


Space-Fluid Adaptive Sampling: A Field-Based, Self-organising Approach

Roberto Casadei[1(✉)], Stefano Mariani[2], Danilo Pianini[1], Mirko Viroli[1], and Franco Zambonelli[2]

[1] Alma Mater Studiorum—Università Bologna, Cesena, Italy
{roby.casadei,danilo.pianini, mirko.viroli}@unibo.it
[2] Università di Modena e Reggio Emilia, Reggio Emilia, Italy
{stefano.mariani, franco.zambonelli}@unimore.it

Abstract. A recurrent task in coordinated systems is managing (estimating, predicting, or controlling) signals that vary in space, such as distributed sensed data or computation outcomes. Especially in large-scale settings, the problem can be addressed through decentralised and situated computing systems: nodes can locally sense, process, and act upon signals, and coordinate with neighbours to implement collective strategies. Accordingly, in this work we devise distributed coordination strategies for the estimation of a spatial phenomenon through collaborative adaptive sampling. Our design is based on the idea of dynamically partitioning space into regions that compete and grow/shrink to provide accurate aggregate sampling. Such regions hence define a sort of virtualised space that is "fluid", since its structure adapts in response to pressure forces exerted by the underlying phenomenon. We provide an adaptive sampling algorithm in the field-based coordination framework. Finally, we verify by simulation that the proposed algorithm effectively carries out a spatially adaptive sampling.

Keywords: Field-based coordination · Distributed leader election · Aggregate processes · Fluidware · Space-fluid computation

1 Introduction

A recurrent problem in engineering is dealing with (e.g., estimating, predicting, or controlling) phenomena that vary in space. Examples include the displacement of waste in a city, the pollution in a geographical area, the temperature in a large building. The problem can be addressed by deploying sensors and actuators in space, processing collected data, and possibly planning actuations [31]. In many settings, the computational activity can (or have to) be performed *in-network* [5] in a *decentralised* way: in such systems, nodes locally sense, process,

M. H. ter Beek and M. Sirjani (Eds.): COORDINATION 2022, LNCS 13271, pp. 99–117, 2022.
https://doi.org/10.1007/978-3-031-08143-9_7

and act upon the environment, and coordinate with *neighbour* nodes to collectively self-organise their activity. However, in general there exists a trade-off between performance and efficiency, that suggests to concentrate the activities on few nodes, or to endow systems with the capability of autonomously adapt the *granularity* of computation [32].

In this work, we focus on sampling signals that vary in space. Specifically, we would like to sample a spatially distributed signal through device coordination and self-organisation such that the samples accurately reflect the original signal and the least amount of resources is used to do so. In particular, we push forward a vision of space-fluid computations, namely computations that are fluid, i.e. change seamlessly, in space and – like fluids – adapt in response to pressure forces exerted by the underlying phenomenon. We reify the vision through an algorithm that handles the shape and lifetime of leader-based "regional processes" (cf. [21]), growing/shrinking as needed to sample a phenomenon of interest with a (locally) maximum level of accuracy and minimum resource usage. For instance, we would like to sample more densely those regions of space where the spatial phenomenon under observation has high variance, to better reflect its spatial dynamics. On the contrary, in regions where variance is low, we would like to sample the phenomenon more sparsely, so as to, e.g., save energy, communication bandwidth, etc. while preserving the same level of accuracy.

Accordingly, we consider the *field-based coordination* framework of *aggregate computing* [2,29], which has proven to be effective in modelling and programming self-organising behaviour in situated networks of devices interacting asynchronously. On top of it, we devise a solution that we call *aggregate sampling*, inspired by the approaches of self-stabilisation [28] and density-independence [3], that maps an input field representing a signal to be sampled into a regional partition field where each region provides a single sample; then, we characterise the aggregate sampling error based on a distance defined between stable snapshots of regional partition fields, and propose that an *effective* aggregate sampling is one that is locally optimal w.r.t. an error threshold, meaning that the regional partition cannot be improved simply by merging regions. In summary, we provide the following contributions:

- we define a model for distributed collaborative adaptive sampling and characterise the corresponding problem in the field-based coordination framework;
- we implement an algorithmic solution to the problem that leverages self-organisation patterns like gradients [6,28] and coordination regions [21] as well as *aggregate processes* [4];
- we experimentally validate the algorithm to verify interesting trade-offs between sparseness of the sampling and its error.

The rest of the paper is organised as follows. Section 2 covers motivation and related work. Section 3 provides a model for distributed sampling and the problem statement. Section 4 describes an algorithmic solution to the problem of sampling a distributed signal using the framework of aggregate computing. Section 5 performs an experimental validation of the proposed approach. Finally, Sect. 6 provides conclusive thoughts and delineates directions for further research.

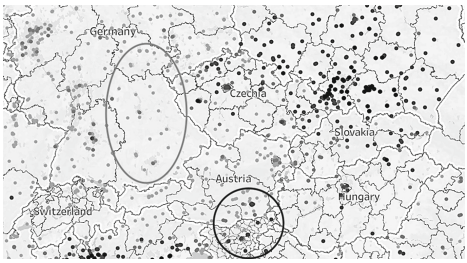

Fig. 1. Air quality statistics map taken from https://archive.ph/dMJO2. There are areas where the underlying phenomenon does not vary significantly in space (light-grey oval), hence sampling could be "sparsified" with tolerable loss of accuracy in modelling the observed phenomenon. In others (darker circle) variance is high, requiring a more detailed spatial sampling.

2 Motivation and Related Work

Consider a wireless sensor network (WSN) of any topology deployed across a geographical area to monitor a spatially-distributed phenomenon, such as, for instance, air quality, as depicted in Fig. 1. We want to dynamically and adaptively find a partitioning that minimises the number and maximises the size of regions, while preserving as much as possible the underlying information. Hence, in areas with low variance, we want our regions to be larger, as many samples will report similar values. Conversely, in areas with high spatial variance, smaller regions are necessary as even proximal samples may have very different values.

There are several approaches in the literature that attempt to solve similar problems, in different research areas and with different techniques. In *adaptive sampling* [17,27] the goal is to extend or reduce the set of samples drawn depending on temporal dynamics or across space. There, most of the literature is about designing fixed strategies for sensor placement (at design-time), sensor selection (at run-time), or so-called sampling designs, that is, how the sampling process may be adaptive to either network-level measures (energy consumption, communication costs, sensor distance, etc.) or domain-level measures (information gain, entropy, correlation, etc.). Our approach fundamentally differs in core aspects, such as full distribution of computations, and full self-adaptiveness to the observed phenomenon without any a-priori knowledge.

Other research focuses on *mobile* sensors, e.g., robots [7,8,15,25], hence the goal is to move them to the most informative sensing locations. Some work aims to adapt the sampling process to cope with *spatial* (and, often, temporal) phenomena, preserving some spatial properties of the WSN or the phenomenon under observation [9,12,26]. These pursue a goal similar to the one of this work, but with different techniques: we leverage a programmable, adaptive, and self-organising approach based on the field calculus and aggregate programming, whereas they adopt heterogeneous tools rooted in geometric frameworks (such as Voronoi tessellation), information theory, and optimisation.

Finally, there are approaches explicitly relying on adaptive and spatial *clustering* techniques [13,14,30], where device partitioning is meant to improve energy efficiency or reduce communication costs by electing leaders that perform sampling on behalf of the whole cluster. These approaches are mostly driven by network-level metrics, whereas we rely on any arbitrary univariate statistics of the phenomenon under observation.

In [30], the proposed approach combines distance measurements, connectivity, and density information of sensors (not the underlying phenomenon) to define clusters with similar deployment density that group devices in close proximity. The objective is to produce better deployments of sensors whose aggregate measurements can benefit energy consumption. On the contrary, our objective is to create irregularities in the device network to better represent the phenomenon under observations, with a trade-off about not wasting resources while doing so. Accordingly, we do clustering through leader election as in [30], but based on univariate statistics of the observed phenomenon, rather than on network-related information. With this respect, the work in [14] groups together sensors with similar readings, hence is similar to ours in how clusters are formed. However, the authors attempt to build clusters with minimal variance in size, that is contrary to our goal of adaptively shape clusters with different sizes so as to better sample the underlying spatial phenomenon. Finally, the work in [12] is a rare example of an adaptive algorithm that is also concerned with up-scaling sampling when is needed, whereas most efforts go in the direction of down-scaling for energy saving purposes. In this it is similar to our own proposal. However, they exploit the assumption there measures coming from sensors in close proximity are highly correlated, whereas we specifically focus on those domains where the opposite may be true.

3 Distributed Aggregate Sampling: Model

In order to define the problem and characterise our approach, we leverage the event structure framework [1].

3.1 Computational Model

We consider a computational model where a set of *devices* compute at discrete steps called *computation rounds* and interact with *neighbour devices* by exchanging messages. Executions of such systems can be modelled through *event structures* [18]. Following the general approach in [1], we enrich the event structure with information about the devices where events occur.

Definition 1 (Situated event structure). *A* situated event structure *(ES) is a pair* $\langle E, \rightsquigarrow, d \rangle$ *where*

- E *is a countable set of* **events**,
- $\rightsquigarrow \subseteq E \times E$ *is a* **messaging** *relation from a* **sender** *event to a* **receiver** *event (these are also called* **neighbour *events***),

– $d : E \to \Delta$, *where* Δ *is finite, maps an event to the device where it occurs,*

such that:

– *the transitive closure of* $\rightsquigarrow$ *forms an irreflexive partial order* $< \subseteq E \times E$, *called* **causality** *relation (an event* ϵ *is in the* **past** *of another event* ϵ' *if* $\epsilon < \epsilon'$, *in the* **future** *if* $\epsilon' < \epsilon$, *or* **concurrent** *otherwise);*
– *for any* $\delta \in \Delta$, *the projection of the ES to the set of events* $E_\delta = \{\epsilon \in E \mid d(\epsilon) = \delta\}$ *forms a well-order, i.e., a sequence* $\epsilon_0 \rightsquigarrow \epsilon_1 \rightsquigarrow \epsilon_2 \rightsquigarrow \ldots$;

We also define:

– *Notation* $recvs(\epsilon) = \{\delta \in \Delta \mid \delta = d(\epsilon') \implies \epsilon \rightsquigarrow \epsilon'\}$ *to denote the set of receivers of* ϵ, *i.e., the devices receiving a message from* ϵ.
– *Notation* $T^-_{\epsilon_0} = \{\epsilon : \epsilon < \epsilon_0\}$ *to denote the past event cone of* ϵ_0 *(finite set).*
– *Notation* $T^+_{\epsilon_0} = \{\epsilon : \epsilon_0 < \epsilon\}$ *to denote the future event cone of* ϵ_0.
– *Notation* $X|_{E'}$ *to denote the projection of a set, function, or ES thereof* X *to the set of events* $E' \subseteq E$. *Note that the projection of an event structure to the future event cone of an event is still a well-formed ES.*

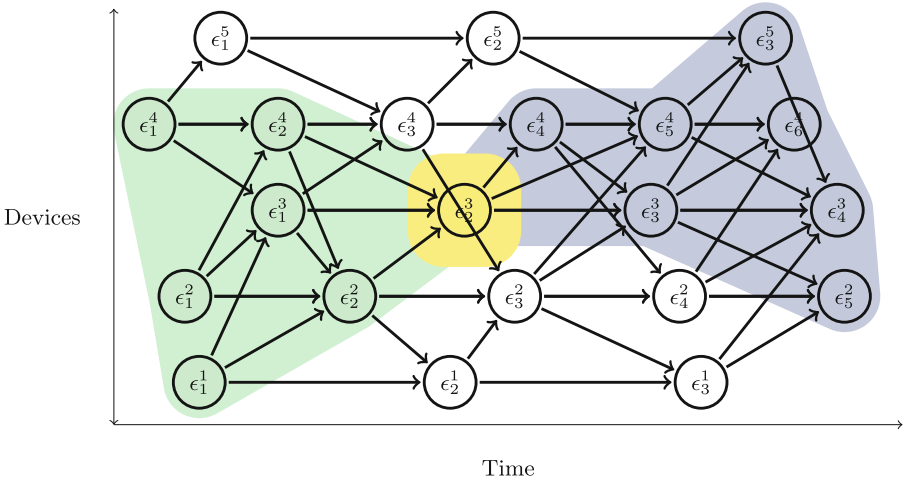

Fig. 2. Example of an event structure. In the node labels, superscripts denote device identifiers, while subscripts are progressive numbers denoting subsequent rounds at the same device. The blue (resp. green) background denotes the future (resp. past) of a reference event denoted with a yellow background. (Color figure online)

An example of ES is given in Fig. 2, events denote computation rounds. Note that self-messages (i.e., messages from an event to the next on the same device) model persistence of *state* over time. In the computation model we consider, based on [1], each event ϵ represents the execution of a program taking all incoming

messages, and producing an outgoing message (sent to all neighbours) and a result value associated with ϵ. Such "map" of result values across all events defines a computational field, as follows.

Definition 2 (Computational field). *Let* $\mathbf{E} = \langle E, \rightsquigarrow, d \rangle$ *be an event structure. A* computational field *on* $\mathbf{E}$ *is a function* $f : E \to \mathbb{V}$ *that maps every event* ϵ *in* E *(also called the* domain *of the field) to some value in a value set* $\mathbb{V}$.

Computational fields are essentially the "distributed values" which our model deals with; hence computation is captured by the following definition.

Definition 3 (Field computation). *Let* $\mathbf{E} = \langle E, \rightsquigarrow, d \rangle$ *be an event structure, and* $\mathbb{F}_{E,\mathbb{V}}$ *be the set of fields on domain* E *with range* $\mathbb{V}$*, i.e.,* $\mathbb{F}_{E,\mathbb{V}} = \{f_i : E \to \mathbb{V}\}$. *A* field computation *over* $\mathbf{E}$ *is a function* $\Phi_{\mathbf{E}} : \mathbb{F}_{E,\mathbb{V}} \to \mathbb{F}_{E,\mathbb{V}}$ *mapping an input field to an output field on the same domain of* $\mathbf{E}$.

This definition naturally extends to the case of zero or multiple input fields.

Definition 4 (Field operator). *A field operator (or field program) is denoted as the result of computation on any possible environment, hence it is a function* P *taking an ES and yielding the field computation that would occur on it, namely* $P(\mathbf{E}) = \Phi_{\mathbf{E}}$.

3.2 Self-stabilisation

We now provide the definitions necessary to model *self-stabilisation* following the approach in [28]. Namely, the following definitions capture the idea of *adaptiveness* whereby as the environment of computation stabilises, then the result of computation stabilises too, and such a result does not depend on previous transitory changes.

Definition 5 (Static environment). *An event structure* $\mathbf{E} = \langle E, \rightsquigarrow, d \rangle$ *is said to be a* static environment *if it has stable topology, namely all events of a given device always share the same set of receivers, i.e.,* $\forall \epsilon, \epsilon', d(\epsilon) = d(\epsilon') \Rightarrow recvs(\epsilon) = recvs(\epsilon')$.

Definition 6 (Stabilising environment). *An event structure* $\mathbf{E} = \langle E, \rightsquigarrow, d \rangle$ *is said to be a* stabilising environment *if it is eventually static, i.e.,* $\exists \epsilon_0 \in E$ *such that* $\mathbf{E}|_{T^+_{\epsilon_0}} = \langle E|_{T^+_{\epsilon_0}}, \rightsquigarrow|_{T^+_{\epsilon_0}}, d|_{T^+_{\epsilon_0}} \rangle$ *is static. In this case we say it is* static since *event* ϵ_0.

Definition 7 (Stabilising field). *Let event structure* $\mathbf{E} = \langle E, \rightsquigarrow, d \rangle$ *be a stabilising environment, static since event* ϵ_0. *A field* $f : E \to \mathbb{V}$ *is said* stabilising *if it eventually provides stable output, i.e.,* $\exists \epsilon > \epsilon_0$ *such that* $\forall \epsilon' > \epsilon$ *it holds that* $d(\epsilon) = d(\epsilon') \implies f(\epsilon) = f(\epsilon')$.

Definition 8 (Stabilising computation). *A field computation* $\Phi_{\mathbf{E}} = \mathbb{F}_{E,\mathbb{V}} \to \mathbb{F}_{E,\mathbb{V}}$ *is said* stabilising *if, when applied to a stabilising input field, it yields a stabilising output field.*

Definition 9 (Self-stabilising operator). *A field operator (or program) P is said self-stabilising, if in any stabilising environment $\mathbf{E}$ it yields a stabilising computation $\Phi_{\mathbf{E}}$ such that, for any pair of input fields f_1, f_2 eventually equal, i.e. $f_1|_{T^+_\epsilon} = f_2|_{T^+_\epsilon}$ for some event ϵ, their ouput is eventually equal too, i.e., there exists a $\epsilon' > \epsilon$ such that $\Phi_{\mathbf{E}}(f_1)|_{T^+_{\epsilon'}} = \Phi_{\mathbf{E}}(f_2)|_{T^+_{\epsilon'}}$*

3.3 Problem Definition

We start by introducing the notion of *regional partition*, which is a finite set of non-overlapping contiguous clusters of devices: a notion that prepares the ground to that of an *aggregate sampling* which we introduce in this paper.

Definition 10 (Regional partition field). *Let $\mathbf{E} = \langle E, \rightsquigarrow, d\rangle$ be a stabilising environment static since event ϵ_0. A* regional partition field *is a stabilising field $f : E \to \mathbb{V}$ on $\mathbf{E}$ such that:*

- ***(finiteness)** the image $Img(f) = \{f(x) \mid x \in E\}$ is a finite set of values;*
- ***(eventual contiguity)** there exists an event $\epsilon'_0 > \epsilon_0$ such that for any pair of events $\epsilon_1, \epsilon_n \in T^+_{\epsilon'_0}$ with $\epsilon_1 < \epsilon_n$, if $f(\epsilon_1) = f(\epsilon_n)$ then there exists a sequence of events (ϵ_i) connecting ϵ_1 to ϵ_n where $f(\epsilon_i) = f(\epsilon_1) = f(\epsilon_n)\ \forall n$.*

Note that the set of domains of regions induced by f is defined by $regions(f) = \{f^{-1}(v) : v \in Img(f)\}$.

An example of a regional partition field is shown in Fig. 3. Notice that for any pair of events in the same space-time region there exists a path of events entirely contained in that region. Also notice that, by this definition, different disjoint regions denoted by the same value r are not possible.

Definition 11 (Aggregate sampling). *An* aggregate sampling *is a stabilising computation $\Phi_S : \mathbb{F}_{E,\mathbb{V}} \to \mathbb{F}_{E,\mathbb{V}}$ that, given in input a field to be sampled, it outputs a regional partition field.*

Once we have defined an aggregate sampling process in terms of its inputs, outputs, and stabilising dynamics, we need a way to measure the error introduced by the aggregate sampling. To this purpose, we introduce the notion of a *stable snapshot*, namely a field consisting of a sample of one event per device from the stable portion of a stabilising field.

Definition 12 (Stable snapshot). *Let $\mathbf{E} = \langle E, \rightsquigarrow, d\rangle$ be an event structure, and $f : E \to \mathbb{V}$ be a stabilising field on $\mathbf{E}$ which provides stable output from $\epsilon_0 \in E$. We define a* stable snapshot *of field f as a field obtained by restricting f to a subset of events in the future event cone of ϵ_0* and *with exactly one event per device, i.e., a field $f_S : E_S \to \mathbb{V}$ such that $E_S \subseteq T^+_{\epsilon_0}$,* and $\forall \epsilon, \epsilon' \in E_S : d(\epsilon) = d(\epsilon') \implies \epsilon = \epsilon'$, and $\forall \epsilon \in T^+_{\epsilon_0}, \exists \epsilon' \in E_S : d(\epsilon') = d(\epsilon)$.

Definition 13 (Spatial field distance). *A* spatial field distance $\mu : \mathbb{F}_{E,\mathbb{V}} \times \mathbb{F}_{E,\mathbb{V}} \to \mathbb{R}^+_0$ *is a function from pairs of spatial fields with the same domain to non-negative reals, such that:*

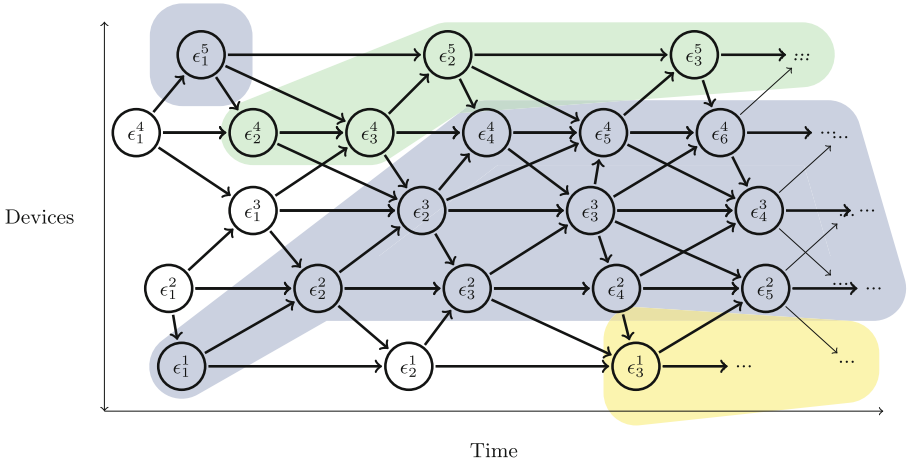

Fig. 3. Example of a regional partition field with regions r_{blue}, r_{green}, r_{yellow}, r_{white} (the background is used to denote the output of the field). Notice that contiguity does not hold everywhere and anytime but only since event ϵ_2^3.

- (identity) $\mu(f, f') = 0 \iff f = f'$;
- (additivity) $\mu(f, f') \geq \mu(f|_{E'}, f'|_{E'})$ *for any* $E' \subseteq E$.

Definition 14 (Aggregate sampling error). *Let* $\Phi_{\mathbf{E}} : \mathbb{F}_{E,\mathbb{V}} \to \mathbb{F}_{E,\mathbb{V}}$ *be an aggregate sampling, and consider an input field* $f_i : E \to \mathbb{V}$ *and corresponding output regional partition* $f_o : E \to \mathbb{V}$. *We say that* f_o samples f_i within error η according to distance μ, *if the distance of stable snapshots of* f_i *and* f_o *in any region is not bigger then* η, *that is: let* ϵ_0 *be an event from which* $\mathbf{E}$ *is static and* f_i *and* f_o *are stable, let* f_i^s *and* f_o^s *be stable snapshots of* f_i *and* f_o *in the future cone of* ϵ_0, *then for any region* $E' \in regions(f_o^s)$, *we have* $\mu(f_i^s|_{E'}, f_o^s|_{E'}) \leq \eta$.

The aggregate sampling error provides a measure of *accuracy*, but in general we are also interested in *efficiency*, namely, in the ability of a regional partition to be accurate while relying on a small number of regions. Among the various possible definitions, we introduce the following notion of local optimality, stating that no region in the regional partition could be attached to an existing region without exceeding the desired error.

Definition 15 (Local optimality of a regional partition). *Let* $\Phi_{\mathbf{E}} : \mathbb{F}_{E,\mathbb{V}} \to \mathbb{F}_{E,\mathbb{V}}$ *be an aggregate sampling, and consider an input field* f_i *and corresponding output regional partition* f_o *such that* f_o *samples* f_i *within error* η *according to distance* μ. *We say that* f_o *is* locally optimal *under error* η *if for no pairs of contiguous regions* $E', E'' \in regions(f_o)$ *we have* $\mu(f_i^s|_{E' \cup E''}, f_o^s|_{E' \cup E''}) \leq \eta$—*with spatial fields* f_i^s *and* f_o^s *obtained as in Definition 14.*

Notice that, since our goal is to deal with dynamic phenomena and large-scale environments, we are not interested in finding globally optimal solutions, but

rather heuristics for self-organising behaviour. So, we are now ready to define the goal operator for this paper.

Definition 16 (Effective sampling operator). *An* effective sampling operator *is a self-stabilising operator P_η, parametric in the error bound η, such that in any stabilising environment* $\mathbf{E}$ *and stabilising input f_i, a locally optimal regional partition within error η is produced.*

4 Aggregate Computing-Based Solution

In this section, we define an *effective aggregate sampling* within the framework of Aggregate Computing [2,29]. The approach is rooted in the idea that a system can be partitioned into regions by identifying *leader* devices (cf. algorithms for sparse-choice leader election [16]) associated with a set of devices expanding until the *aggregate sampling error* within the growing region is under some threshold—while ensuring there is no overlap with other regions (i.e., each device is associated with exactly one leader). More precisely:

1. each device announces its candidature for leader;
2. each device propagates to neighbours the candidature of the device it currently recognises as leader, fostering expansion of its corresponding region;
3. in a region, the *aggregate sampling error* is computed, which monotonically increases with the hop-distance from the leader;
4. devices discard candidatures whose errors exceed a threshold;
5. in case multiple valid candidatures (i.e., whose error is under the threshold) reach a device, one is selected based on a *competition policy*.

Competition and Leader Strength. Although competition among leaders could be realised in several ways, many techniques may lead to non-self-stabilising behaviour: for instance, if the winning leader is selected randomly in the set of those whose error is under threshold, regions may keep changing even in a static environment. In this work, we propose a simple strategy: every leader associates its candidature with the local value of a field that we call *leader strength*; in case of competing candidatures, the highest such value is selected as winner, breaking the symmetry. The leader strength can be of any orderable type, and its choice impacts the overall selection of the regions by imposing a selection priority over leaders (hence on region-generation points).

Region Expansion and Error Metric. Inspired by previous work on distributed systems whose computation is independent from device distribution [3], we accumulate an *error metric* along the path from the leader device towards other devices along a gradient [6], a distributed data structure proven to be self-stabilising [28]. We thus have two major drivers: 1. the leader strength affects the creation of the regions by influencing the positions of their source points; 2. the error metric influences the expansion in space of the region across all directions, mandating its size and (along with the interaction with other regions) its shape.

For instance, a metric could be the absolute value of the difference in the perceived signal between two devices: devices perceiving very different values would tend not to cluster together (even if spatially close), as they would perceive each other as farther away (leading to irregular shapes).

For the competition process to progress, we need that, in zones where multiple regions are expanding concurrently, multiple gradients run concurrently and overlapping [19]. This requires special handling in the aggregate computing framework, hence we modeled the expansion of each region as a separate aggregate process [4].

As the next section verifies through simulations, linking region expansion with the error metric enables to find locally optimal sampling regions, in the sense that all regions are necessary: be removing some region (e.g. merging it with a nearby region) less resources will be used, indeed, but accuracy would be compromised.

5 Evaluation

In this section, we validate the behaviour of the proposed effective aggregate sampling algorithm. The goals of the evaluation are the following:

- *stabilisation*: we expect the algorithm to be *self-stabilising*, and thus to behave in a self-stabilising way under different conditions (as per Definition 9);
- *high information (entropy)*: we expect the algorithm to split areas with different measurements, namely, to dynamically increase the number of regions on a per-need basis to minimise the *aggregate sampling error* (as per Definition 14);
- *error-controlled upscaling*: we expect the algorithm to not abuse of region creation, but to keep the minimum number of regions (hence of the largest size) required to maintain accuracy (as per Definition 15), intuitively, grouping together devices with similar measurements.

Clearly, upscaling and high information density are at odds: maximum information is achieved by maximising the number of regions, and thus assigning each device its unique region, but this would prevent any upscaling. On the other hand, the maximum possible upscaling would be achieved when all devices belong to the same region, thus minimising information. We want our regions to change in space "fluidly" and opportunistically tracking the situation at hand, achieving a trade-off between upscaling and amount of information (as per Definition 16).

5.1 Scenario

We challenge the proposed approach by letting the algorithm operate on different deployments of one thousand devices and different data sources. We deploy devices into a square arena with different topologies: *i) grid (regular grid)*: devices are regularly located in a grid; *ii) pgrid (perturbed/irregular grid)*: starting from a grid, devices' positions are perturbed randomly on both axes;

iii) uniform: positions are generated with a uniform random distribution; *iv) exp (exponential random)*: positions are generated with a uniform random distribution on one axis and with an exponential distribution on the other, thus challenging device-distribution sensitivity. In all cases, we avoid network segmentation by forcing each device to communicate *at least* with the eight closest devices.

We simulate the system when sampling the following phenomena: *(i) Constant*: the signal is the same across the space, we expect the system to upscale as much as possible; *(ii) Uniform*: the signal has maximum entropy, each point in space has a random value, we thus expect the system to create many small regions; *(iii) Bivariate gaussian (gauss)*: the signal has higher value at the center of the network, and lower towards the borders, producing a gaussian curve whose expected value is located at the center of the network, we expect regions to be smaller where the data changes more quickly; *(iv) Multiple bivariate gaussian (multi-gauss)*: similar to the previous case, but the signal value is built by summing three bivariate gaussian whose expected value is one third of the previous gaussian, and whose expected values are located along the diagonal of the network (bottom-left corner, center, top-right corner); *(v) Dynamic*: the system cycles across the previous states, we use this configuration to investigate whether and how the proposed solution adapts to changes in the structure of the signal.

5.2 Parameters

The proposed solution can be tuned using three major knobs: the leader strength, the error tolerance, and the distance metric. In these experiments, we fix the error tolerance to a constant value throughout the board, while we try three different solutions for the leader strength and the distance metric.

For the former, we consider: *(i) value*: the local value of the tracked signal s; *(ii) mean*: the neighborhood-mean value of the tracked signal s, assuming N to be the set of neighbors (including the local device), and s_i to be the value of the tracked signal at device $i \in N$, the value is computed as: $M = \sum_{i \in N} s_i / |N|$; *(iii) variance*: the neighborhood-variance of the tracked signal s, assuming M_i to be the neighborhood-mean computed at device $i \in N$, the value is computed as: $\sum_{i \in N} (M_i - s_i)^2 / |N|$.

For the latter, we consider: *(i) distance*: the spatial distance is used as distance metric; *(ii) diff*: assuming that s_i is the value of the tracked signal at device i, the distance between two neighboring devices a and b is measured as: $e_{ab} = e_{ba} = min(\epsilon, |s_a - s_b|)$, where $\epsilon \in \mathbb{R}_+$, $\epsilon = 0$ iff $a = b$, $0 < \epsilon \ll 1$ otherwise, we bound the minimum value to preserve the triangle inequality; *(iii) mix*: we mix the two previous metrics so that both error and physical distance concur in the distance definition, assuming $\overline{ab}$ to be the spatial distance between devices a and b, we measure the mix metric as: *(iv)* $\overline{ab} \cdot e_{ab}$.

5.3 Metrics

We investigate the behaviour of the system through the following metrics, assuming, at any given time, a set of devices D partitioned into a set of regions $R = R_1 \cup \cdots \cup R_{|R|}$, where each R_r is a set of devices $\{D^r_1, \ldots, D^r_{|R_r|}\}$, and each device D^r_d reads the local value of the tracked signal s^r_d:

- *Region count $|R|$ (regions).* Counting the regions provides an indication about efficiency (Definition 15): more partitions should be present in contexts where the sampled signal has higher entropy;
- *Mean region size $\mu_R = \sum_{i=1}^{|R|} |R_i|/|R|$ (devices).* Ancillary to the region count, but density-sensitive: when devices are distributed irregularly (as in the **exp** deployment, see Sect. 5.1), we expect this metric to be less predictable;
- *Standard deviation of the mean of the signal in regions $\sigma(\mu_s)$ (same unit of the signal).* Proxy for inter-region difference, high values indicate large differences between different regions. The mean signal inside region R_r is computed as:

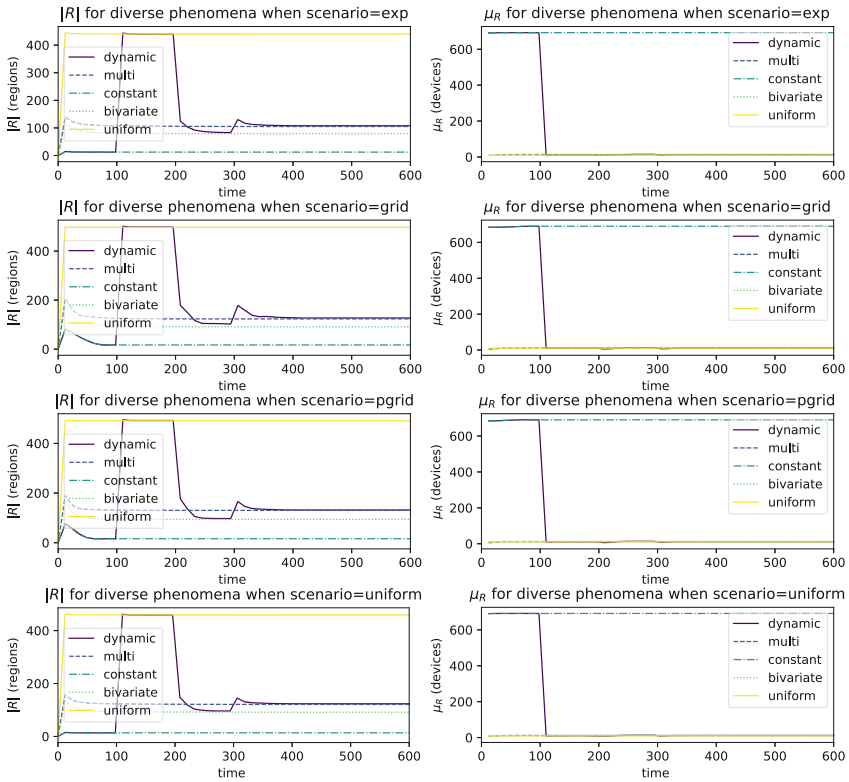

Fig. 4. Region count (left column) and size (right column) across deployments and scenarios. The system behaves very similarly regardless of the device disposition. As expected, the higher information density leads to more smaller regions. The dynamic scenario shows that the partitions change in response to changes in the signal.

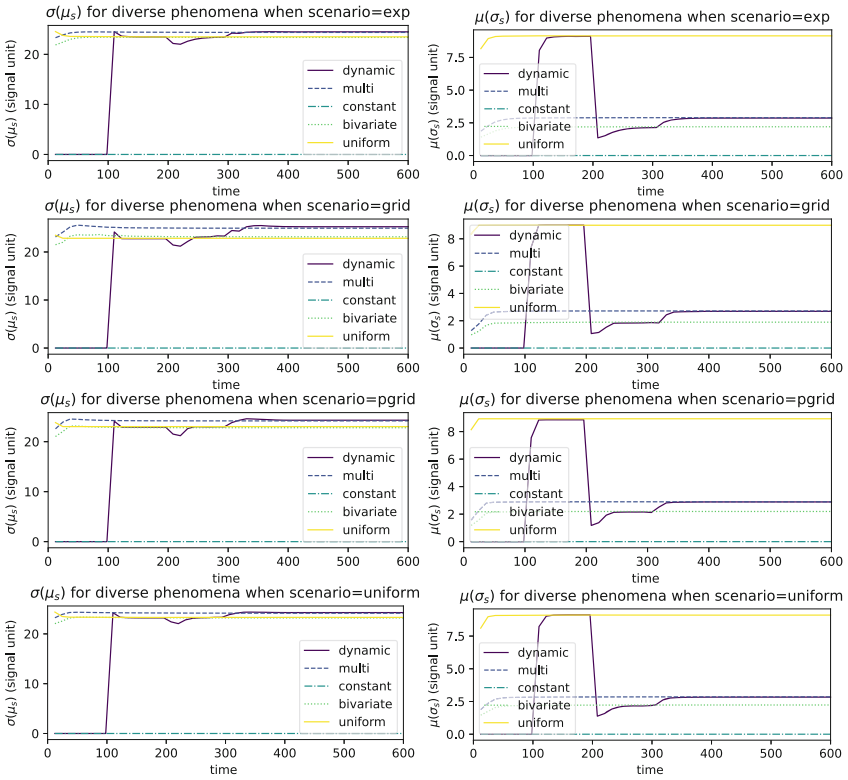

Fig. 5. Standard deviation of the mean region value (left) and mean standard deviation (right) across deployments and scenarios, indicating respectively how much the regions readings differ from each other (the higher the more different) and how the regions are internally similar (the lower the more homogeneous are regions). The constant and uniform random signals work as baselines: in the former case, very large areas gets formed, while in the latter most regions count a single device (as expected). In the other cases, inter-region differences is maximized (they get as high as the most extreme case) keeping internal consistency under control.

$\mu_s^{R_r} = \sum_{i=1}^{|R_r|} s_i^r / |R_r|$, the mean of the means of the signal is $\mu_s^R = \sum_{i=1}^{|R|} \mu_s^{R_i} / |R|$, thus $\sigma(\mu_s) = \sqrt{\frac{1}{|R|} \sum_{i=1}^{|R|} (\mu_s^{R_i} - \mu_s^R)^2}$;

– *Mean standard deviation of the signal in regions $\mu(\sigma_s)$ (same unit of the signal).* Proxy for intra-region error. The lower this value, the more similar are the signal readings inside regions, hence the lower the error induced by the grouping (Definition 14). The standard deviation of a the tracked signal inside region R_r is computed as: $\sigma_s^{R_r} = \sqrt{\frac{1}{|R_r|} \sum_{i=1}^{|R_r|} (s_i^r - \mu_s^{R_r})^2}$, thus $\mu(\sigma_s) = \sum_{i=1}^{|R|} \sigma_s^{R_i} / |R|$;

– *Standard deviation of the standard deviation of the signal in regions $\sigma(\sigma_s)$ (same unit of the signal).* Proxy metric for the consistency of

partitioning. Higher values indicate that partitions have different internal error, hence behave differently (striving to satisfy Definition 16). Computed as: $\sigma(\sigma_s) = \sqrt{\frac{1}{|R|}\sum_{i=1}^{|R|}(\sigma_s^{R_r} - \mu(\sigma_s))^2}$.

5.4 Implementation and Reproducibility

We rely on a prototype implementation realised in the Protelis programming language [23]. The simulations were realised using Alchemist [22], the data analysis has been performed using Xarray [10] and matplotlib [11]. For each element in the cartesian product of the device deployment type, signal form, leader strength, and distance metric, an experiment was performed. Each simulation has been repeated 100 times with a different random seed; random seeds control both the evolution of the system (the order in which devices compute) and their position on the arena (except for the regular grid deployment, which is not randomised); the presented results are the average across all repetitions; when a chart does not mention some of the parameters, then the results that are presented are also averaged across all values the parameter may assume for the simulation set. The experiment has been open sourced, publicly released[1], documented, equipped with a continuous integration system to guarantee replicability, and published as a permanently available, reusable artifact [24].

5.5 Results

We present the major findings of our analysis, whose complete version counts 630 charts, available to the interested reader in the experiment repository. In Fig. 4,

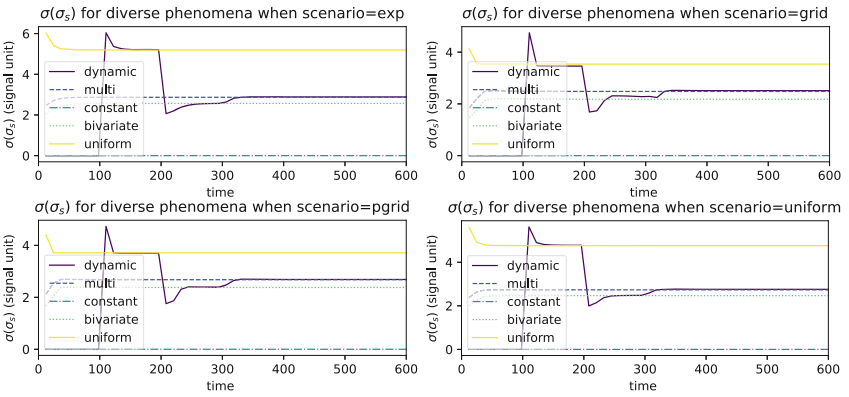

Fig. 6. Intra-region partitioning homogeneity, measured as the standard deviation across regions of the standard deviation of the signal inside regions. Higher values indicate that different regions are more heterogeneous, namely that some have more error than others.

[1] https://github.com/DanySK/Experiment-2022-Coordination-Space-Fluid.

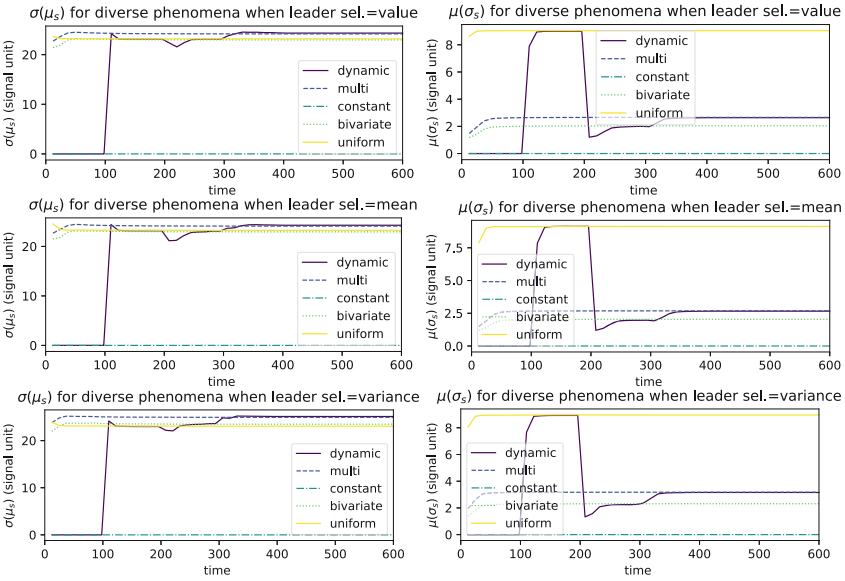

Fig. 7. Effect of different leader selection policies. The behaviour of the system is similar regardless of the way the region leader is selected.

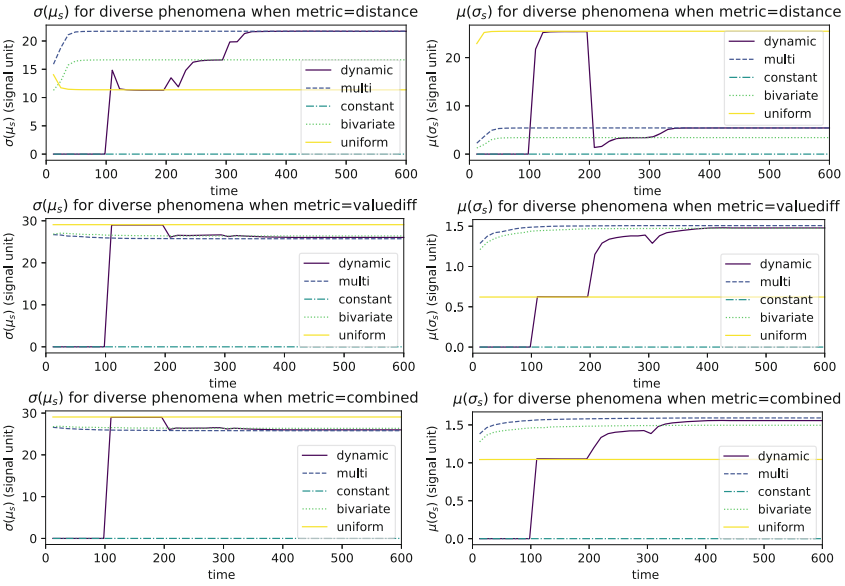

Fig. 8. Effect of different error measurement metrics. The system is very sensible to the metric used to accumulate error, which directly impacts the way distance is perceived, thus determining the maximum size and number of areas.

we show that our prototype implementation stabilises, as after a short transition all values become stable. Of course, in the dynamic case, these transitions are present throughout the experiment. As expected, the aggregate sampler defines a number of regions that differs depending on the underlying phenomenon under observation. From Fig. 5 and Fig. 6, we observe that the system tries indeed to maximise inter-region differences and minimise intra-region differences, thus effectively accounting or the tradeoff between *high information (entropy)* and *error-controlled upscaling* (as per Definition 16). Finally, Fig. 7 and Fig. 8 show how the algorithm responds to changing parameters. As expected, while modifying the leader selection policy has minimal impact on the behaviour of the system, changing the error metric modifies its behaviour greatly. In all cases, we observe that the driver signal with higher information entropy (uniform) generates more smaller regions than all other signals, while the one with lowest information entropy (constant) always produces few (usually one) large regions. The reason is that the leader selection impacts the originating point of a region, but it is its expansion (driven by the metric) that in the end determines both extension and shape.

6 Conclusions and Future Work

In this paper, we tackled the problem of defining an aggregate sampler sensitive to the spatial dynamics of the phenomenon under observation. In particular, we wanted to minimize the sampling error (minimum when all available sampling devices are used) while also minimising the regions count, two contrasting needs.

We formalised the problem within the framework of field calculus and aggregate computing, suitable to represent situated, large-scale, and dynamic computations. We thus designed a spatial adaptive aggregate sampler based on a leader election strategy that dynamically creates and grows/shrinks sampling clusters (or regions) based on error metric and leader strength. Through simulation, the sampler is shown to satisfy the mentioned tradeoff (*effective sampling*, Definition 16).

As measuring performance and efficiency of such an adaptive algorithm is far from trivial, we exploited several metrics to validate intended behaviour. However, as a follow-up work we would like to synthesize a single indicator able to measure both accuracy and efficiency, using information theory such as those derived from entropy (e.g. mutual information). Also, we are analyizing openly available air pollution datasets to design new simulations based on real-world data, so as to better emphasize the impact that our aggregate sampler could have for policy making based on spatial phenomena. Finally, future work will be devoted to investigating how space-fluid sampling can integrate with time-fluid aggregate computations [20].

Acknowledgements. This work has been supported by the MIUR PRIN 2017 Project "Fluidware" (N. 2017KRC7KT) and the MIUR FSE REACT-EU PON R&I 2014-2022 (N. CCI2014IT16M2OP005).

References

1. Audrito, G., Beal, J., Damiani, F., Viroli, M.: Space-time universality of field calculus. In: Serugendo, G.D.M., Loreti, M. (eds.) Coordination Models and Languages - 20th IFIP WG 6.1 International Conference, COORDINATION 2018, Held as Part of the 13th International Federated Conference on Distributed Computing Techniques, DisCoTec 2018, Madrid, Spain, 18–21 June 2018. Proceedings. LNCS, vol. 10852, pp. 1–20. Springer (2018). https://doi.org/10.1007/978-3-319-92408-3_1
2. Beal, J., Pianini, D., Viroli, M.: Aggregate programming for the internet of things. IEEE Comput. **48**(9), 22–30 (2015). https://doi.org/10.1109/MC.2015.261
3. Beal, J., Viroli, M., Pianini, D., Damiani, F.: Self-adaptation to device distribution in the internet of things. ACM Trans. Auton. Adapt. Syst. **12**(3), 12:1–12:29 (2017). https://doi.org/10.1145/3105758
4. Casadei, R., Viroli, M., Audrito, G., Pianini, D., Damiani, F.: Aggregate processes in field calculus. In: Riis Nielson, H., Tuosto, E. (eds.) COORDINATION 2019. LNCS, vol. 11533, pp. 200–217. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22397-7_12
5. Fasolo, E., Rossi, M., Widmer, J., Zorzi, M.: In-network aggregation techniques for wireless sensor networks: a survey. IEEE Wirel. Commun. **14**(2), 70–87 (2007). https://doi.org/10.1109/MWC.2007.358967
6. Fernandez-Marquez, J.L., Serugendo, G.D.M., Montagna, S., Viroli, M., Arcos, J.L.: Description and composition of bio-inspired design patterns: a complete overview. Nat. Comput. **12**(1), 43–67 (2013). https://doi.org/10.1007/s11047-012-9324-y
7. Garg, S., Ayanian, N.: Persistent monitoring of stochastic spatio-temporal phenomena with a small team of robots. In: Fox, D., Kavraki, L.E., Kurniawati, H. (eds.) Robotics: Science and Systems X, University of California, Berkeley, USA, July 12–16, 2014 (2014). https://doi.org/10.15607/RSS.2014.X.038. http://www.roboticsproceedings.org/rss10/p38.html
8. Graham, R., Cortés, J.: Cooperative adaptive sampling via approximate entropy maximization. In: Proceedings of the 48th IEEE Conference on Decision and Control, CDC 2009, Combined with the 28th Chinese Control Conference, 16–18 December 2009, Shanghai, China, pp. 7055–7060. IEEE (2009). https://doi.org/10.1109/CDC.2009.5400511
9. Hamouda, Y.E.M., Phillips, C.I.: Adaptive sampling for energy-efficient collaborative multi-target tracking in wireless sensor networks. IET Wirel. Sens. Syst. **1**(1), 15–25 (2011). https://doi.org/10.1049/iet-wss.2010.0059
10. Hoyer, S., Hamman, J.: xarray: N-D labeled arrays and datasets in Python. J. Open Res. Softw. **5**(1) (2017). https://doi.org/10.5334/jors.148
11. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55
12. Lee, E.K., Viswanathan, H., Pompili, D.: SILENCE: distributed adaptive sampling for sensor-based autonomic systems. In: Schmeck, H., Rosenstiel, W., Abdelzaher, T.F., Hellerstein, J.L. (eds.) Proceedings of the 8th International Conference on Autonomic Computing, ICAC 2011, Karlsruhe, Germany, 14–18 June 2011, pp. 61–70. ACM (2011). https://doi.org/10.1145/1998582.1998594
13. Lin, Y., Megerian, S.: Sensing driven clustering for monitoring and control applications. In: 4th IEEE Consumer Communications and Networking Conference, CCNC 2007, Las Vegas, NV, USA, 11–13 January 2007, pp. 202–206. IEEE (2007). https://doi.org/10.1109/CCNC.2007.47

14. Liu, Z., Xing, W., Zeng, B., Wang, Y., Lu, D.: Distributed spatial correlation-based clustering for approximate data collection in WSNs. In: Barolli, L., Xhafa, F., Takizawa, M., Enokido, T., Hsu, H. (eds.) 27th IEEE International Conference on Advanced Information Networking and Applications, AINA 2013, Barcelona, Spain, 25–28 March 2013, pp. 56–63. IEEE Computer Society (2013). https://doi.org/10.1109/AINA.2013.26
15. Manjanna, S., Hsieh, A., Dudek, G.: Scalable multi-robot system for non-myopic spatial sampling. CoRR **abs/2105.10018** (2021). https://arxiv.org/abs/2105.10018
16. Mo, Y., Beal, J., Dasgupta, S.: An aggregate computing approach to self-stabilizing leader election. In: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), Trento, Italy, 3–7 September 2018, pp. 112–117. IEEE (2018). https://doi.org/10.1109/FAS-W.2018.00034
17. Mousavi, H.K., Sun, Q., Motee, N.: Space-time sampling for network observability. CoRR **abs/1811.01303** (2018). http://arxiv.org/abs/1811.01303
18. Nielsen, M., Plotkin, G.D., Winskel, G.: Petri nets, event structures and domains, part I. Theor. Comput. Sci. **13**, 85–108 (1981). https://doi.org/10.1016/0304-3975(81)90112-2
19. Pianini, D., Beal, J., Viroli, M.: Improving gossip dynamics through overlapping replicates. In: Lluch Lafuente, A., Proença, J. (eds.) COORDINATION 2016. LNCS, vol. 9686, pp. 192–207. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39519-7_12
20. Pianini, D., Casadei, R., Viroli, M., Mariani, S., Zambonelli, F.: Time-fluid field-based coordination through programmable distributed schedulers. Log. Methods Comput. Sci. **17**(4) (2021). https://doi.org/10.46298/lmcs-17(4:13)2021
21. Pianini, D., Casadei, R., Viroli, M., Natali, A.: Partitioned integration and coordination via the self-organising coordination regions pattern. Future Gener. Comput. Syst. **114**, 44–68 (2021). https://doi.org/10.1016/j.future.2020.07.032
22. Pianini, D., Montagna, S., Viroli, M.: Chemical-oriented simulation of computational systems with ALCHEMIST. J. Simul. **7**(3), 202–215 (2013). https://doi.org/10.1057/jos.2012.27
23. Pianini, D., Viroli, M., Beal, J.: Protelis: practical aggregate programming. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 13–17 April 2015, pp. 1846–1853 (2015). https://doi.org/10.1145/2695664.2695913
24. Pianini, D., WhiteSource Renovate: Danysk/experiment-2022-coordination-space-fluid: 0.5.0-dev08+67e7add (2022). https://doi.org/10.5281/ZENODO.6473292
25. Rahimi, M.H., Hansen, M.H., Kaiser, W.J., Sukhatme, G.S., Estrin, D.: Adaptive sampling for environmental field estimation using robotic sensors. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, 2–6 August 2005, pp. 3692–3698. IEEE (2005). https://doi.org/10.1109/IROS.2005.1545070
26. Szczytowski, P., Khelil, A., Suri, N.: Asample: adaptive spatial sampling in wireless sensor networks. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, SUTC 2010 and IEEE International Workshop on Ubiquitous and Mobile Computing, UMC 2010, Newport Beach, California, USA, 7–9 June 2010, pp. 35–42. IEEE Computer Society (2010). https://doi.org/10.1109/SUTC.2010.37
27. Thompson, S.K.: Adaptive cluster sampling. J. Am. Stat. Assoc. **85**(412), 1050–1059 (1990). https://doi.org/10.1080/01621459.1990.10474975. https://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10474975

28. Viroli, M., Audrito, G., Beal, J., Damiani, F., Pianini, D.: Engineering resilient collective adaptive systems by self-stabilisation. ACM Trans. Model. Comput. Simul. **28**(2), 1–28 (2018). https://doi.org/10.1145/3177774
29. Viroli, M., Beal, J., Damiani, F., Audrito, G., Casadei, R., Pianini, D.: From distributed coordination to field calculus and aggregate computing. J. Log. Algebraic Methods Program. **109**, 100486 (2019). https://doi.org/10.1016/j.jlamp.2019.100486
30. Virrankoski, R., Savvides, A.: TASC: topology adaptive spatial clustering for sensor networks. In: IEEE 2nd International Conference on Mobile Adhoc and Sensor Systems, MASS 2005, The City Center Hotel, Washington, USA, 7–10 November 2005, p. 10. IEEE Computer Society (2005). https://doi.org/10.1109/MAHSS.2005.1542850
31. Wu, F., Kao, Y., Tseng, Y.: From wireless sensor networks towards cyber physical systems. Pervasive Mob. Comput. **7**(4), 397–413 (2011). https://doi.org/10.1016/j.pmcj.2011.03.003
32. Yao, J.T., Vasilakos, A.V., Pedrycz, W.: Granular computing: perspectives and challenges. IEEE Trans. Cybern. **43**(6), 1977–1989 (2013). https://doi.org/10.1109/TSMCC.2012.2236648