithm is applicable to static and dynamic cases, both.

```
For each object (o) in Nobj
  if (!(mobile))
  // whether run for mobile or static core
    if (!(is_safe(o))
      if (o.orientation_shift == -1)
      // change in orientation Never performed
        if (o.dom < 180)
        // direction of motion (orientation) of object
         o.orientation_shift = o.dom + 180;
        else
          o.orientation_shift=Math.Abs((o.dom+180)-360);
      else
        if (o.orientation_shift < 180)
          o.orientation_shift=o.orientation_shift + 180;
        else
          o.orientation_shift=
                 Math.Abs((o.orientation_shift+180)-360);
    else
      o.orientation_shift = o.dom;
  else
    if (!(is_safe(o))
      o.orientation_shift = 315;
    else
      o.orientation_shift = o.dom;
```

```
bool is_safe (Object o)
  bool found = false;
  Array neigh_arr = o.neighbors;
  if (neigh_arr.Count > 0)
    For each neighbor (n) in neigh_arr
      if(n.neighbors.Count>o.neighbors.Count)
        found = true;
  return found;
```

The algorithm relies only on proximity sensing (sensing objects within specified ZoI), without any information exchange between peers. In static core scenario, change is orientation is equal to the inversion of the orientation, whereas, in dynamic core scenario, change in orientation means acquiring orientation equal to orientation of the core. It is important to mention that no separate mechanism is implemented to inform the volatile objects about orientation of the mobile core. It can be implemented by making a distinction between volatile objects and objects which are not volatile, or by introduction of a 'different' object which knows that it is part of a moving core (and propagating orientation information to all objects). But both strategies deny the basic requirements of objects having the same functionality and one-hop neighborhood sensing as only mechanism of coordination. Our investigations suggest the solution to coordination problem (using proximity sensing only) is very challenging and we set it aside for future research. Instead, we use a global value of mobile core orientation (315^o) accessible to all volatile objects.

In static case, objects having experienced a change in orientation start a flip-flop behavior at the boundary of the core. It is due to the fact that after change in orientation for the first time, the motion of the object fetches it from an unsafe zone to safe zone. As soon as object finds itself in a safe zone, it flips again justifying its intrinsic desire to follow the orientation of its liking. Similarly in the dynamic case, objects having experienced a change in orientation start to move along the periphery of the moving core. In both cases, the volatile objects stabilize at the periphery of the core. But stabilization can prove to be a multi-step procedure, if an object already in flip-flop situation, encounters a new moving object which changes its CI. As a result, the object may perform few more moves in desired orientation before it reaches to a stabilized condition.

We ensured that volatile objects reach to the stabilized condition even if it is multi-staged. Then we calculated the 'dispersion' of the volatile object which we can define now. *Dispersion is the distance traveled by a volatile object after reaching at stabilization.* It is calculated by finding the difference between initial and final position of the object. In dynamic core case, the difference taken is between the expected position of the object (with reference to the dynamic core) and the position after stabilization (final position as before). Fig. 5 and Fig. 6 show the results of static and dynamic case respectively.

4.2 Analysis

The x-axis of both the graphs (Fig. 5 and Fig. 6) represents the % age of volatile objects, whereas the y-axis represents the average dispersion (summed for all volatile objects and averaged for 10 readings) experienced by the objects. In static case (Fig. 5), we show results of ZoI equal to 10 and 20, whereas, in dynamic case (Fig. 6), we show results of ZoI equal to 10 only. We will discuss about reason for this difference shortly.

In static case, the overall average dispersion at ZoI equal to 20 is higher than ZoI equal to 10. In addition, average dispersion of greater number of objects is higher than less number of objects.

General pattern of average dispersion from lesser to greater % age of volatile objects is ascending. Similar reasoning can be applied to the dynamic core case. On average, the average dispersion in dynamic case is around 75% more than that of static core case, in similar settings.

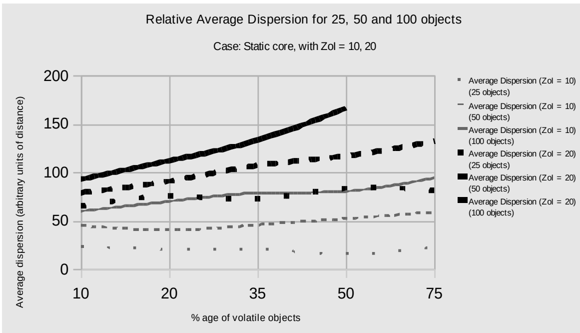

Figure. 5: Relative Average Dispersion -static case

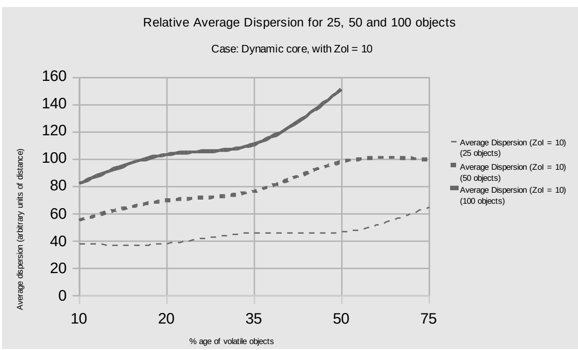

Figure. 6: Relative Average Dispersion -dynamic case

As discussed in last paragraph of section 3, the strategy adopted for maximizing the dispersion while keeping the crowd connected, promises to be very robust. But, during simulation, we encountered cases when dispersion was infinite due to a local maximum. In this case, sub-crowd within a crowd emerged due to local maximum of CI. As we investigated, the local maximum of CI emerged in the cases shown in Table 1.

Table 1. Cases of disconnection of crowd caused due to local maximum of CI.

Case	Core Type	ZoI	N_{obj}	PV_{obj}	Disconnection (%)
Case 1	Static	10	100	75	20
Case 2	Static	20	50	75	10
Case 3	Static	20	100	50	30
Case 4	Static	20	100	75	70
Case 5	Dynamic	10	50	50	20
Case 6	Dynamic	10	50	75	20
Case 7	Dynamic	10	100	35	40
Case 8	Dynamic	10	100	50	40
Case 9	Dynamic	10	100	75	60

We considered it unrealistic to plot case 4 and case 9 when undesirable phenomenon (frequent appearance of disconnections) exceeds 50%. In dynamic core case, when ZoI is equal to 20, the disconnections are so frequent that we opted not to show the results all together.

The simulation results presented are helpful to consider an application scenario based on collaboration range, density of objects and possible % age of volatile objects to build application expectations accordingly. For example in static case, the percentage of disconnections increases with increase in ZoI (i.e. from 10 to 20) from 0% to 30% (compare case 3 indicating 30% in Table 1 with non-existence of such a tuple in the table indicating a 0% disconnection), for same number of individuals and volatility. For the same set of observation, the dispersion increases from 80 to 165 (see Fig. 5) essentially 8 times as that of ZoI in both the cases.

5. RELATED WORK

In a wider context, spatial self-organization has been a subject of interest in agent technology [8] and robotics [9]. Research in AI and cognitive science has also focussed on flocking behavior [10] and swarm intelligence [11]. In addition to traditional distributed computing approaches of self-organization [12], efforts of mimicking the natural phenomena are also being investigated [11]. In [13], authors have presented a review of self-organization applications in computer science. These include applications related with Middleware (Grid computing, Coordination systems, Adaptive systems and Pervasive systems), Information systems (database organization and retail store), Robotic systems (mobile coordination and self-assembly) and Networks (MANET, WSN and Amorphous computing). Spatial (orientation in this case) self-organization falls in the category of coordination systems (particularly large-scale multi-agent systems) and self-assembly in robotics. Both domains have their own set of opportunities and challenges; one concentrating on logical distributed coordination mechanisms whereas other focusing on physical goal achievement. In a crowd simulation study relying on individuals equipped with light-weight helping device, we neither have liberty to incorporate software stack of high value (incorporating storage, energy awareness, intelligent coverage etc.), nor we have numerous sensing capabilities and related collaborative mechanisms to self-organize. In fact, the device is only provided with neighborhood sensing mechanism and extrinsic orientation sensing module. Using local measures of proximity sensing, the crowd self-organizes on-demand without any pre- (post-) qualification.

Spatial awareness using minimal object capabilities has not been an area of much research interest. The closest we get is a mechanism presented in [14] which provides fine-grained relative position information to co-located devices on the basis of peer-to-peer sensing, thus overcoming dependence on any external infrastructure. But it depends on range and angle-of-arrival measurements using ultrasonic signals between the nodes. In [15], authors propose Cooperative Artifacts that are able to cooperatively assess their situation in the world, without need for supporting infrastructure in the environment. The main contribution of this effort is distributed cooperation between peer nodes. It does not address spatial awareness explicitly and in comprehensive details.

An area closely related with our research is spatial formation control in robotics and WSN. In [16], authors present a new formation control framework for small scale, distributed mobile robot teams that can adjust their formation to adapt to a variety of circumstances. Specifically, a neighbor-referenced decentralized approach is proposed to generate and keep geometric shapes. In [17], spatial data correlations are exploited to group sensor nodes into clusters of high data aggregation efficiency. The problem of selecting the set of cluster heads is defined as the weighted connected dominating set problem. Then centralized and distributed algorithms are developed to select the cluster heads. In [18], authors present a distributed Topology Adaptive Spatial Clustering algorithm that partitions the network into a set of logically isotropic, non-overlapping clusters without prior knowledge of the number of clusters, cluster size and node coordinates. This is achieved by devising a set of weights that encode distance measurements, connectivity and density information within the locality of each node. The derived weights form the terrain for holding a coordinated leader election in which each node selects the node closer to the center of mass of its neighborhood to become its leader. The clustering algorithm also employs a dynamic density reachability criterion that groups nodes according to their neighborhood density properties. And in [19], authors propose a decentralized flocking algorithm, which can avoid collision between a robot and its neighbors and the collision between robots and obstacles when there are obstacles in the environment. In all these efforts the control on spatial formations is achieved through complex collaboration mechanisms. In our case, we do not need explicit formation control on the crowd. Instead the requirement is to ensure a sense of belonging to a crowd.

The main motivation of the work comes from [5], in which authors argue that maximizing coverage as well as maintaining network connectivity using the resource constrained node is a non-trivial problem. We applied this problem to generate useful application scenarios in which connectivity with the crowd and dispersion proved to be contradicting requirements.

6. CONCLUSION

Self-organization of orientation in a crowd can be achieved at global level with a local level coordination mechanism only. In this paper, we targeted conflicting requirements of togetherness and dispersion in the case of diverging crowd. Maximum tolerable limits of dispersion were identified in diverse settings of collaboration range, number of objects willing to diverge and density of objects within a crowd. The simulation results presented are helpful to consider an application scenario based on these parameters and to build expectations accordingly.

7. ACKNOWLEDGMENTS

This work is supported under the FP7 ICT Future Enabling Technologies program of the European Commission under grant agreement No 231288 (SOCIONICAL).

8. REFERENCES

[1] Loscos C., Marchal D., Meyer A.: Intuitive crowd behaviour in dense urban environments using local laws. In Theory and Practice of Computer Graphics, IEEE, 2003, pp. 122–129.

[2] A. Ferscha, K. Zia, "LifeBelt: Silent Directional Guidance for Crowd Evacuation", To appear in: 13th International Symposium on Wearable Computers (ISWC '09),

[3] Zia K., Ferscha A.: Spatial Self-Organization in Networks of Things. In proceedings of 3rd International Workshop on Self-Organizing Systems 2008.

[4] Dressler F.: Self-Organization in Ad Hoc Networks: Overview and Classification. In Ad Hoc Networks (2006).

[5] Ghosha A., Sajal K.: Coverage and Connectivity issues in wireless sensor networks: a survey. In Pervasive and Mobile Computing, Elsevier, June 2008.

[6] Lerner, A., Chrysanthou, Y., and Dani, L. 2007. Crowds by example. Computer Graphics Forum 26, 3, 655-664.

[7] Helbing D., et al.: Simulating Dynamical Features of Escape Panic. In Nature, 2000.

[8] Gervasi V., Prencipe G.: Flocking by a set of autonomous mobile robots. Technical report TR-01-21, University of Pisa (2001).

[9] Mataric M., Zordan V., Williamson M.: Making complex articulated agents dance: an analysis of control methods drawn from robotics, animation and biology. In: Autonomous Agents and Multi-Agent Systems 2(1) (1999) 23-44.

[10] Reynolds C.W.: Flocks, herds and schools: a distributed behavioral model. In: Computer Graphics 21(4) (1987) 25-34.

[11] Mondada F., Guignard A., Bonani M., Floreano D., B¨r D., Lauria M.: Swarm-bot: from concept to implementation. In: Lee G., Yuj J. (eds): Proc. of the IEEE/RSJ International Conference on Intelligent Robot and Systems (2003) 1626-1631.

[12] F. Zambonelli, M.P. Gleizes, M. Mamei, R. Tolksdorf, Spray Computers: Explorations in Self organization", Journal of Pervasive and Mobile Computing 1(1), May 2005.

[13] Mamei, M., Menezes, R., Tolksdorf, R., Zambonelli, F.: Case studies for self organization in computer science. Journal of Systems Architecture (2006).

[14] Azas H., at al.: A relative positioning system for co-located mobile devices. In 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys 2005, Seattle, Washington, USA, June 6-8, 2005. ACM Press, 177–190.

[15] Trohbach S., et al.: Cooperative artefacts: Assessing real world situations with embedded technology. In 6th International Conference on Ubiquitous Computing, UbiComp 2004, Nottingham, UK, September 7-10, 2004. LNCS, vol. 3205. Springer, 250–267.

[16] Lee G., et al.: Decentralized formation control for small-scale mobile robot teams. In Proceedings of the 2006 JSME Conference on Robotics and Mecha tronics, Waseda, Japan, May 26-28, 2006.

[17] Zhang C., et al.: Spatial Data Correlation Based Clustering Algorithms for Wireless Sensor Networks. In 3rd International Conference on Innovative Computing Information and Control, June 2008.

[18] Virrankoski R. and Savvides A.: TASC: topology adaptive spatial clustering for sensor networks. In MASS 2005: Proceedings of the 2nd IEEE International Conference on Mobile Adhoc and Sensor Systems, 2005.

[19] Yang Y., et al.: A Decentralized and Adaptive Flocking Algorithm for Autonomous Mobile Robots. In The 3rd International Conference on Grid and Pervasive Computing – Workshops, May 2008.