FFI-rapport 2015/00491

Exploitation of spatio-temporal correlations of traffic in ad hoc wireless networks

Reinert Korsnes and Mariann Hauge

FFI Forsvarets forskningsinstitutt

Exploitation of spatio-temporal correlations of traffic in ad hoc wireless networks

Reinert Korsnes and Mariann Hauge

Norwegian Defence Research Establishment (FFI)

11 November 2015

FFI-rapport 2015/00491

1249

P: ISBN 978-82-464-2610-5
E: ISBN 978-82-464-2611-2

Keywords

Kommunikasjonsnettverk

Kommunikasjonsprotokoller

Trådløs kommunikasjon

Taktisk informasjonsnett

Approved by

Jan Erik Voldhaug	Project Manager
Anders Eggen	Director

English summary

This report contributes to work on improving stability and capacity of military mobile ad hoc wireless networks (MANET). This improvement is necessary in order to provide the flexibility, capacity and robustness that are needed to support high maturity levels of NATO NEC (Network Enabled Capability). The report elaborates on search methods for available link capacities in such networks based on sparse measurements of local traffic load and information of the location of nodes.

Wireless communication produces electromagnetic radiation that may indicate the level of local traffic load. A stochastic (scalar) field Z can formally represent the intensity of local traffic load. Access to estimates of such fields can therefore help to optimize the routing function and improve network availability for mission critical information. The field Z will normally exhibit spatial and temporal correlations enabling reconstruction of the value of the field from sparse measurements. A contribution in this report is the formulation, implementation and illustration of a method to reconstruct time varying stochastic fields from sparse measurement samples. The cross-correlation of the field decreases gradually with increased distance in time and space.

The report also describes a scalable method for simulation of the field under question. This method can be used to simplify simulation of the reconstruction methods in a network simulator. This work may be relevant for activities studying if effects from traffic-generating events generate secondary events (cascades) that must be taken into account by the routing protocol. The proposed techniques in this report may be used for development of simple methods to exploit correlations in data traffic load within a network. The report argues that simple approaches often have more general relevance as compared to complex methods when data and experience are scarce.

Sammendrag

Denne rapporten presenterer arbeid som har potensial til å kunne forbedre stabilitet og kapasitet til militære ad hoc trådløse mobile kommunikasjonsnettverk (MANET). Slik forbedring er nødvendig for at de militære nettverkene skal ha den fleksibilitet, kapasitet og robusthet som er påkrevet for å kunne støtte høye modenhetsgrader av NbF (Nettverksbasert Forsvar). Rapporten gjennomgår muligheter for søk etter tilgjengelige lokale transportkapasiteter i slike nettverk basert på spredte målinger og informasjon om posisjonen til nodene. Et hovedpoeng er at rutingprotokoller i nettverket skal kunne bruke informasjon om nettverkslasten til assistanse for å finne gode transportveier for data i nettverket samt til overvåking, og for å gjøre tiltak for utbedringer. Nytten av informasjonen avhenger av presisjon og tidspunkt for slik tilgang samt hvordan rutingprotokollene er bygd opp.

Et stokastisk (skalar) felt Z kan representere hvor opptatt de lokale transportkapasitetene er i et område ved et gitt tidspunkt. Målinger av elektromagnetisk stråling på aktuelle frekvenser kan gi indikasjoner på slik lokal intensitet i datatrafikken. Feltet Z vil kunne forventes å være korrelert i både rom og tid. Det er da generelt komprimerbart og kan rekonstrueres ut fra spredte målinger. Dette kan gjøres ved hjelp av kriging-teknikker eller metoder som kategoriseres som komprimert måling der komprimeringen er en del av selve sampleprosessen. Et bidrag i denne rapporten er formulering, implementering og illustrasjon av en metode som viser hvordan tidsvarierende stokastiske felter effektivt kan rekonstrueres ut fra spredte målinger. Krysskorrelasjonen i feltet avtar gradvis med økt avstand i rom og tid.

Rapporten presenterer også en skalerbar metode for å simulere det elektromagnetiske feltet. Denne metoden kan bli brukt i en nettverkssimulator for å redusere kompleksiteten ved simulering av de foreslåtte metodene for rekonstruksjon av slike felter. Dette kan ha relevans for å studere effekter av at trafikkgenererende hendelser kan utløse sekundære hendelser som er viktige å ta hensyn til for ruteprotokollen. De foreslåtte metodene er av interesse for utvikling av enkle metoder for å utnytte korrelasjoner i datatrafikk. Rapporten argumenterer for at enkle metoder ofte kan være mer generelt relevante sammenlignet med komplekse metoder når en har lite tilgang på realistiske data.

Contents

1 Introduction

Tactical military communication systems typically include mobile *ad hoc* networks (MANET). A MANET is a self-configuring network of mobile routers (and associated hosts) connected by wireless links. The routers are free to move randomly and organize themselves arbitrarily. The network's wireless topology may therefore change rapidly and unpredictably. A MANET may operate in a standalone fashion, or it may be connected to other parts of the military networks, or to the Internet.

The ability of a MANET to establish connections in an ad hoc fashion in areas without any pre-planned infrastructure, and to support mobile users, makes it very useful for military operations and for search and rescue teams in areas without local infrastructure. This flexibility, adaptability and support for mobility comes with the expense of reduced network stability and capacity, and complex network management.

Traditionally, routing protocols for MANET search for the minimum number of hops between the source and the destination. Both Optimized Link State Routing (OSLR) [13] and Ad hoc On-Demand Distance Vector (AODV) routing [43], the two most cited routing protocols for MANET, use shortest paths as their routing metric. A shortest path metric does not take traffic load, link characteristics or other path quality characteristics into account when calculating the route. Hence a shortest path metric is neither able to support load balancing in the network, nor to support specific requirements for quality of service (QoS) (e.g., minimum data rate, transmission delay, jitter). This is a recognized problem related to mobile *ad hoc* networks, and many proposals call for improvements.

The proposals, which aim to improve MANET QoS tend to address the routing metric. This is an algorithm a routing protocol applies to make routing decisions. The most common routing metric is the shortest path metric as mentioned above and the minimum cost (as used by Open Shortest Path Forwarding (OSPF) [42] routing). Larsen [34] gives an introduction to various metrics and different classes of metrics. The next version of the OLSR [49] protocol will be prepared to use a range of different additive routing metrics.

In the discussion in this paper, the focus is on metrics used to describe available data capacity and the link and path capacity. Some popular methods for this purpose are briefly described below.

Several approaches attempt to include data from measurements of link capacities when calculating routes. Per-hop Round Trip Time (RTT) [1] measures the round trip delay seen by unicast active probes between each neighbor pair. An exponentially weighted moving average of the probe samples are used in the metric. Per-hop Packet Pair Delay (PktPair) [30] measures the delay between a pair of active probe packets immediately following each other. The first probe is small and the second is large. The neighbor calculates the delay between the receipt of the first and the second probe. An exponentially weighted moving average is used also for these measurements. Expected Transmission Count (ETX) [15] is a third variant that use active probes to estimate the channel quality and the load on the link. This metric estimates the number of retransmissions needed to send unicast packets by measuring the loss rate of broadcast probes between pairs of neighboring nodes. These three metrics

are compared in [21] and show that the ETX metric perform best in their static network scenario. They also make another observation that is important to keep in mind; when the source is mobile, none of the smart metrics are able to outperform the basic shortest hop count metric.

Access to cross layer information in the network may facilitate improved performance in *ad hoc* mobile networks. Such information can be in the form of data on signal to noise ratio (SNR) at the receiver side for each link. However, measurements of SNR may give imprecise prognoses for packet loss [33]. Data measurement on local radio load can similarly contribute to estimates facilitating improved routing [50]. The radio load parameter is a measure for how busy the medium around a node is. The present work elaborates on this idea.

A main classification of routing protocols for MANET are the two classes of proactive and reactive protocols. Proactive protocols are similar to the common Internet protocols. They keep state of routes from all to all at all times. OSLR [13] is such a protocol. Reactive protocols conversely search for a route on demand when a source needs to send data to a particular destination. AODV [43] is an example of a reactive protocol.

Most proposals for QoS routing protocols are based on reactive mechanisms. There are two main reasons for that. With a reactive scheme, it is possible to specify one QoS requirement for one route search and different requirements for the next route search. With a proactive scheme, maintaining many different paths between all sources and destinations is complex and costly. With a reactive scheme the path decisions are made with recent and thus accurate resource measurements. In a proactive scheme, a path is always calculated based on resource measurements that have aged somewhat. One recent protocol that considers both QoS and policy decisions are described in [28]. [26] and [4] are survey papers that cover reactive QoS protocols well.

Proactive protocols are often the preferred choice for routing in military networks since these networks often need quick set up times for voice calls and other real time applications. Proactive protocols also provide more predictable signaling overhead. Some research on QoS routing has also been done for proactive schemes. Several metrics have been suggested to support QoS for OSLRv2 [49]. OLSR-ETX [25] is one example which uses ETX measurements for stable route establishment. OSPF with support for Multiple Topologies OSPF-MT [27] has also been studied to support multiple QoS paths in a proactive network. It is able to maintain several different paths between a source and destination. These protocols require metrics measurements on all nodes in the network and general distribution of the resulting data.

In the present work, we assume that there are some correlation both in space and time for the traffic load, and thus link capacity in the network. Given this correlation, it is possible to reduce the number of measurement samples needed to reconstruct a complete picture of the traffic load in the whole network, or selected parts of the network. We use simple approaches to collect timely traffic load information to improve QoS by exploiting spatio-temporal correlations or predictability within the data. The phrase “simple” is here supposed to guide selection of system design (models) in situations of limited data (experience) which presumably are consistent with many possible

models. Simplicity helps to minimize bias from conflicting unforeseen requirements. The principle of simplicity conforms to common sense and the old principle of Occam's razor. It also acts according to recent results on inference maximizing entropy [46] which may contribute to formal procedures within system analysis.

To our knowledge, current routing protocols do not take advantage of the mentioned correlation and heuristic relation in the link quality data. We suggest directions for research that might be interesting to pursue in order to exploit this for QoS routing and load balancing purposes.

Appendix B describes an algorithm to simulate a time varying stochastic field which is assumed to represent local intensity of traffic load. The cross-correlation in this field increases with decreased distance in time and space. The presented illustrations show situations where resent temporally and spatially sparse measurements at few nodes, provide sufficient information to reconstruct the whole current field. A challenge for network protocol design is in this case to demonstrate, with minimal additional complexity, the ability to exploit such situations to improve QoS.

The remainder of this report describes different techniques for sampling, sample positions and reconstruction of the sampled filed. It is left for future work to study how the presented techniques best can be exploited in routing protocol metrics.

2 Sample strategies for traffic monitoring

Assume a wireless network operating in a restricted area. Figure 2.1 illustrates an idealized spatially and temporally correlated distribution of the intensity of radio transmission within the area. This

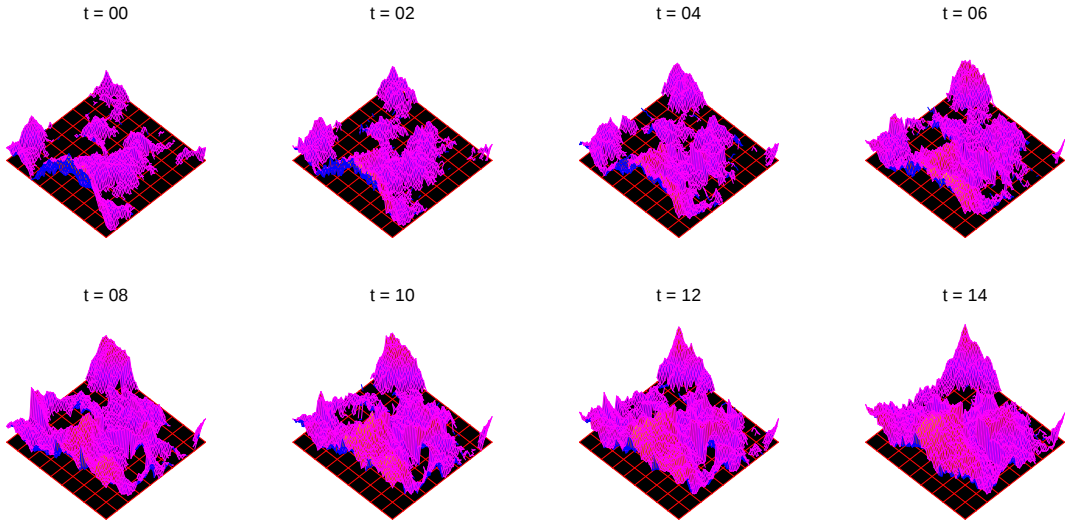

Figure 2.1 Simulated spatial distribution of intensity of radio communication in a square area of length 4000 m for different points in time (t).

spatial-temporal distribution results from generation of fractal landscapes which may reflect cascades

of traffic generating events taking place in the actual area [37]. Appendix B describes the algorithm applied to produce the distribution in Figure 2.1. Fractal-like distributions of local traffic intensity may result from a process where an event can generate secondary events and so on. This type of distributions may serve as test examples for development of network routing protocols exploiting correlations between available capacities of geographically close data links. Quick and sparse sampling may in this case provide more up to date information with little signaling overhead as compared to slower and denser sampling.

Routing protocols may utilize complete reconstruction of the scalar field representing intensities of local traffic load to optimize routing. Such estimates can contribute to improvements of calculations of link costs which current QoS routing protocols typically use to direct data packets. Compressive sensing (Appendix A) can provide such reconstruction from greedy non-adaptive sampling of the field. Reconstruction may also come from direct interpolation/extrapolation based on sparse randomized samples within subsets of the network. Local conditional reporting (LCR) of measurements at nodes may in this case bring cost effectiveness into this sampling strategy. A node may for example only report measurements satisfying given (local) conditions for which it hosts data. These data can be produced directly from local measurements of electromagnetic radiation and previous history of its own reports. The content of received data packets from neighboring nodes may also affect local decisions to report data.

Note that LCR is a generalization of thresholding where nodes only report samples reflecting for example traffic intensity below a given threshold assuming those not reporting measure intensities above this threshold (the conjugate set). Control mechanisms (local or signaled) may in this case make the nodes change (switch) this reporting rule to reduce the number of reports without loosing information.

Section 4 below explores kriging to estimate connected regions with low traffic intensity. Adaptive sampling to identify connected regions for easy data transport may in general provide more greedy sampling. A possible adaptive sampling strategy is identification of linear borders for easy data transport. A routing protocol may in this case explore routes on different sides of these borders. Note that dynamic control typically includes a trade-off between exploitation and exploration. Exploitation means to utilize current information whereas exploration includes search for new information via for example randomized search.

The present work focuses on concepts for efficient search for connected regions with high transport capacities. Section 5 outlines diffusion limited aggregation (DLA), which is a self-organized growth process based solely on randomized simple interactions between nodes. Neighboring nodes in *ad hoc* wireless networks normally interact frequently to advertise presence. Hence there can be little extra cost including DLA processes in this data exchange.

Note that reconstruction of the whole field of traffic intensities and correlations may provide data for different metrics in the routing function. Hence it is not obvious that the most greedy and fast sampling approach is superior to slower complete reconstructions. Also note that application

of kriging techniques includes estimation of models for spatial(-temporal) correlation of traffic intensities. Data on such correlation may have interest as assistance for routing decisions.

3 Rapid accumulation and distribution of network traffic information

Assume an *ad hoc* network that samples correlated data traffic to optimize routing decisions. Section 4 below illustrates that only a minor fraction of the nodes may need to take part in this data harvesting in order to produce estimates of traffic intensities in the whole network. This subset of sampling nodes behave similar to a collection tree for a sensor network for timely accumulation and distribution of information on available link capacities. The sampling nodes may be chosen randomly in the network or they can be concentrated in areas of low local traffic load assuming the selection of measurement nodes is done distributed, based on locally acquired data. An even more greedy strategy is to concentrate the sampling in "grey zones" of measurements values classified to be between "high" and "low" threshold values (to partition the network into calm and busy areas of interest for efficient routing). A node may alternatively be classified to belong to a "grey zone" if it measures a "high" value at the same time as close neighbors measure "low" values. This approach is similar to systematic spatial thresholding within image analysis [38].

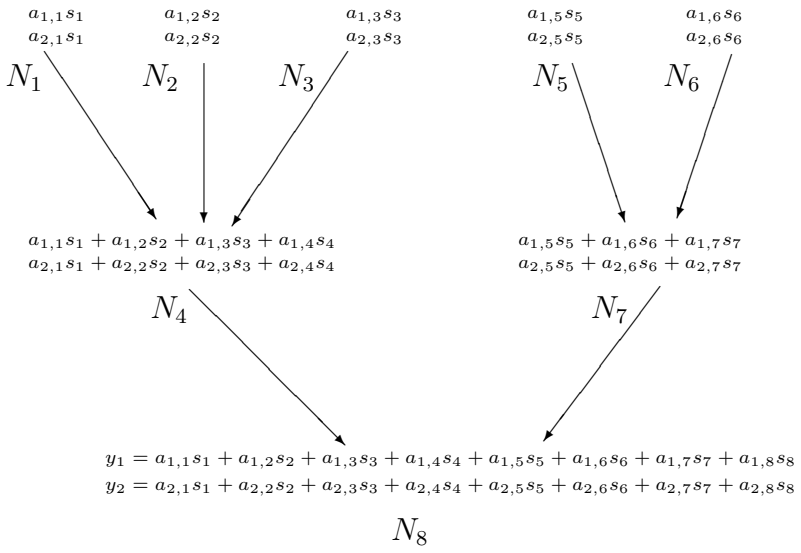

Figure 3.1 Compressive sensing is here realized with an accumulation tree performing linear measurements.

Figure 3.1 illustrates a possible implementation of compressive sensing to reconstruct correlated measurements from a subset of nodes (see Appendix A).The nodes $N_1, N_2, \ldots, N_8$ here respectively produce the samples $s_1, s_2, \ldots, s_8$ and node N_8 obtain the linear measurements y_1 and y_2.

Korsnes [31] illustrates potentials for simple and rapid organization of (suboptimal) collection trees in (Euclidean) weighted network close to their percolation threshold. Signaling in networks which are

close to their percolation threshold, can exhibit an universal phenomena known from statistical physics and which may facilitate understanding and exploitation of the design of routing protocols [29]. A variety of contributions have pointed out how signaling networks can relate to critical phenomena, spatial-temporal synchronization and correlation length [5, 23, 45, 44, 35, 41, 53, 54, 3, 24]. Fast signaling between subsets of nodes in a network may utilize such regularity to minimize cost of data signaling.

Note that estimates of the field $\boldsymbol{Z}$ representing local network traffic load, can be used to affect traffic patterns in the actual network even if each node does not have direct access to the estimates. For example, unmanned autonomous vehicles (UAVs) at elevated positions may be available to affect traffic patterns acting as relay or providing input to routing protocols. Such UAVs may directly sample the field or receive information from key nodes in the network. Figure 3.2 illustrates such a situation where a swarm of five UAVs locates and positions themselves at the maxima of a field which they measure. The swarm samples in an exploitative way by moving in the direction of the local gradient of the field. Other UAVs perform exploration by sampling at random positions. This combination of sampling increase the possibility to locate maximas including the global maxima. The UAVs in this case (in an internal collaborative way) position themselves autonomous and independent of the ground relays. Note that signaling between the actual ground relays and the UAVs may provide improved and faster positioning.

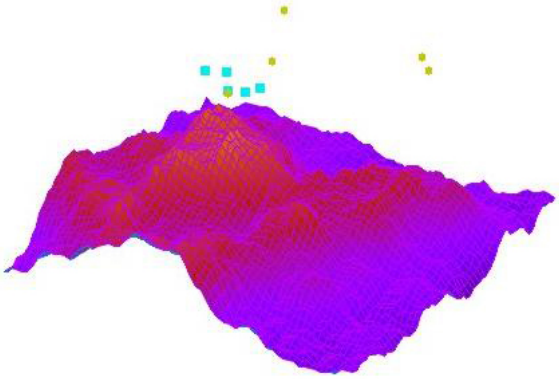

Figure 3.2 Illustration of a swarm of UAVs locating the maxima of a field assumed to represent local electromagnetic radiation in a radio based communication network.

4 Kriging

4.1 Spatial and spatio-temporal kriging

Kriging is a set of techniques to construct a (scalar) field Z based on sparse sampling of it [2, 14, 32, 36, 39, 40]. Its first applications were within geostatics to interpolate or predict values of static random fields at not observed localities. Later developments apply kriging to interpolate dynamic time dependent fields [11, 12, 48]. Both spatial and space-time kriging are based on models of the fields covariance as a function of distance (in time and space). Let $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ respectively represent

two spatial positions at a given time.

Assuming spatial isotropy and shift invariance, the correlation $C_{\mathrm{s}}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ between $Z(\boldsymbol{r}_1)$ and $Z(\boldsymbol{r}_2)$ has the form:

$$C_{\mathrm{s}}(\boldsymbol{r}_1, \boldsymbol{r}_2) = C_{\mathrm{s}}(r) \tag{4.1}$$

where r is the distance between $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. The temporal covariance $C_{\mathrm{t}}(t_1, t_2)$ similarly has the form (assuming stationarity):

$$C_{\mathrm{t}}(t_1, t_2) = C_{\mathrm{t}}(t) \tag{4.2}$$

where $t = t_2 - t_1$.

Spatio-temporal kriging is applied as a model for correlation of field values at different positions and times. A common assumption for this spatio-temporal covariance function is separability. This means that the correlation $C_{\mathrm{s,t}}(\boldsymbol{r}_1, t_1, \boldsymbol{r}_2, t_2)$ between the field at the position $\boldsymbol{r}_1$ and time t_1 ($Z(\boldsymbol{r}_1, t_1)$ and at the position $\boldsymbol{r}_2$ and time t_2 ($Z(\boldsymbol{r}_2, t_2)$ is separable:

$$C_{\mathrm{s,t}}(\boldsymbol{r}_1, t_1, \boldsymbol{r}_2, t_2) = C_{\mathrm{s}}(\boldsymbol{r}_1, \boldsymbol{r}_2)\, C_{\mathrm{t}}(t_1, t_2) \tag{4.3}$$

Substituting from Equations 4.1 and 4.2 gives:

$$C_{\mathrm{s,t}}(\boldsymbol{r}_1, t_1, \boldsymbol{r}_2, t_2) = C_{\mathrm{s}}(r)\, C_{\mathrm{t}}(t) \tag{4.4}$$

These assumptions enable simple calculation of covariances.

4.2 Spatial kriging

The outline below describes a simple application of spatial kriging and illustrates its data requirements. The spatial kriging produces estimates of a stochastic field in the neighborhood of nodes that have provided local measurements of it (sufficiently close in time). The stochastic field represents for example an indicator of for example locally available link capacities within an *ad hoc* network at a given point in time.

Assume that $I_k, k = 1, 2, \ldots, n$, denote the measured intensity at neighboring sampling nodes of node $\boldsymbol{N}$. Kriging types of interpolation means to estimate linear predictors of the intensity I at the node $\boldsymbol{N}$.

$$I = V_1 I_1 + V_2 I_2 + \ldots + V_n I_n \tag{4.5}$$

Let $\langle \cdot, \cdot \rangle$ denote the covariance operator between two stochastic variables. Simple least square minimization then gives values of $\boldsymbol{V} = (V_1, V_2, \ldots, V_n)^T$:

$$\boldsymbol{V} = \boldsymbol{G}^{-1} \boldsymbol{Y} \tag{4.6}$$

where

$$\boldsymbol{G} = \begin{bmatrix} \langle I_1, I_1 \rangle & \langle I_1, I_2 \rangle & \ldots & \langle I_1, I_n \rangle \\ \langle I_2, I_1 \rangle & \langle I_2, I_2 \rangle & \ldots & \langle I_2, I_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle I_n, I_1 \rangle & \langle I_n, I_2 \rangle & \ldots & \langle I_n, I_n \rangle \end{bmatrix} \tag{4.7}$$

is the common Gram matrix, and

$$\boldsymbol{Y} = [\langle I_1, I\rangle, \langle I_2, I\rangle, \ldots, \langle I_n, I\rangle]^T \tag{4.8}$$

Assuming the stochastic field is stationary (and ergodic), it can be derived from one realization of the field. The covariance $\langle I(a), I(b)\rangle$ between the intensities $I(a)$ and $I(b)$ at two locations a and b, are in the present work assumed to depend solely on the distance between these locations. This function (of distance) is derived from the semivariogram via the formula:

$$2\gamma(a,b) = E(|I(a) - I(b)|^2) = E(I(a)^2) - 2\langle I(a), I(b)\rangle + E(I(b)^2) \tag{4.9}$$

where $E(\cdot)$ is the expectation operator: Assuming, from stationarity, $\langle I(a), I(a)\rangle = \langle I(b), I(b)\rangle$, Equation 4.9 gives:

$$\langle I(a), I(b)\rangle = \langle I(a), I(a)\rangle - \gamma(a,b) \tag{4.10}$$

Assume each node with a given probability p participates to collect data from the measurements of the field of local intensity of data traffic. The resulting set of data represents the whole field and hence this procedure includes data compression for minimization of data transport (reduced signaling overhead).

Figure 4.1 shows the reconstruction of the field of 2.1 based on 180 and 400 sample points. Figure 4.2

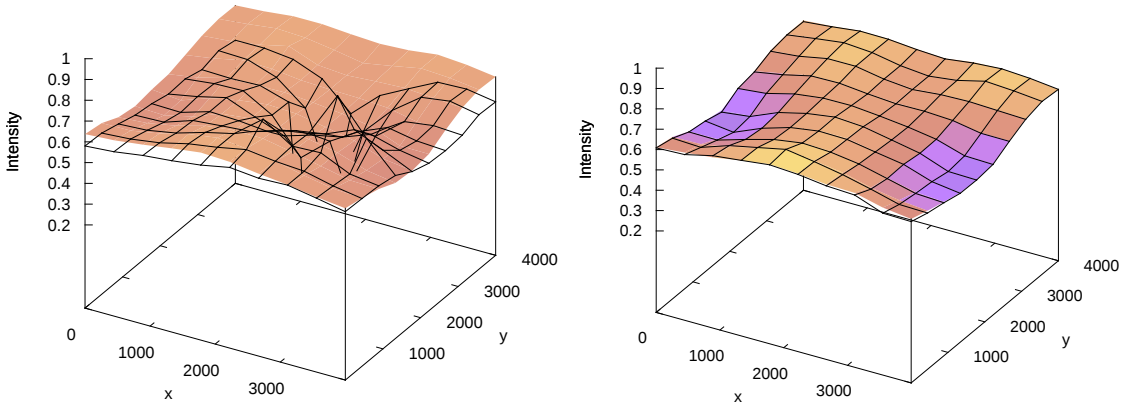

Figure 4.1 Spatial kriging. Left: Reconstruction of random field based on 180 random sample points. Right: similar reconstruction based om 400 random sample points. Note that the former example shows poor reconstruction.

indicates that the prediction error fall off rapidly from 100 to 200 samples and seems asymptotically to approach zero for the number of samples larger than 300.

4.3 Spatio-temporal kriging

Figure 4.3 shows estimates of semivariograms for simulated spatio-temporal fields (see Figure 2.1 and Appendix B). Figure 4.4 illustrates 1000 samples of the field at random times and locations. The

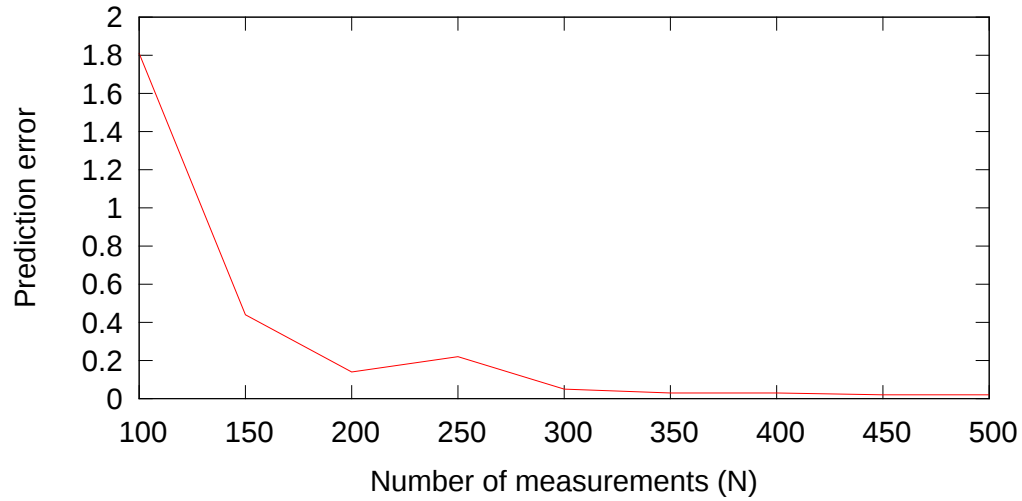

Figure 4.2 Kriging prediction error as a function number of independent random measurements (N). The error exhibit asymptotic behavior for $N > 300$.

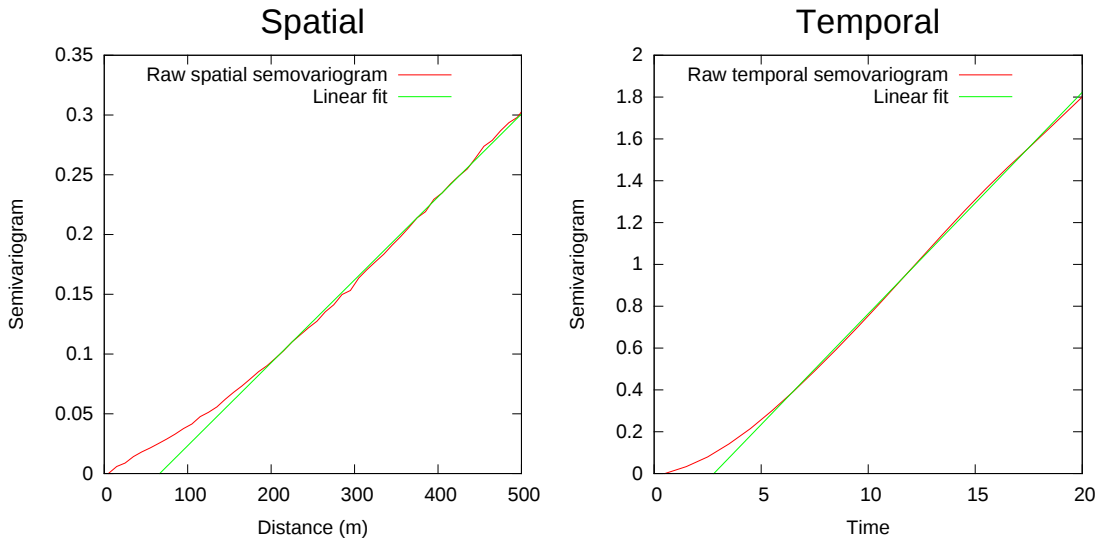

Figure 4.3 Spatial and temporal semivariogram for simulated field for $t = 1, 2, \ldots, 64$.

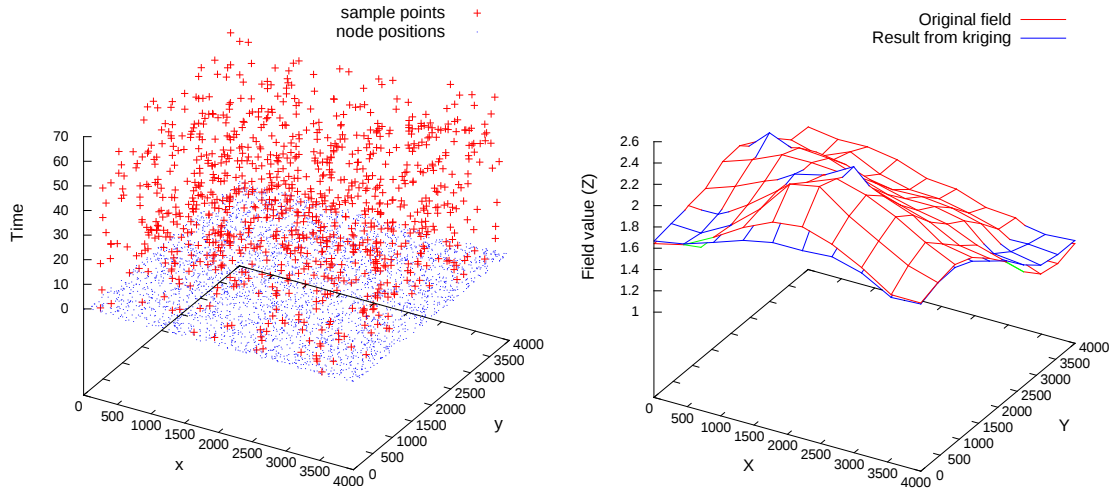

Figure 4.4 Left: 1000 samples of the field at random times and locations. Right: result from estimating the field from 72 samples with 10 prior time steps.

figure also illustrates the result from spatio-temporal kriging for the field at a given instant of time (time point 60). The kriging procedure in this case utilizes data from 10 time points previous to the actual current time for the field (in total 72 samples). This technique seems not to well reproduce the field for substantially less density of samples than the given example.

5 Diffusion-limited aggregation

Diffusion-limited aggregation (DLA) is a process where particles randomly move and cluster together in aggregates. When a moving particle hits an aggregate, it sticks to it by a given probability [52]. The aggregates are known as Brownian trees which are fractals of dimension about 1.71 in 2-D for unrestricted random motion. Such aggregation is a common model for growth of fractal trees for which the below arguments apply.

Virtual particles can similarly to physical ones randomly move within a graph $\boldsymbol{G}$ and form a fractal (Brownian) tree $\boldsymbol{T}$. Such random walkers can be data packets starting at random nodes and follow edges (links) between vertices (nodes). When the particle hits a cluster after moving from a vertex V via the edge E, then (V, E) joins the cluster. Assume a given node acts as a starting point of such cluster formation. This cluster constitutes a collection tree where E has the direction of the final step of the particle (data packet).

Figure 5.1 illustrates a simulated example of pathways from leaf nodes within a Brownian three including all nodes of an underlying network $\boldsymbol{G}$. The number of nodes in $\boldsymbol{G}$ is in this case 4000 with random positions within a square of side length 4000 m. Each node has links to nodes within the range of 150 m. Two special cases for construction of Brownian trees are shown in the figure. In the left picture, nodes are allowed to stick to the graph only if they satisfy the local condition that the

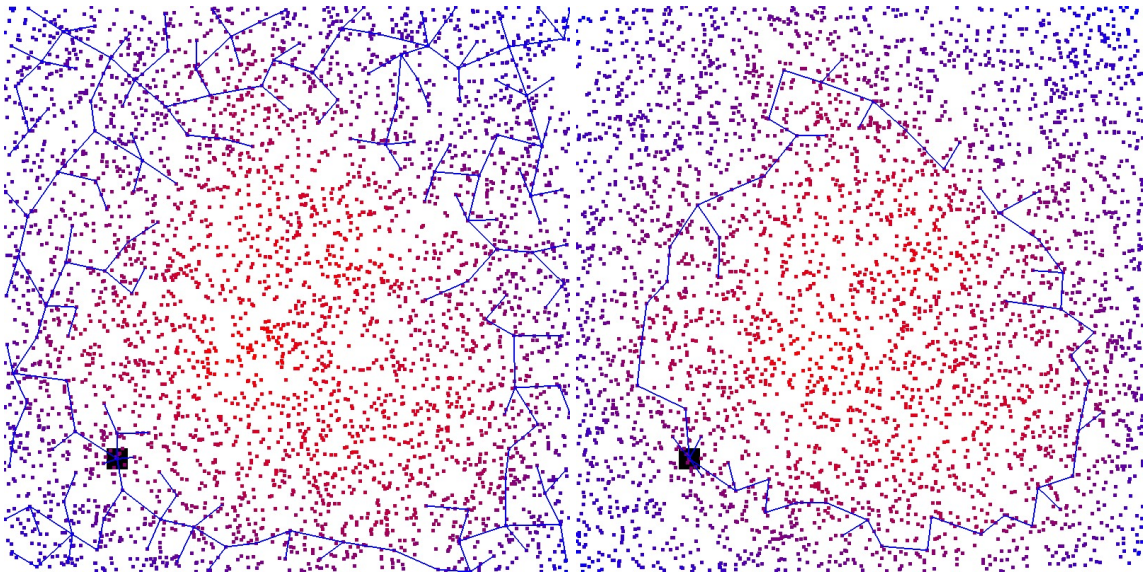

Figure 5.1 Left: Brownian tree with 170 nodes concentrating on areas of low values of random field (blue). Right: Similar Brownian tree sampling at border nodes (55 nodes).

local field is below a threshold value. In the right picture, the nodes must satisfy the condition of being in a “grey zone” i.e. a node belongs to a “grey zone” if it measures a “high” value at the same time as close neighbors measure “low” values (or the other way around). These cases show how the Brownian trees can be tailored to provide sampling points that are reduced to a minimum and at the same time give sufficient information about a specific characteristics of the field in question.

Figure 5.2 shows the reconstruction of the field of Figure 2.1 based on 150 and 400 sample points respectively. The sample-points are the nodes that are traversed as a Brownian tree is grown.

6 Conclusions

Methods for improvements of QoS in a network may benefit from identification of areas with low traffic intensity. Assume a MANET where data packets compete to be forwarded as fast as possible between nodes along their routes. Data packets will tend to move significantly faster (and with less probability of packet loss) in areas with little local data traffic as compared to areas with high traffic load.

This report has explored some techniques that can be useful in order to simulate a stochastic field, sample, compress and reconstruct the field in a network in a timely manner and with sparse sampling points. The goal is for each node in the network to acquire real-time knowledge of the intensity of the electromagnetic filed of the network (traffic load) and to use this information to aid routing decisions.

The present work applies a very high number of nodes in the examples (4000 nodes). The high number of nodes was selected in order to do a first evaluation of the applicability of using the different

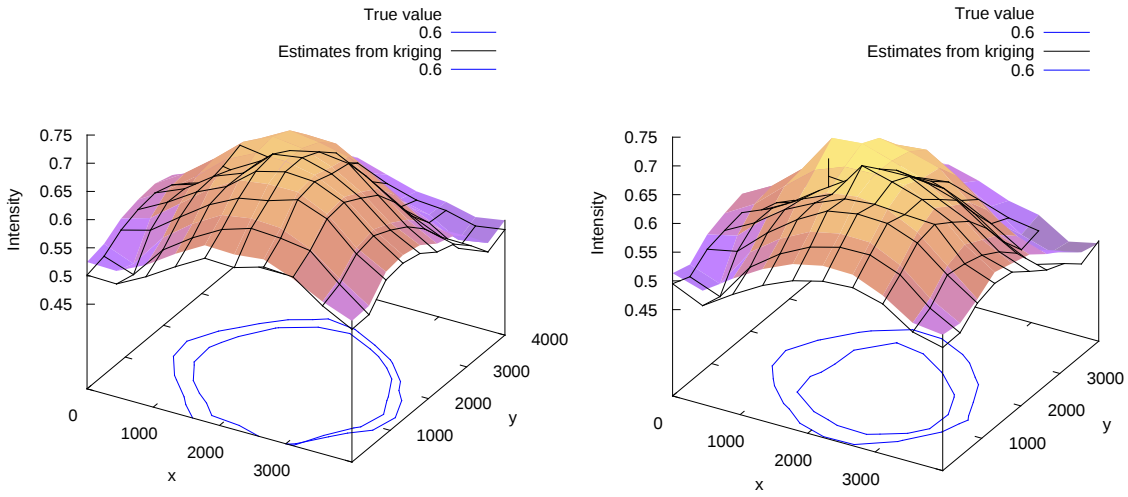

Figure 5.2 Left: Reconstruction of random field based on 150 Brownian tree sample points. Right: similar reconstruction based om 400 random sample points. Note that the former example show poor reconstruction at the border.

techniques to improve routing decisions. In future work we will study if the techniques are valid also for networks of fewer nodes. It is also left for future work to study how well a routing protocol can utilize the information that can be provided by the described techniques.

References

[1] Atul Adya, Paramvir Bahl, Jitendra Padhye, Alec Wolman, and Lidong Zhou. A multi-radio unification protocol for ieee 802.11 wireless networks. In *Proceedings of the First International Conference on Broadband Networks*, BROADNETS '04, pages 344–354, Washington, DC, USA, 2004. IEEE Computer Society.

[2] F.P. Agterberg. *Geomathematics: Mathematical Background and Geo-Science Applications*. Developments in Geotectonics. Elsevier, 1974.

[3] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente. Synchronization processes in complex networks. *Physica D*, 224:27–34, 2006.

[4] R. Asokan. A review of quality of service (qos) routing protocols for mobile ad hoc networks. In *Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on*, pages 1–6, 2010.

[5] Ginestra Bianconi. Mean field solution of the Ising model on a Barabási–Albert network. *Physics Letters A*, 303(2–3):166–168, 2002.

[6] Thomas Blumensath. *The Geometry of Compressed Sensing*, chapter 2. Springer, 2013.

[7] E.J. Candes and T. Tao. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[8] E.J. Candès and M.B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[9] Emmanuel Candes and Justin Romberg. Sparsity and Incoherence in Compressive Sampling, November 2006.

[10] Emmanuel Candes, Justin Romberg, and Terence Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements, December 2005.

[11] L.De Cesare, D.E Myers, and D Posa. Estimating and modeling space–time correlation structures. *Statistics & Probability Letters*, 51(1):9–14, 2001.

[12] G. Christakos, P. Bogaert, and M.L. Serre. *Temporal GIS: Advanced Functions for Field-based Applications*. Geosciences online libray. Springer, 2001.

[13] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[14] Noel Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239, 1990.

[15] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.*, 11(4):419–434, July 2005.

[16] David L. Donoho. For Most Large Underdetermined Systems of Linear Equations the Minimal L1-norm Solution is also the Sparsest Solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[17] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[18] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[19] David L. Donoho, Adel Javanmard, and Andrea Montanari. Information-Theoretically Optimal Compressed Sensing via Spatial Coupling and Approximate Message Passing. *CoRR*, abs/1112.0708, 2011.

[20] David L. Donoho and Jared Tanner. Counting faces of randomly-projected polytopes when the projection radically lowers dimension, September 2006.

[21] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(4):133–144, August 2004.

[22] Alain Fournier, Donald S. Fussell, and Loren C. Carpenter. Computer Rendering of Stochastic Models. *Commun. ACM*, 25(6):371–384, 1982.

[23] F. Ginelli, R. Livi, A. Politi, and A. Torcini. Relationship between directed percolation and the synchronization transition in spatially extended systems. *Phys. Rev. E*, 67:046217, Apr 2003.

[24] J. Gómez-Gardeñes, Y. Moreno, and A. Arenas. Paths to Synchronization on Complex Networks. *Phys. Rev. Lett.*, 98:034101, 2007.

[25] A. Kaplan H. Rogge, E. Baccelli. Packet Sequence Number based ETX Metric for Mobile Ad Hoc Networks. draft-funkfeuer-manet-olsrv2-etx-01 (work in progress), March 2010.

[26] L. Hanzo and R. Tafazolli. A survey of qos routing solutions for mobile ad hoc networks. *Communications Surveys Tutorials, IEEE*, 9(2):50–70, 2007.

[27] M. Hauge, M.A. Brose, J. Sander, and J. Andersson. Multi-topology routing for qos support in the consis convoy manet. In *Communications and Information Systems Conference (MCC), 2012 Military*, pages 1–8, 2012.

[28] M. Hauge, L. Landmark, E. Larsen, P. E. Engelstad, and O. Kure. Resilient internetwork routing with qos support over heterogeneous mobile military networks. In *STO-MP-IST-123 Symposium on Cognitive Radio and Future Networks*, 2014.

[29] Tetsuya Ishikawa and Tomohisa Hayakawa. Analysis of Critical Phenomenon on Gossip Protocol Using Back-Ultradiscritization. In Sophie Tarbouriech and Miroslav Krstic, editors, *Nonlinear Control Systems*, volume 9, pages 418–423, 2013.

[30] S. Keshav. A control-theoretic approach to flow control. *SIGCOMM Comput. Commun. Rev.*, 25(1):188–201, January 1995.

[31] Reinert Korsnes. Rapid self-organised initiation of ad hoc sensor networks close above the percolation threshold. *Physica A: Statistical Mechanics and its Applications*, 389(14):2841 – 2848, 2010.

[32] D.G. Krige. A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand. Master's thesis, University of the Witwatersrand, 1951.

[33] M. Lampe, H. Rohling, and W. Zirwas. Misunderstandings about link adaptation for frequency selective fading channels. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, volume 2, pages 710–714 vol.2, 2002.

[34] E. Larsen. Rutemetrikker i MANET. FFI-Rapport 2012/01318, Norwegian Defence Research Establishment, 2012.

[35] D.-S. Lee. Synchronization transition in scale-free networks: Clusters of synchrony. *Phys. Rev. E*, 72:026208, 2005.

[36] Link, R.F. and Koch, G.S. *Experimental Designs and Trend-Surface Analysis*. Ed. Plenum, New Your, 1970.

[37] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1982.

[38] Kanti V. Mardia and T. J. Hainsworth. A Spatial Thresholding Method for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):919–927, 1988.

[39] G. Matheron. *Principles of geostatistics*, volume 58. Society of Economic Geologists, 1963.

[40] G. Matheron. The Intrinsic Random Functions and Their Applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

[41] Adilson E. Motter, C. S. Zhou, and J. Kurths. Enhancing complex-network synchronization. *Europhys. Lett.*, 69(3):334–340, 2005.

[42] J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998. Updated by RFC 5709.

[43] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[44] A. Pikovsky, M. Rosenblum, J. Kurths, and S. H. Strogatz. Synchronization: A Universal Concept in Nonlinear Sciences. *Phys. Today*, 56:47–47, 2003.

[45] A. Pikovsky, M. G. Rosenblum, and J. Kurths. *Synchronization, A Universal Concept in Nonlinear Sciences*. Cambridge University Press, Cambridge, 2001.

[46] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A. Dill. Nonadditive Entropies Yield Probability Distributions with Biases not Warranted by the Data. *Phys. Rev. Lett.*, 111, December 2013.

[47] Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Science/Engineering/Math, May 1986.

[48] M. Serre. *Environmental Spatiotemporal Mapping and Ground Water Flow Modeling Using the BME and ST Methods*. PhD thesis, University of North Carolina at Chapel Hill, 1999.

[49] P. Jacquet T. Clausen, C. Dearlove and U. Herberg. The Optimized Link State Routing Protocol version 2. draft-ietf-manet-olsrv2-19 (work in progress), March 2013.

[50] X. Tao. *Traffic Balancing: A Method for Exploiting System Capacity in Wireless Ad Hoc Networks*. PhD thesis, Ottawa-Carleton Institute for Electrical and Computer Engineering, 2005.

[51] Garrett A. Warnell, Dikpal Reddy, and Rama Chellappa. Adaptive rate compressive sensing for background subtraction. In *ICASSP*, pages 1477–1480. IEEE, 2012.

[52] T.A. Witten, Jr and Leonard M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters*, 47(19):1400–1403, 1981.

[53] Changsong Zhou and Jürgen Kurths. Dynamical Weights and Enhanced Synchronization in Adaptive Complex Networks. *Physical Review Letters*, 96(16):164102, 2006.

[54] Changsong Zhou, Adilson E. Motter, and Jurgen Kurths. Universality in the synchronization of weighted random networks. *Phys. Rev. Lett.*, 96:034101, April 2006.

Appendix A Compressive sensing in brief

Assume the problem of reproducing a real vector $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]^T$ from linear measurements. The Hilbert space representation theorem sheds light on generalization and properness of this question. This theorem states that a linear functional L on a Hilbert space H can be identified as an inner product with a unique member $f \in H$ such that $L(\cdot) = \langle \cdot, f \rangle$ [47]. This means that for any spanning subset M of H, the set of "measurements" (or projections) $\{\langle \boldsymbol{g}, \boldsymbol{s} \rangle \,|\, \boldsymbol{g} \in M\}$ uniquely defines $\boldsymbol{s}$. Smaller sets of measurements are needed for recovery of $\boldsymbol{s}$ if the possible set of signals is restricted. The main idea behind compressive sensing [9, 7, 10, 17, 18, 16, 8] is to exploit such restrictions to minimize number of measurements needed for signal recovery. Blumensath [6] outlines various generalizations to understand its geometric aspects.

Assume an $n \times n$ matrix $\boldsymbol{B}$ with rows consisting of orthonormal basis vectors $\boldsymbol{\psi}_i, i = 1, \ldots, n$. B in this case represents a coordinate transformation from the standard basis in $\mathbb{R}^n$ and into the coordinate system given by $\boldsymbol{\psi}_i, i = 1, 2, \ldots n$. The signal $\boldsymbol{s}$ is considered to be sparse if $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{s}$ has at most k components different from zero. This concept of sparsity probably originates from common experience from wavelet and Fourier analysis of real life signals. The original idea of compressive sensing is to exploit sparsity to solve the otherwise under-determined linear inverse problem:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{A.1}$$

where $\boldsymbol{A}$ is an $m \times n$ sampling matrix typically with random entries and $k \leq m \ll n$. The vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T$ represents linear measurements of the (transformed) signal $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$:

$$y_i = \sum_{j=1}^{n} a_{i,j}\, x_j \tag{A.2}$$

where $a_{i,j}$ is the (i, j) entry of the matrix $\boldsymbol{A}$.

A computationally efficient estimate of x from Equation A.1 is [18]:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}}{\operatorname{argmin}} \|\boldsymbol{B}\boldsymbol{x}\|_{l_1} \tag{A.3}$$

If $m \geq C \cdot \mu^2(A, B) \cdot k \cdot \log n$, Equation A.1 has a unique solution [9, 20, 19].

The above outline of compressive sensing is a non-adaptive procedure for information collection. However, the approach may be applied in an adaptive way. The sparsity of for example a scalar field (or video) may gradually change during time. Hence the procedure of compressive sensing can adapt to this change [51]. Note that such temporal variation of sparsity may be partly predictable due to repetitiveness due to time of the day or modes of operation.

Appendix B Modified diamond-square algorithm

This section describes a modification of the diamond-square algorithm [22] to produce time varying fractal landscape. Figure B.1 illustrates the original diamond-square algorithm which fills up an array with height coordinate values. The resulting values form the height of a fractal "landscape" (or

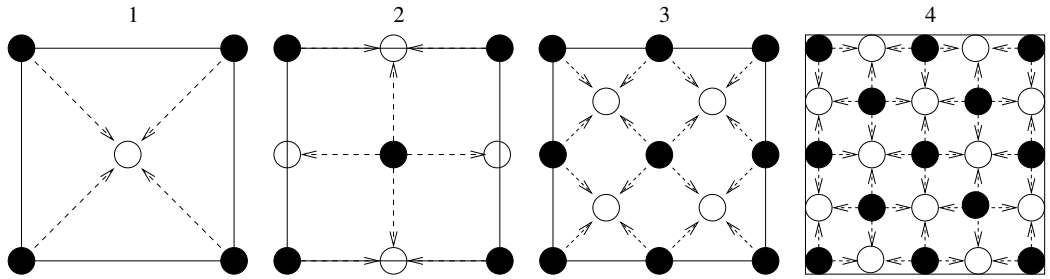

Figure B.1 Diamond-square algorithm:

stochastic field). The diamond-square algorithm works as follows (ref B.1) The four corners of an array are first given numbers. The algorithm then sets the middle point equal to the average value of the corners plus a random component (step 1). The "diamond corners" then gets the average value of nearest neighbor vertically plus a random component (step 2). The algorithm subsequently perform the similar operation of the four sub-squares of the original square and so on until the sub-squares are empty (step 3, 4 etc.).

Figure B.2 shows an example of generation of such a landscape.

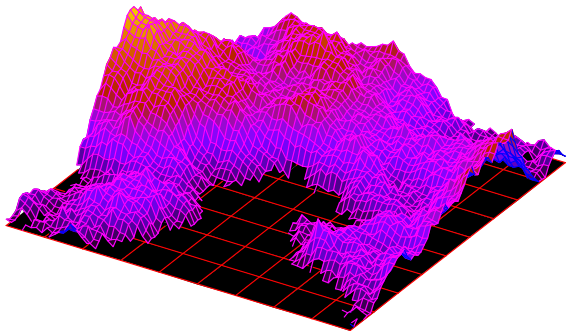

Figure B.2 Fractal landscape produced by diamond-square algorithm on an 65×65 *array after application of a lower (constant) cut-off value.*

The following Ada source listing implements the diamond-square algorithm. Note the simplicity of the Ada implementation.

```
with Ada.Numerics.Generic_Elementary_Functions;
with Ada.Numerics.Float_Random, Text_IO;
use  Ada.Numerics, Ada.Numerics.Float_Random, Text_IO;
procedure frac2d is

   subtype Real is Float;
```

```
package E_F      is new Ada.Numerics.Generic_Elementary_Functions (Real);
package Real_Io  is new Text_IO.Float_Io    (Real);
package Int_Io   is new Text_IO.Integer_Io (Integer);
use E_F,Int_Io,Real_Io;

N : Constant Natural := 5;
a : array(-2**N..2**N,-2**N..2**N) of Real := (others => (others => 0.0));
G_normal: Float_Random.Generator;
file1 : File_Type;

function Random(a,b : Real) return Real is
begin
  return Real(Random(Gen => G_normal))*(b-a) + a;
end Random;

i1 : Constant Integer := a'first(1);
i2 : Constant Integer := a'last(1);
j1 : Constant Integer := a'first(2);
j2 : Constant Integer := a'last(2);

function G(i1,i2,j1,j2 : Integer) return Real is
begin
  return (Real(i2-i1)*Real(j2-j1)/10.0)**0.5/1.0;
end G;

procedure diamond(i1,i2,j1,j2 : Integer) is
   G1 : Real    := G(i1,i2,j1,j2);
   i0 : Integer := (i1 + i2)/2;
   j0 : Integer := (j1 + j2)/2;
   id,id1,id2,jd,jd1,jd2 : Integer;
begin

   id  := i0 - i1;
   id1 := (if i1 - id in a'range(1) then i1 - id else i1 + id);
   id2 := (if i2 + id in a'range(1) then i2 + id else i2 - id);
   jd  := j0 - j1;
   jd1 := (if j1 - id in a'range(1) then j1 - jd else j1 + jd);
   jd2 := (if j2 + id in a'range(1) then j2 + jd else j2 - jd);

   a(i1,j0) := (a(i1,j1) + a(i1,j2) + a(i0,j0) + a(id1,j0))/4.0 + G1*Random(-1.0,1.0);
   a(i2,j0) := (a(i2,j1) + a(i2,j2) + a(i0,j0) + a(id2,j0))/4.0 + G1*Random(-1.0,1.0);
   a(i0,j1) := (a(i1,j1) + a(i2,j1) + a(i0,j0) + a(i0,jd1))/4.0 + G1*Random(-1.0,1.0);
   a(i0,j2) := (a(i1,j2) + a(i2,j2) + a(i0,j0) + a(i0,jd2))/4.0 + G1*Random(-1.0,1.0);

end diamond;

procedure f(m,n,i1,i2,j1,j2 : Integer) with
    Pre => i2 - i1 < 2 or j2 - j1 < 2;
procedure f(m,n,i1,i2,j1,j2 : Integer) is
   G1 : Real    := G(i1,i2,j1,j2);
   i0 : Integer := (i1 + i2)/2;
   j0 : Integer := (j1 + j2)/2;
begin
   if m < n then
      f(m+1,n,i1,i0,j1,j0);
      f(m+1,n,i0,i2,j1,j0);
      f(m+1,n,i1,i0,j0,j2);
      f(m+1,n,i0,i2,j0,j2);
```

```
        elsif m = n then
           a(i0,j0) := (a(i1,j1) + a(i2,j1) + a(i1,j2) + a(i2,j2))/4.0 + G1*Random(-1.0,1.0);
           diamond(i1,i2,j1,j2);
        end if;
    end f;

begin

    Float_Random.Reset(Initiator => 1234,Gen => G_normal);
    a(i1,j1) := 1.0; a(i1,j2) := 1.0; a(i2,j1) := 1.0; a(i2,j2) := 1.0;

    for idepth in 1..N+1 loop
        f(1,idepth,i1,i2,j1,j2);
    end loop;

    Create(file1,Out_File,"rkut2d.dat");
    for i in a'range(1) loop
        for j in a'range (2) loop
            Put(file1,a(i,j),8,3,0);
        end loop;
        New_Line(file1);
    end loop;
    Close(file1);

end frac2d;
```

Figure B.3 illustrates the two first steps in the present modification of the diamond-square algorithm. The third dimension here represents time. The set of arrows pointing from a filled disk and onto a

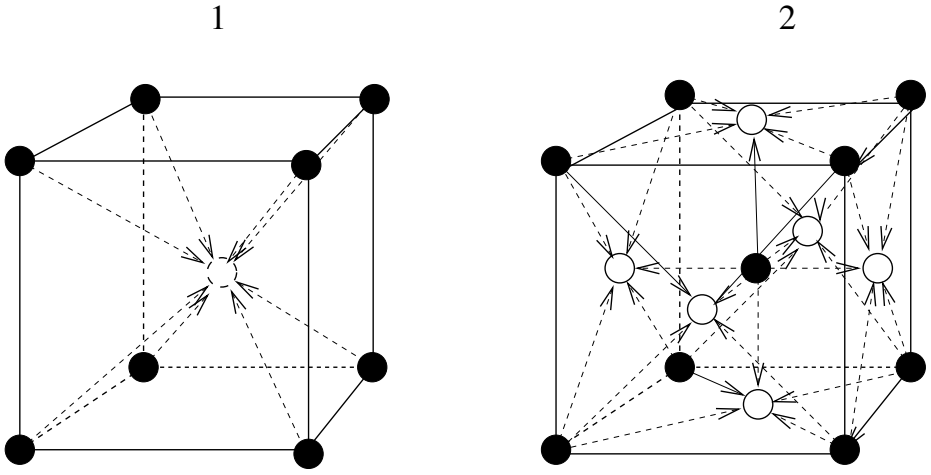

Figure B.3 Two first steps in the present modification of the diamond-square algorithm producing a correlated sequence of spatial landscapes.

non-filled (white) disk indicates that the algorithm set the value of the non-filled disk to the average value of these filled disks plus a random component. The algorithm produces for each index along the third dimension, a fractal landscape. The correlation decreases with increased separation in time. Figure B.4 shows an example of generation of such a landscape. The Ada source listing below implements the modified diamond-square algorithm.

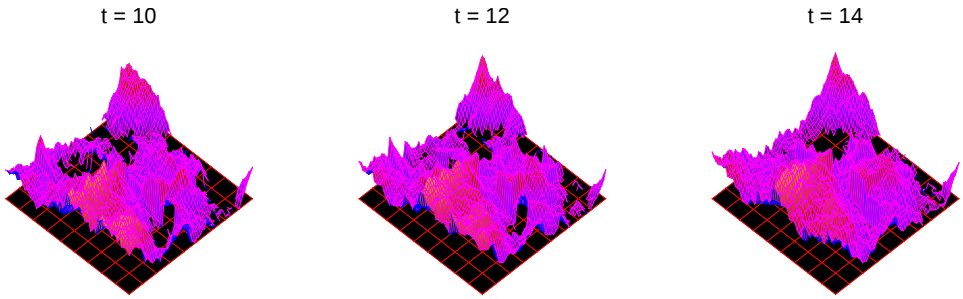

Figure B.4 Sub-sequential (correlated) fractal landscape produced by the modified diamond-square algorithm of Figure B.3 on an $65 \times 65 \times 65$ *array after application of a lower (constant) cut-off value.*

```
with Ada.Numerics.Generic_Elementary_Functions;
with Ada.Numerics.Float_Random, Text_IO;
use  Ada.Numerics, Ada.Numerics.Float_Random, Text_IO;
procedure frac3d is

    subtype Real is Float;

    package E_F      is new Ada.Numerics.Generic_Elementary_Functions (Real);
    package Real_Io  is new Text_IO.Float_Io    (Real);
    package Int_Io   is new Text_IO.Integer_Io (Integer);

    use E_F, Int_Io, Real_Io;

    M : Constant Natural := 5;
    N : Constant Natural := 5;

    a : array(-2**N..2**N,-2**N..2**N,-2**M..2**M) of Real := (others => (others => (others => 0.0)));

    G_normal: Float_Random.Generator;

    function Random(a,b : Real) return Real is
    begin
      return Real(Random(Gen => G_normal))*(b-a) + a;
    end Random;

    i1 : Constant Integer := a'first(1);
    i2 : Constant Integer := a' last(1);
    j1 : Constant Integer := a'first(2);
    j2 : Constant Integer := a' last(2);
    k1 : Constant Integer := a'first(3);
    k2 : Constant Integer := a' last(3);

    function G(i1,i2,j1,j2,k1,k2 : Integer) return Real is
    begin
      return (Real(i2-i1)*Real(j2-j1)*Real(k2-k1)/500.0)**(1.0/3.0);
    end G;

    procedure square (i1,i2,j1,j2,k1,k2 : Integer) is
       G1 : Real := G(i1,i2,j1,j2,k1,k2);
       i0 : Integer := (i2 + i1)/2;
```

```
   j0 : Integer := (j2 + j1)/2;
   k0 : Integer := (k2 + k1)/2;
   id,id1,id2 : Integer;
   jd,jd1,jd2 : Integer;
   kd,kd1,kd2 : Integer;
begin
   a(i0,j0,k0) := (a(i1,j1,k1) + a(i2,j1,k1) + a(i1,j2,k1) + a(i2,j2,k1)
                +  a(i1,j1,k2) + a(i2,j1,k2) + a(i1,j2,k2) + a(i2,j2,k2))/8.0;
   kd  := k0 - k1;
   kd1 := (if k1 - kd in a'range(3) then k1 - kd else k1 + kd);
   kd2 := (if k2 + kd in a'range(3) then k2 + kd else k2 - kd);

   a(i0,j0,k1) := (a(i1,j1,k1) + a(i2,j1,k1) + a(i1,j2,k1) + a(i2,j2,k1)
                   + a(i0,j0,k0) + a(i0,j0,kd1))/6.0 + G1*Random(-1.0,1.0);
   a(i0,j0,k2) := (a(i1,j1,k2) + a(i2,j1,k2) + a(i1,j2,k2) + a(i2,j2,k2)
                   + a(i0,j0,k0) + a(i0,j0,kd2))/6.0 + G1*Random(-1.0,1.0);

   id  := i0 - i1;
   id1 := (if i1 - id in a'range(3) then i1 - id else i1 + id);
   id2 := (if i2 + id in a'range(3) then i2 + id else i2 - id);

   a(i1,j0,k0) := (a(i1,j1,k1) + a(i1,j2,k1) + a(i1,j1,k2) + a(i1,j2,k2)
                   + a(i0,j0,k0) + a(id1,j0,k0))/6.0 + G1*Random(-1.0,1.0);
   a(i2,j0,k0) := (a(i2,j1,k1) + a(i2,j2,k1) + a(i2,j1,k2) + a(i2,j2,k2)
                   + a(i0,j0,k0) + a(id2,j0,k0))/6.0 + G1*Random(-1.0,1.0);

   jd  := j0 - j1;
   jd1 := (if j1 - jd in a'range(3) then j1 - jd else j1 + jd);
   jd2 := (if j2 + jd in a'range(3) then j2 + jd else j2 - jd);

   a(i0,j1,k0) := (a(i1,j1,k1) + a(i2,j1,k1) + a(i1,j1,k2) + a(i2,j1,k2)
                   + a(i0,j0,k0) + a(i0,jd1,k0))/6.0 + G1*Random(-1.0,1.0);
   a(i0,j2,k0) := (a(i1,j2,k1) + a(i2,j2,k1) + a(i1,j2,k2) + a(i2,j2,k2)
                   + a(i0,j0,k0) + a(i0,jd2,k0))/6.0 + G1*Random(-1.0,1.0);
end square;

procedure diamond(i1,i2,j1,j2,k1,k2 : Integer) is
   G1 : Real := G(i1,i2,j1,j2,k1,k2);
   i0 : Integer := (i2 - i1)/2;
   j0 : Integer := (j2 - j1)/2;
   k0 : Integer := (k2 - k1)/2;
begin

   a(i1,j0,k1) := (a(i1,j1,k1) + a(i1,j2,k1))/2.0 + G1*Random(-1.0,1.0);
   a(i0,j1,k1) := (a(i1,j1,k1) + a(i2,j1,k1))/2.0 + G1*Random(-1.0,1.0);
   a(i2,j0,k1) := (a(i2,j1,k1) + a(i2,j2,k1))/2.0 + G1*Random(-1.0,1.0);
   a(i0,j2,k1) := (a(i1,j2,k1) + a(i2,j2,k1))/2.0 + G1*Random(-1.0,1.0);

   a(i1,j0,k2) := (a(i1,j1,k2) + a(i1,j2,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i0,j1,k2) := (a(i1,j1,k2) + a(i2,j1,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i2,j0,k2) := (a(i2,j1,k2) + a(i2,j2,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i0,j2,k2) := (a(i1,j2,k2) + a(i2,j2,k2))/2.0 + G1*Random(-1.0,1.0);

   a(i1,j1,k0) := (a(i1,j1,k1) + a(i1,j2,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i2,j1,k0) := (a(i2,j1,k1) + a(i2,j1,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i1,j2,k0) := (a(i1,j2,k1) + a(i1,j2,k2))/2.0 + G1*Random(-1.0,1.0);
   a(i2,j2,k0) := (a(i2,j2,k1) + a(i2,j2,k2))/2.0 + G1*Random(-1.0,1.0);
```

```
    end diamond;

    procedure f(m,n,i1,i2,j1,j2,k1,k2 : Integer) with
           Pre => i2 - i1 < 2 or j2 - j1 < 2;
    procedure f(m,n,i1,i2,j1,j2,k1,k2 : Integer) is
       i0 : Constant Integer := (i1 + i2)/2;
       j0 : Constant Integer := (j1 + j2)/2;
       k0 : Constant Integer := (k1 + k2)/2;
    begin
       if m < n then
          f(m+1,n,i1,i0,j1,j0,k1,k0);
          f(m+1,n,i0,i2,j1,j0,k1,k0);
          f(m+1,n,i1,i0,j0,j2,k1,k0);
          f(m+1,n,i0,i2,j0,j2,k1,k0);

          f(m+1,n,i1,i0,j1,j0,k0,k2);
          f(m+1,n,i0,i2,j1,j0,k0,k2);
          f(m+1,n,i1,i0,j0,j2,k0,k2);
          f(m+1,n,i0,i2,j0,j2,k0,k2);
       elsif m = n then
          square (i1,i2,j1,j2,k1,k2);
          diamond(i1,i2,j1,j2,k1,k2);
       end if;
    end f;
    file1 : File_Type;
    cfile1 : String(1..2);
begin

    Float_Random.Reset(Initiator => 12340,Gen => G_normal);

    a(i1,j1,k1) := 1.0;
    a(i2,j1,k1) := 1.0;
    a(i1,j2,k1) := 1.0;
    a(i2,j2,k1) := 1.0;

    a(i1,j1,k2) := 2.0;
    a(i2,j1,k2) := 2.0;
    a(i1,j2,k2) := 2.0;
    a(i2,j2,k2) := 2.0;

    for idepth in 1..N+1 loop
        f(1,idepth,i1,i2,j1,j2,k1,k2);
    end loop;

    for k in a'range(3) loop
        cfile1 := (if k < 10 then "0" & Integer'Image(k)(2..2) else Integer'Image(k)(2..3));
        Create(file1,Out_File,"rkut3d_" & cfile1 & ".dat");
        for i in a'range(1) loop
            New_Line(file1);
            for j in a'range (2) loop
                Put(file1,a(i,j,k),10,2,0);
            end loop;
        end loop;
        Close(file1);
    end loop;

end frac3d;
```