Article

Cellular Simulation for Distributed Sensing over Complex Terrains

Tuyen Phong Truong[1,2]*, Bernard Pottier[1]* and Hiep Xuan Huynh[2]

[1] Université de Bretagne Occidentale, Lab-STICC, UMR CNRS 6285, France
[2] Can Tho University; Can Tho city, Vietnam
* Correspondence: tptuyen@ctu.edu.vn, pottier@univ-brest.fr

Abstract: Long range radio transmissions open new sensor application fields, in particular for environment monitoring. As an example, the *LoRa* radio protocol enables to connect remote sensors at distance as long as ten kilometers in line-of-sight. However, the large area covered also bring several difficulties, such as the placement of sensing devices in regard to geography topology, or the variability of communication latency. Sensing the environment also carries constraints related to the interest of sensing points in relation with a physical phenomenon. Criteria for designs are thus evolving a lot from the existing methods, especially in complex terrains. This article describes simulation techniques based on geography analysis to compute long range radio coverages and radio characteristics in these situations. As radio propagation is just a particular case of physical phenomena, it is shown how a unified approach also allows to characterize the behavior of potential physical risks. The case of heavy rainfall and flooding is investigated. Geography analysis is achieved using segmentation tools to produce cellular systems which are in turn translated into code for high-performance computations. The paper provides results from practical complex terrain experiments using LoRa, that confirm the accuracy of the simulation, scheduling characteristics for sample networks, and performance tables for simulations on middle range Graphics Processing Units (GPUs).

Keywords: Cellular automata; Complex terrain; LoRa; Parallel processing; Radio signal propagation

---

1. Introduction

Climate change and natural evolution impact seriously on several social and economic aspects, including living conditions, human health, and development. Autonomous observation is a key point for understanding evolution and the management of territories. This is achieved using small stations equipped with sensors, a transceiver, and power supply. Distributed sensors sample physical behavior. They enable to analyze environmental changes and to predict short term and medium term evolutions.

Wireless Sensor Networks (WSNs) have offered several frameworks to connect sensors. Radio standard such as 802.15.4 [1] have proposed short range mesh connected topologies suitable for indoor or local deployments. Recent innovations include *low power long range radio* systems, such as *LoRa* [2] or Sigfox [3]. Long range means covering of large surface, more risk during transmissions, long signal latencies, and more difficulties coming from the ground topology. *Star organization*, where a central node addresses remote sensors is the more natural technique to collect data on surfaces as in the order of ten thousand square meters.

Complex geographic areas such as shores or islands, hills, valleys oppose physical difficulties to the radio propagation. At the sensor level, the *coverage* also defines zones where information is accessible, whatever it is, wildlife including insects, physical elements such as water or gas.

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

Therefore, a *precise coverage computation* is critical for monitoring efficiency matching *radio connectivity objectives*, and sensing accuracy in regard to physical phenomena. This article describes principles of a general method based on geo-localized cellular systems. It explains how a radio coverage or a physical coverage can be computed from the same framework, allowing to obtain a better match between the observation engines and observed phenomena.

The introduction firstly gives a brief view on principles behind three components in the methodology: (i) the terrain analysis (Section 1.1), (ii) the signal propagation and coverage (Section 1.2), and (iii) the representation of physical phenomena behaviors (Section 1.3). Generic algorithms for sensing deployments can be built in relation with metrics for these components. This section will refer to a grid of points having geometric and geographic coordinates. Preliminary examples present a large region with mountains and creeks along the seashore, likely to be equipped with sensors. This level of explanation match needs at application engineering level.

Further sections will detail tool principles, with the presentation of the cellular methodology (Section 2), then algorithms that simulate the physical case of flooding (Section 3), and long range radio propagation (Section 4). Appendices will present validation results obtained for these algorithms.

1.1. A characterization of terrain complexity

Sensor investigation in a zone almost certainly starts with a glance at a geographic map or an aerial photography. If the zone includes mountains, hills, rivers, or shores, two questions will come, about the sensing point locations, the reachability of an infrastructure network, the adequacy with the sensing objectives. Design tools will help to take into account the geography characteristics and possible obstacles to signal propagation, while keeping track of these objectives. Sensing systems are application specific, point to point, and differ a lot from mobile radio communication based on regular coverages.

Terrain complexity metrics allow to measure ground irregularity, and offer the possibility to split and isolate zones of similar characteristics. Figure 1 displays complexity for a complex zone[1] based on Digital Elevation Models (DEMs) analysis.

Several metrics are known for this. *Terrain Ruggedness Index* (TRI) and *Topographic Position Index* (TPI) for DEM provide a quantitative measure of surface turbulence [4]. TRI is the square root of the summed squared deviation in elevation between a cell and its eight neighbor grid cells [5]. TPI is the difference in elevation value of a center cell and the mean of eight adjacent neighbor cells [6]. Both TRIs and TPIs grids illustrate the distribution of terrain heterogeneity and may be displayed in the form of a map as shown Figure 1. Figure 2 display analysis result for four cases, also showing how terrain characteristics allow separating concerns about sensor layouts. Partitioning the ground according to terrain complexity is a preliminary operation for network layout and selection of the pair to pair radio transmissions, to be obtained by computer programs.

---

1 Soummam Valley, leading to Bejaia city, the north of Algeria.

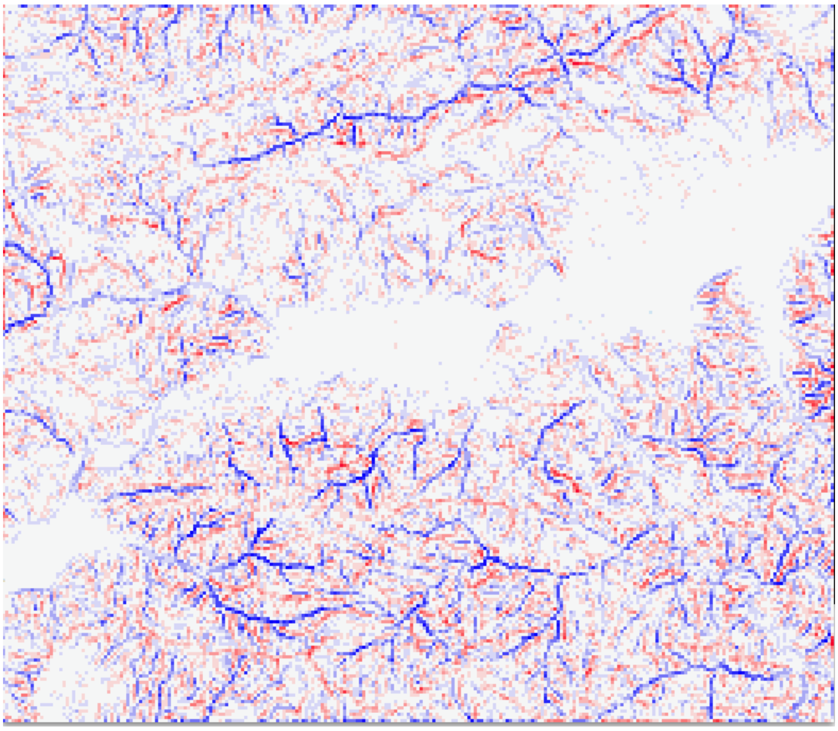

Figure 1. TPI terrain complexity for Soummam river in Algeria (see also Figure 4). Red lines represent higher points, difficult to overcome and blue lines are for lower points, difficult to reach. The white zone signals a flat ground without remarkable obstacles, the case of Soummam banks. This grid is 262 x 226 points, representing 30 x 25 kilometers.

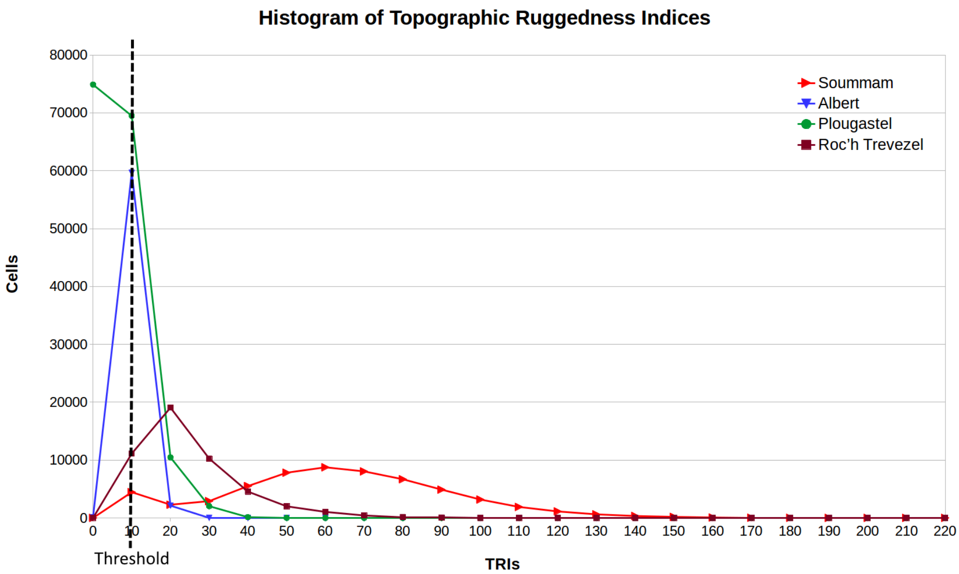

Figure 2. The histogram of topography ruggedness. Four zone analysis with high variability (red, Soummam), low variability (blue, Brest town, with several deep valleys and shore), a zone with an high percentage of sea surface (green, Brest bay), medium variability with low size hills (brown, Arree mountains.

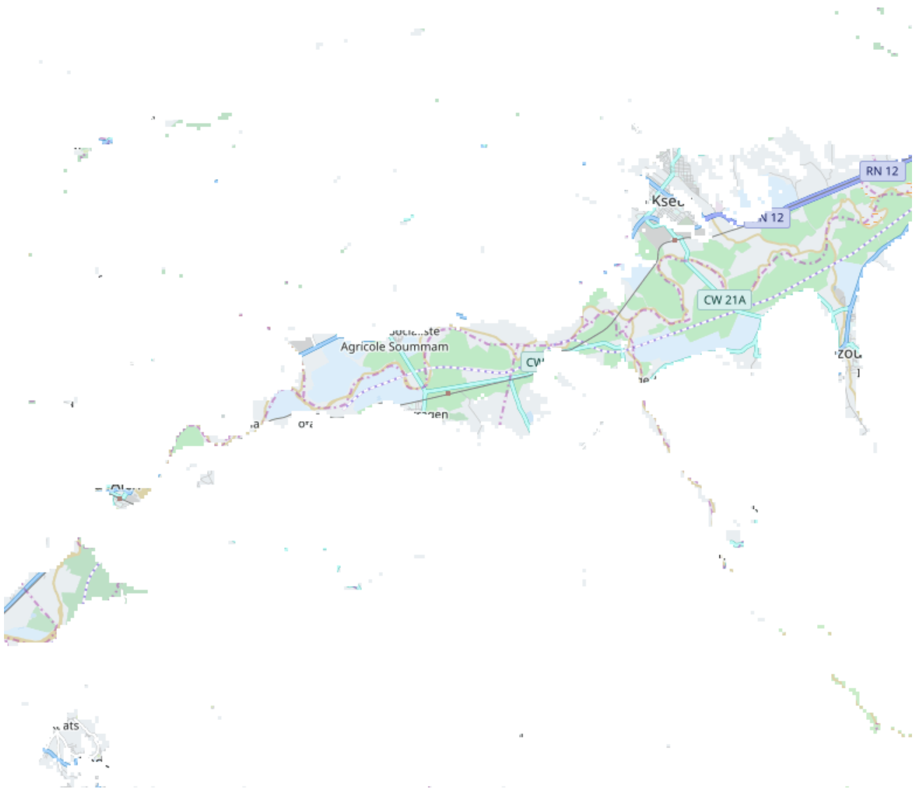

Figure 3. The subsystem of low complexity. Progressive thresholding on TRI produces a space partitioning according Terrain complexity. This is a flat zone around river Soummam, with complexity below threshold line on the right (Figure 2).

1.2. Designing for long distance radio coverage

Setup of long distance radio communication links is difficult if there is an opposition of the geographical topology.

In smooth terrain areas, radio propagation can be considered as *near line-of-sight (NLoS)*. In this case, the communication distance is expressed in relation with power reduction along the path. *Free space path loss (FSPL)* equation was proposed for this aim (see Section 5 and Formula 9 in Section 5).

Concerning complex terrains, the choice of a radio position can be application context dependent, topology-dependent, or it can come from a layout algorithm. Thus, it is critical to obtain a description of zones covered by a signal, plus other characteristics such as propagation delay from the source, and quality of the signal. Section 5 will explain how this can be done using cellular algorithms that mimic the spatial physical propagation over the terrain.

Figure 4 illustrates results of a coverage computation, from an expected source in the middle of the figure. The irregular yellow shape reflects the difficulty to choose source and destination given that any position between them can interrupt the signal propagation. Deploying a network of several nodes on long distances is not tractable by hand.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

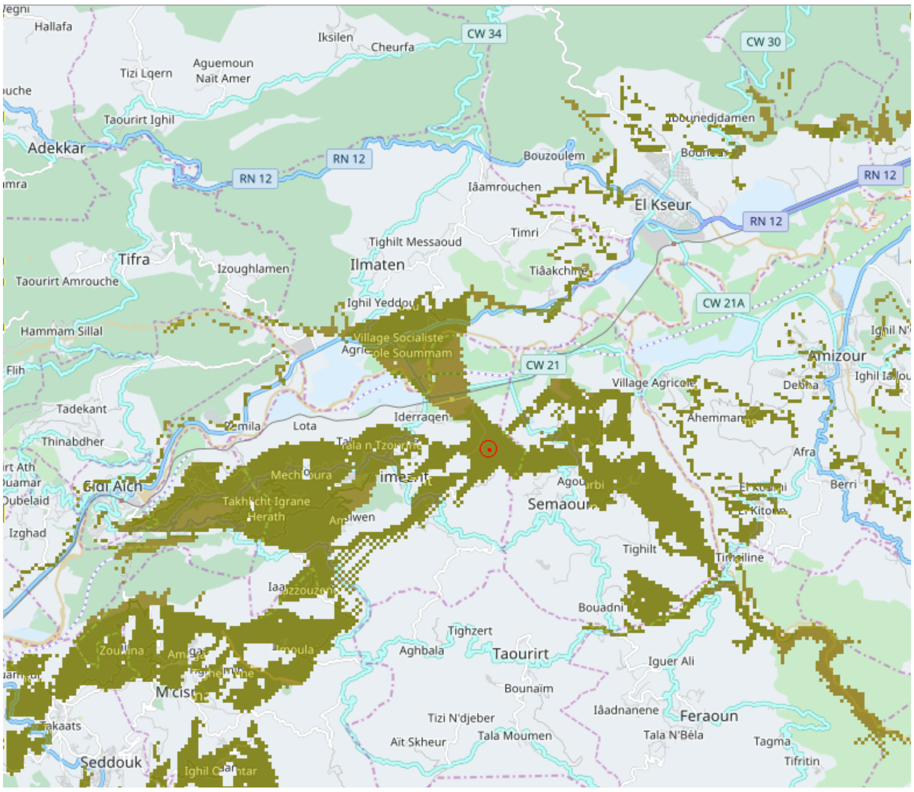

Figure 4. Display of a radio coverage for an emitter (red point), located above Soummam (36.622141028, 4.799995422)[2], elev:165.0. Coverage is shown in dark yellow, spreading over 30 km on the zone width. This location was chosen at random, giving a percentage of 32% grid points receiving the long range signal.

1.3. *Observing physical phenomena*

Another component for observation methodology is the necessary focus on the phenomenon to monitor. The nature of the sensing varies a lot, depending on application field. Sensing can be related to biology, climate, geology, human activities, or composition of parameters. Complex terrains will bring a complex behavior, as it is the case for radio signal propagation.

To illustrate the observation problem, let us consider heavy rain in the same region as in Section 1.2. The physical fact to be monitored is water streaming from the mountain to the river, then to the sea. Rain can be predicted from the meteorology services simulations, or just observed using radars, satellites, or ground devices. Causality for water flooding is the nature and shape of the terrain, sharing characteristics with terrain complexity and signal propagation.

Cellular simulation presented Section 2 will compute propagation of water given an expected, or existing rain history taken from a public service (this case), or extracted from radar networks [7] or satellite observations [8]. Figure 4 displays the water streaming from the mountain down to the river in the case of a heavy storm which profile is shown Figure 6.

In the resulting maps, high values represent flooding and possible risks. This analysis demonstrates that simulation allows establishing dependencies between risks and positions.

---

2 Geographic positions are proposed in (lat, long) form compatible with familiar map navigators.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

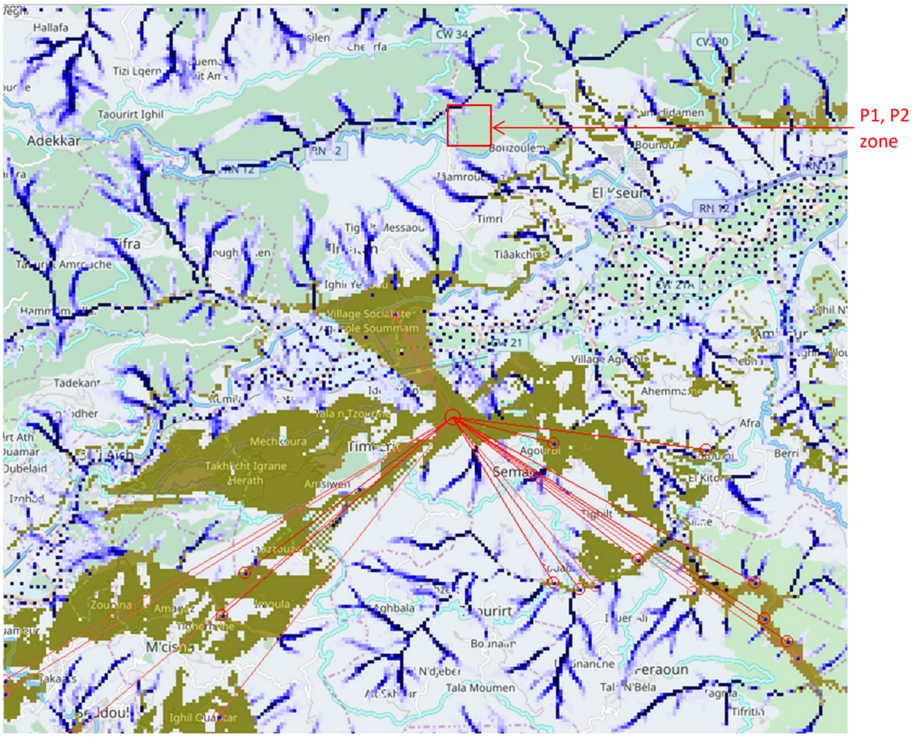

Figure 5. Heavy rainfalls context: a physical simulation has produced positions with the risk of flash flooding (dark blue color). The communication coverage of a base station with a star network is predicted for sensor nodes monitoring level of water in these positions. An algorithm select reachable sensor positions from a network sink, then sort, then extract 15 positions according to the flooding result simulation. More details about accuracy are given in Section 3.

Choice of real sensing positions can be produced by computer programs. As an example, the control that appeared Figure 5 can connect on radio coverage zones and obtain measures from critical positions in the rivers. The difference in communication latencies is another problem in the long range. This is also a place where computer tools can help by producing a schedule of sensor communication automatically.

Table 1. Geo-locations of points where values of water level were recorded.

Point	Color	Latitude	Longitude	Elevation (m)
P1	Red	36.589895361	4.925651550	554.7
P2	Green	36.587414366	4.925651550	536.8

The flexibility of cells embedding local parameters makes possible to represent many physical phenomena such as ocean or river wave effects, rain effects and flooding, air or water pollution, species behavior, sound, and alerts reachable zones, etc. Parallel algorithms have been designed to simulate rain flooding (Section 3) and propagation of the LoRa radio protocol (Section 5.1), the objective being to obtain high performances on this kind of problems.

Now, the article explains the geography space structuration based on cellular systems.

 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

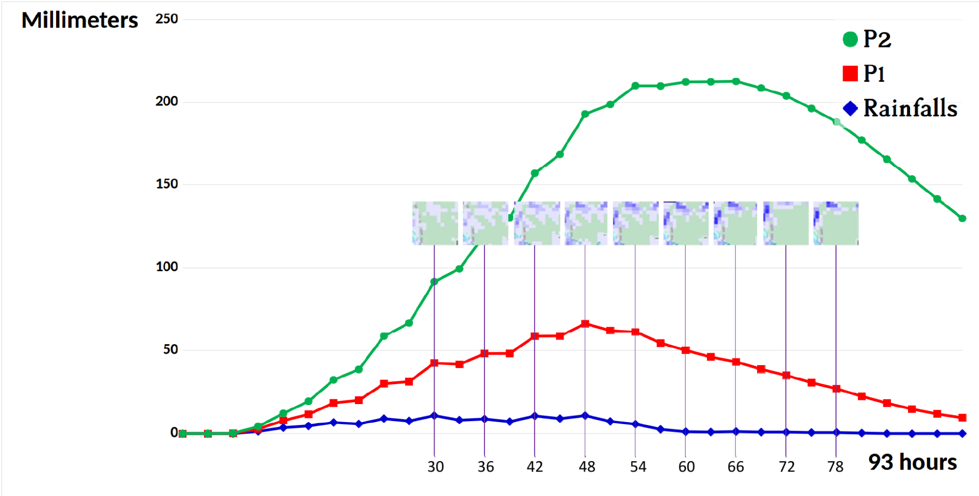

Figure 6. Chart of rainfalls and water levels at two positions in Soummam river zone during a tropical storm from 13 to 16 November 2017. The blue line shows rainfalls in 4 days recorded each 3 hours [9]. The two lines in red, green color present water level at P1, P2 respectively (also see Table 1). The distance between P1 and P2 is in the order of 300 meters. Due to the significant slope of the ground surface in this complex terrain, a large amount of water was accumulated at lower points, causing a flash flood, and possibly catastrophic landslide.

2. Mapping geographic space into cell systems

2.1. Geographic map and user interfaces

Several domains such as physics, medicine, or biology need sensing systems and geometric references. The case of geography includes necessary *projections* of a spherical surface into flat maps, with many options depending on applications, precision and considered location. Software tools exist for conversions [10]. Designing observation systems implies to consider distances and geometry related to the applications, and thus use of one or several reference systems to manage physical phenomena and their perception coherently. This section proposes an explanation of map building and displays to support the meaning of map contents, and data abstraction for geo-localized information additionally fetched from other sources, including simulations.

Tiled web maps are used to represent geographic data and to display information as flat graphics. This appears in proprietary software, or free access map systems such as OpenStreetMap. In *Tiled web maps*, the Earth is accessed by a projection rule called *Web Mercator* [11], and most often by a 3D designation mechanism *XYZ* used by web browsers and tile servers. The Z parameter provides a *zoom factor* from the whole Earth to more and more detailed view of the Earth fragments. At a given zoom factor Z, XY gives the 2D index of a tile inside the rectangle of tiles of this level Z. Based on this, tiles of graphical information is composed and delivered by servers to clients, usually in the form of 256×256 pixel images (Figure 8 presents a map composed from such tiles). These images are composed by a server rendering engine that extracts objects from a database and draws them according to some style (see Figure 7).

The browser *Quickmap* [12] supports standard map tiles, including OpenStreetMap, and a variety of others, either for maps or aerial images. This tool also allows to describes sensor systems, communication links, and mobile trajectories, for example, Low Earth Orbit (LEO) satellites [13].

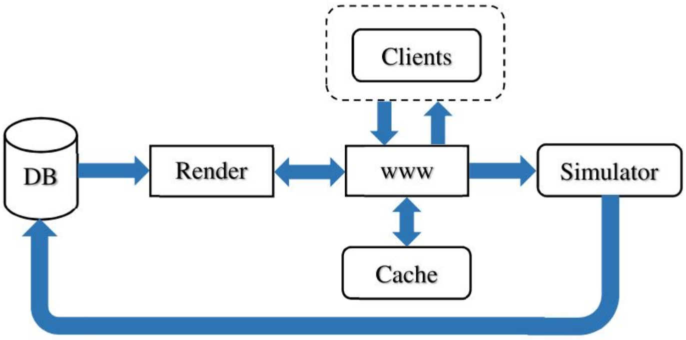

Figure 7. Tile map server architecture: tiles are requested from the web page server that either return a cached image or ask the rendering engine to compose it from a data base of geographical objects.

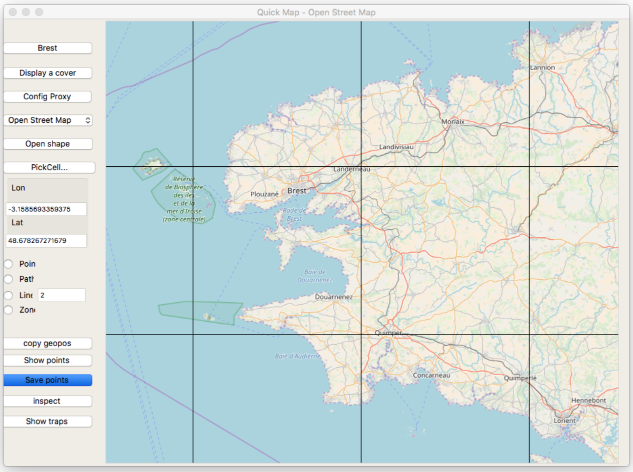

Figure 8. Quickmap tool [12] showing a tile coverage. For this case, zoom factor is 9, and last tile bottom right has x=251, y=177 indexes.

At a map browser level, users are specifying or observing *geometric points relative to a zoom level*. Real points are geographical positions, with coordinates specified according to the current projection system. In the case of Quickmap, they are double precision floating point numbers for latitude and longitude. Another concern is the necessity to use real distances in meters, over the Earth surface, for a task such as coverage computation. In the case of Web Mercator transformations between addressing mechanisms for *(lat, long)* absolute geographical specifications (x, y) distance in meters, (x, y) pixels, and XYZ tile access operations can be managed with formula from [14].

2.2. Definition of cells

Geographical cells can now be defined as *objects* grouping local data, local system behaviors, and graphic representation. This is described as an object-oriented class, as variables, and methods that are strictly local to the cells.

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

- *Cell locations* are bounded to geometric locations on maps, in relation with tile containers, and when possible to geographical locations.
- *Basic cell contents* are extracted from the map or image fragment as supported by the browser tool.
- *Cell content extensions* are obtained from external databases. Most of the cases are digital elevations, and also climate, or weather characteristics and histories.
- *Cell size* is chosen to match a particular physical phenomenon or sensing requirements.
- *Cell behaviors* are local procedures operating on a cell state. They need to be programmed to produce simulation data that in turn can be returned to databases, or displayed on tiles.

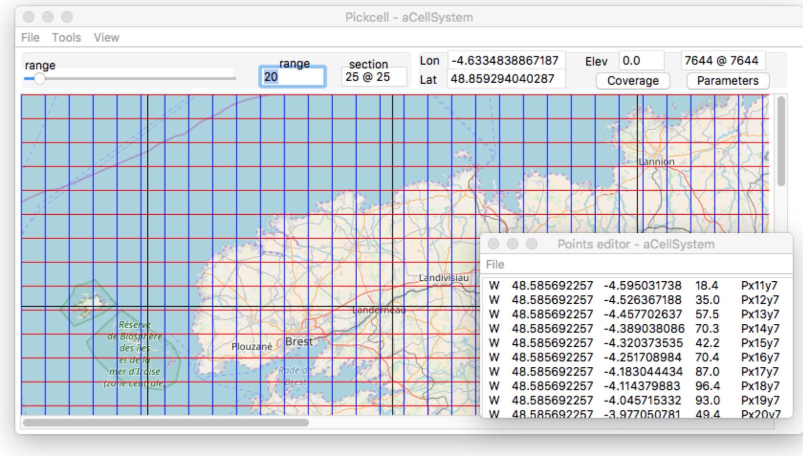

Figure 9. Presentation of a cell system organization over the tiles of Figure 8. Each cell has an identity produced from its location inside the window, and a geographical location. The text window bottom right also displays a plus parameter for the elevation. The cell size is 25×25 pixels, representing 7644 meters.

Examples in Section 1 illustrate how geographical abstraction, representation, and simulation can interact (Figures 1, 4, and 5).

Given a partitioning of a geographical zone into a cell system, we need to relate the observation system and physical behavior. The possibility to simulate physical processes will be of great help to correctly characterize a layout of sensors in regard to physical evolution.

Cellular automata (CA) allow modeling physical activities as processes that exchange information and evolve according to transition rules. The cell concept binds geographical fragments to such processes which assembly is generated automatically.

2.3. Cellular automata principles

Cellular automata were invented by John Von Neumann and colleagues with the aim to build a self-reproducing machine abstraction [15]. To support this goal, a two-dimensional space was configured with automata governed by a small set of states. This representation of space was used in several scientific domains and bound to physical behavior having similar properties [16]. CA can be described, and specified as a discrete space that associate cells. Cells evolve synchronously, steps by steps, following a discrete time. CA can be described in a variety of languages and executed on specific machines [17] following three simple patterns:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

- The *cellular space* is represented by an assembly of similar cells. A common notion of *neighborhoods* defines local communications following observed physical dependencies. The spatial organization can be either regular or irregular, possibly with disconnected subsystems as shown Figure 11, item 3.
- The evolution of each cell is defined in a *set of states* as observed in a real system (quantities, colors, boolean). Change of states are operated by procedures associated representing transition rules from step to step: $State_t \rightarrow State_{t+1}$.
- The *neighborhood* represents physical dependencies, for example, signal propagation, or downward flooding. These dependencies are connectivities from cell to neighbor cells. Common neighborhoods are Von Neumann and Moore with 4 and 8 cardinal directions respectively. Item *"Process architecture"* in Figure 11 illustrates a Moore neighborhood.
- The *transition rule* defines the behavior of each cell evolution under influence of its neighborhood and local *sensed* influences. The state of the whole cell system changes synchronously, time step by time step.

2.4. Cellular automata parallel execution models

A sequential execution model will read states in an array A_t for time t, and loop over, writing a similar array A_{t+1}. Once the step is completed, the two arrays are exchanged, and a new turn begins. CA is easy to parallelize preserving this behavior, grouping operations together and observing turns completion. Parallelism is mandatory because problems are large, if not huge, a lot of small cells can be critical for simulation precision, and some applications require to examine very large geographical regions.

- The synchronous distributed messaging model [18] can support parallel computation by associating cells to communicating processes. In this case, process progress by locked steps, based on messages being sent and received to or from neighbor nodes. The steps are split into two phases, one for communications with the neighborhood, the other to execute the transition rule. Figure 10 shows an internal node representation and the outside connection with three input and output links. This model does not need to specify the relative speed of processes, and can therefore be used for multi-cores or supercomputers.

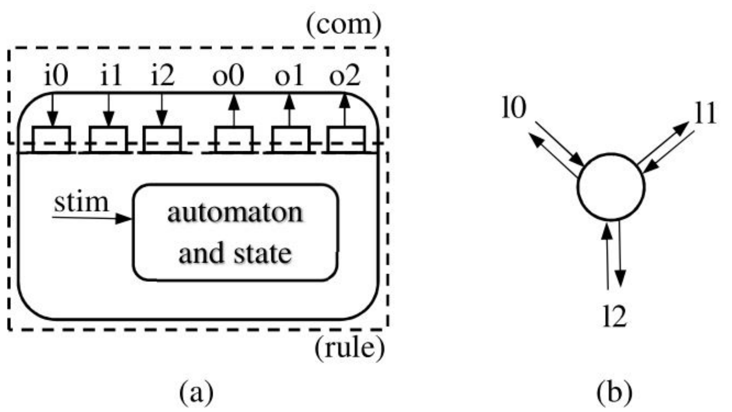

Figure 10. Cell node representation: (a) is the internal architecture with an automaton (*rule*) operating on incoming values and local stimuli (*stim*), based on a local set of variables, filling output communication buffers. Communications (*com*) are operated to and from the input and output buffers. (b) is the external point of view that only shows bidirectional links a cell node.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

- Another way to take advantage of parallelism is to use data parallel Single Instruction Multiple Data (SIMD) processors to execute a group of processes simultaneously. Current graphics accelerators propose solutions up to two thousand processors working concurrently, and exchanging data synchronously in shared memory. State spaces must be copied to the accelerator memory, then a loop of steps can be run completely on the accelerator. This is very efficient.

Production of code from cell systems to this schema has been done for two targets:

1. Asynchronized Occam communicating processes [19,20] targeting KRoC compiler and multi-cores [21],
2. CUDA code production [22] targeting *NVIDIA* tools and accelerators.

2.5. Cellular systems work flow organization

Most of the work described in this article is supported by a set of dedicated tools from University of Bretagne Occidentale (UBO). The design flow can be summarized as follows:

1. **Zone selection** is done by moving a graphical window anywhere, with any level of zoom. The tool extracts graphic tiles, and display the contents.
2. **Cell segmentation** is obtained by splitting the view into rectangles of a given size specified as a width $\times$ height value. Cells will carry an image and a geographical location from the underlying image.
3. **Binding cell together** and producing a cell system implies the choice of a connectivity (Moore, Von Neumann), and possibly filtering cells by colors or elevation. This step also injects external values from a variety of sources, including elevation.
4. **Adding behavior** is programing the cellular system at the local level, given a cell system architecture that step 3 (Figure 11) produces automatically.

Steps 1 to 3 are interactive and can be achieved *in minutes*. Step 4 is a concurrent programming activity specific to a concurrent platform it requires elaboration of CA transition rule and coding for the target platform.

Physical phenomena simulation can reveal places of interest for sensing. In the case of flooding, simulation track accumulation and circulation of water.

3. Heavy rain simulation on a complex terrain

To illustrate cell system interactions and behavior, we use the example of a zone *receiving* and *flooding* rain, as discussed in Section 1.3.

3.1. Managing space

Space is defined by geographical coordinate bounds, and a possible selection operated on cells. Selected cells have common properties, such as an elevation above or under a threshold level (Figure 11, step C), some color characteristics, or some signature produced from observed parameters. Thus the cell system can be rectangular or have arbitrary shapes and isolated subsystems.

Practically, cell systems first appear as arrays of objects that carry geometric coordinate, geographic coordinate, and a pixel array extracted from the original image.

Besides the zone structure, designers need to specify neighborhood, representing local physical influences and producing the necessary connectivity between processes. We can just admit that the space animation will be obtained by messages exchanged between neighbors.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1
Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

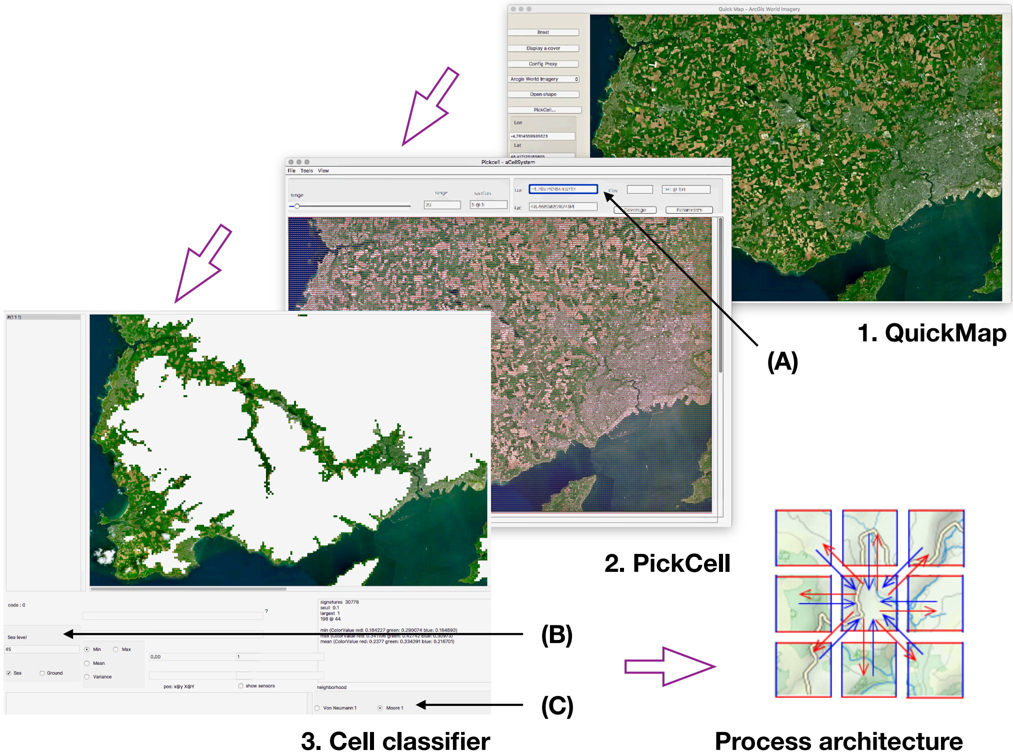

Figure 11. Cell synthesis flow: (1) a zone was located from a Quickmap navigation, and (2) was segmented into cells, then a subsystem was extracted filtering cells with elevations less than 45 m. (3) A cell system was generated following Moore topology. Annotations show controls for geographical positions with a cell size of 5 × 5 pixels representing a 191 × 191 m^2 surface (A), classification was operated for elevation (B). The neighborhood was Moore, radius 1 (C).

Execution software and hardware will allow adaptation to the intermediate communicating process model.

The case of rain reveals several interactions:

1. Millimeters of water falling on the ground. For this simulation, we admit that the quantity can vary with the time, but will remain uniform.
2. Water disappearing locally for reasons such as absorption or evaporation.
3. Water passed *locally* from cell to cell according to elevation differences.

Other cases include different specific problems for signal propagation (Section 5), sound propagation [23], or insect swarms behavior [24]. For flood modeling, Von Neumann neighborhood is convenient. Figure 12 shows such a neighborhood, with the North cell lost, and some elevation differences appearing above and below a center cell.

3.2. Transition rule

Each cell will receive rainwater, dispatch part of this water to neighbor cells with lower elevation, and absorb another part. A complete study will take into account the ground specific

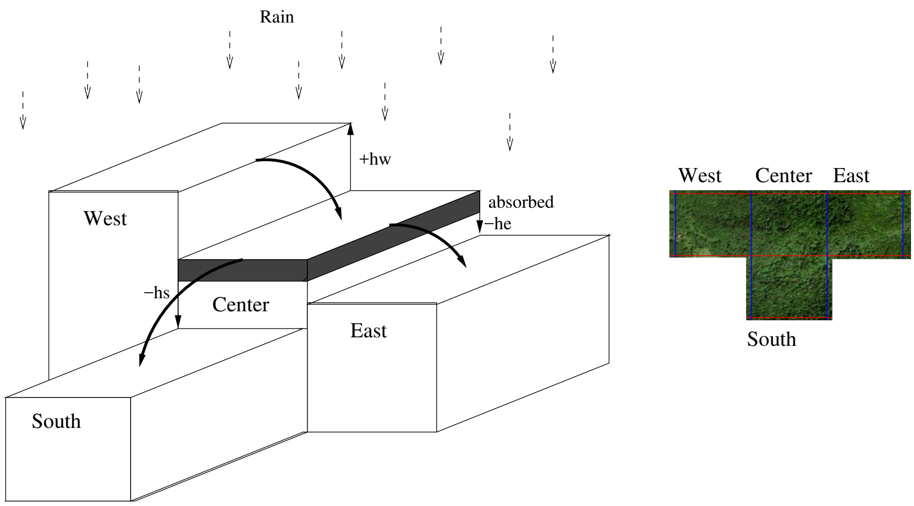

Figure 12. Physical exchange during rain. An incomplete neighborhood from a system shows a center cell with 2 neighbors west and east, and 1 in the south. The physical behavior is water flowing downward, represented here by synchronous messages sending water quantity west to center, and center to the east. Refer to [25] for more realistic behaviors.

characteristics, and current weather[3]. While real dependencies are water leaking from cell to cell, the abstract behavior is represented by messages sent and received to and from neighbors. Vertical water behavior is directly encoded in the transition rule that can be specified as follows:

t: time t represented by a step number
Q_t: water quantity in a center cell at the beginning of time step t
α_t: rainfall at time t
β: percentage of water remaining on each cell after each step
$cell_i.elevation$: elevation value of $cell_i$
$cell_c.elevation$: elevation value of center cell
δ_i: the difference of elevation between neighbor $cell_i$ and the center cell
Δ: sum of all elevation differences
$outF_i$: amount of water out coming from center cell to neighbor $cell_i$
inF_i: amount of water in coming to center cell from neighbor $cell_i$

The water quantity of center cell at time $t + 1$ after local absorption is

$$remaining = Q_t \times \beta \tag{1}$$

But the center cell receives rainfall within its area as following

$$rain = \alpha_t \tag{2}$$

The quantity of water coming from neighbors is taken into account

$$received = \sum_{i=1}^{n} inF_i \tag{3}$$

[3] ECOCLIMAP [26] supporting French meteorology AROME model provide more than 20 parameters for cells of $1km \times 1km$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

where n is a number of neighbors of the center cell.

A center cell also has to distribute water to surrounding neighbors in the proportion of elevation differences. This is calculated as follows

$$\delta_i = cell_c.elevation - cell_i.elevation \tag{4}$$

It is obvious that the water of a center cell only flows to neighbor cells with lower elevations so that only positive values of $delta_i$ are used to compute the total proportion of all differences

$$elevationAbove = \Delta = \sum \delta_i, \forall \delta_i > 0 \tag{5}$$

Equation 6 figures out the amount of water that a cell distributes to its lower neighbors. For higher elevation neighbors, with $\delta_i < 0$, there is no water flowing out, $outF_i = 0$.

$$sent_i = outF_i = Q_t \times (\delta_i / \Delta), \forall \delta_i > 0 \tag{6}$$

Eventually, center cell updates its own quantity of water by summing and subtracting. The whole system executes a synchronous transition from t to $t+1$.

$$Q_{t+1} := remaining + rain + received - \sum_{i=1}^{n} sent_i \tag{7}$$

Notice that this transition rule needs to know the neighbor relative elevations $cell_i.elevation$. In reality, physical rules will discover these relations naturally. In the case of simulations, a preliminary procedure called *neighborhood discovery* is executed to establish initial knowledge such as the number and the elevations of cells around.

4. Parallel algorithms for cellular long range coverage computations

4.1. General idea

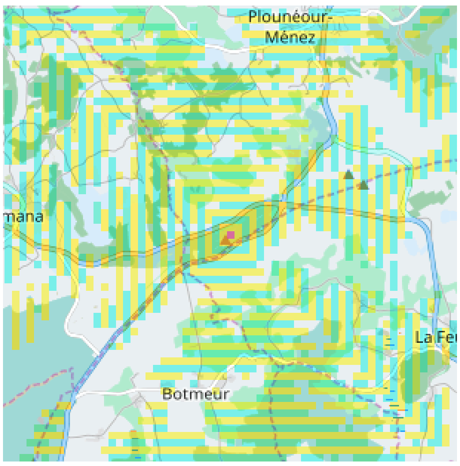

Figure 13. Radio signals propagate in concentric squares step by step. Reachable cells are represented in colored stripes.

Once a geographic space has been selected, a concern is to compute reachable cells in the line of sight from an emitting position. According to section 2.4, we admit that each cell is represented by a process, and simulation is achieved by cycling on a synchronous parallel

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

program: communication with neighbors, updating the local state, and preparing next cycle communications. The *line of sight* (LoS) is a ray broadcast in any direction from a root emitter. Propagation will be stopped or modified by ground obstacles such as hills, valley, etc. The simulation parallel algorithm mimics the physical behavior, by propagating the signal inside a spanning tree rooted at the emitter cell, and covering progressively all the space in concentric "*circles*".

Each new step in the algorithm adds a new circle, and the computation finish in $2 \times log(n)$ steps where n is the number of cells. During ray propagation, the ground profile is collected into a route. A route profile is shown in Figure 14. Routes are completed progressively based on positions and elevations. Each cell can thus decide if the emitter is visible or not by comparing its slope to root to previous ones in the received profile, as shown in Algorithm 1.

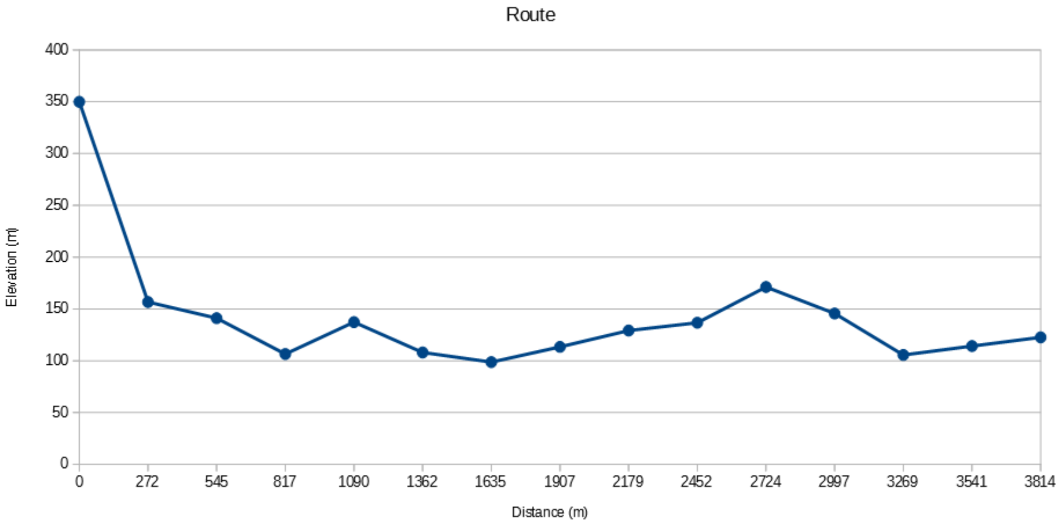

Figure 14. A profile obtained along a route. Let assume that an emitter located at furthest left of the chart, with distance 0 m, elevation 350 m. According to LoS condition three points at distances 2997 m, 3269 m and 3541 m seem cannot receive the signal from the emitter.

Algorithm 1 Setup visible nodes from emitter based on LoS condition

---

procedure FSPL
    Input: $cell_i$ and its neighbors $cell_j$
    Output: $cell_j.visible$

---

    for each $cell_j$ **do**
        if ($cell_j.slope <= cell_i.slope$) **AND** (receivedPower $>$ sensibilityPower) **then**
            $cell_j.visible = TRUE$
        else
            $cell_j.visible = FALSE$
        end if
    end for
end procedure

4.2. Vertical model

Signal propagation is progressing in steps according to the time. It builds progressively a route from the emitter to any other cell, as an array holding traversed locations and signal parameters (Figure 14). As it is the case for most of the cellular simulation algorithms, nodes start with a discovery stage, where geometric coordinates are exchanged to bind communication channels to cardinal directions. This is mandatory to deal with the irregularity of cell system shapes as shown in Figure 11, step (3), and also to maintain horizontal signal direction, as discussed in Section 4.3.

To start the building a propagation, an emitter has to initialize a tracing route with its own identity and geographic location. Other cells start their processes with an empty tracing route. Note that the tracing route, in this system, is the payload of the message for communications within cell system. The root cell then sends its message to all neighbors via eight directions from Moore neighborhood. Each cell accepts only one message then assigns the corresponding owner of the accepted message as its parent. This cell must send an acknowledgment message back to parent cell to confirm the dependency relationship between them. The next step in the transition rule is to insert its own location into the tracing route, then to pack routes and spread out to neighbors. If *side* is the maximum between *width* and *height* then at most, *side* rounds are necessary to cover a whole zone.

4.3. Horizontal model and Directed Breadth-First Search

The original Breadth-First search (BFS) algorithm build a tree covering a whole network starting from a root position. For radio propagation, this tree would allow reaching any cell in a minimum number of steps, propagating a route to this cell without managing signal horizontal directions. In the case of radio signals, there is the necessity to guide the route horizontally to maintain the best approximation of a straight line. A sequential algorithm that reduces horizontal errors to a minimum has been proposed by Bresenham to draw graphic lines on bitmaps [27]. In our case, it was preferred to find a parallel distributed approach based on Directed BFS (DBFS), that could also match particular geographic or electromagnetic dynamic considerations. Figure 15 shows DBFS routes resulting from this algorithm according to an emitting node and some target receivers.

At the difference of a BFS, the cell transition function makes a decision on receipt of route messages based on the source position and its own source position. It adds its data to the route, and it forwards the route to neighbors of interest. The neighborhood has been discovered at initialization stage, associating link indexes and cardinal directions.

5. Radio signal propagation on complex terrains

Radio propagation models are used to describe the qualifications and reliability of links using radio frequency. These models based on the physics of the diffusion of electromagnetic waves with respect to both constructive and destructive interference of environment. They can be practically applied to parameters inside routes transmitted from cell to cell, according to Section 4.

Free space path loss (FSPL) is the simplest model based on the Friss transmission equation. By assuming unobstructed along the transmission path, FSPL represents the line-of-sight decay of an electromagnetic wave as a function of only distance. This model exposes significant limits for complex terrain areas. Empirical path loss models were proposed to cope with the effects of turbulent terrains. An advantage is that these models attempt to estimate power loss as a function of distance and radio frequency taking into account of terrain heterogeneity as well as the effects of real transmission environment [28,29].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1
Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

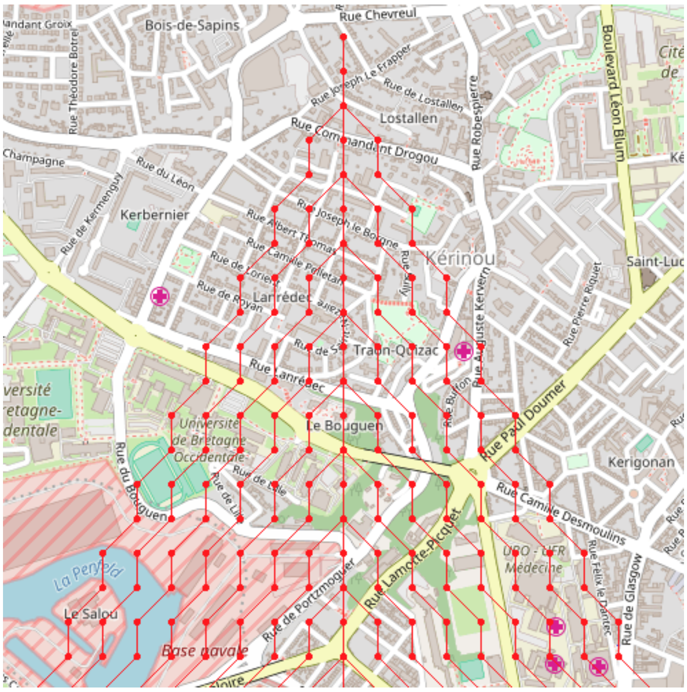

Figure 15. Directed BSF is a distributed parallel algorithm being able to manipulate data on grid cells. Its aim is to mimic point-to-point radio links adopting LoS condition. Segmented lines laid on a map to represent how cells forward incoming signal gradually from a root cell (x=10, y=1) to definite directions.

5.1. Free space path loss model

In terms of radio communication, a *Free Space Condition* is a region where there is no obstacle along the propagation path of radio waves. A radio signal is emitted by a transmitter and then it propagates in any direction at the speed of light. In this case, the signal energy can be received by an antenna in inverse proportion of the distance away from the signal source. In addition, the received power depends on the transmitted power, and gains of both transmit and receive antennas. Fading margin and other loss such as system loss, cables, connectors are also considered. The power of the signal is lost on the path is called the *Free Space Loss* represented by

$$FSPL = 20log_{10}(\frac{4\Pi d}{\lambda}) \tag{8}$$

Algorithm 2 is derived from Equation 8 in order to describe how to calculate the attenuation of signal power due to free space path loss. In consequence, the free space power received is given by equation 9, Friis free space equation [30].

$$P_r(d) = \frac{P_t G_t G_r (\lambda)^2}{(4\pi)^2 d^2 L} \tag{9}$$

where

P_t: transmitted power
$P_r(d)$: received power
G_t: transmitter antenna gain
G_r: receiver antenna gain

d: distance between transmitter and receiver in meters
L: system loss factor
λ: wavelength in meters

Algorithm 2 Received signal power in Free Space Path Loss Model

```
procedure SIGNALPOWER
    Input: txPower, waveLength, distance
    Output: receivedPower
    for each neighbor of cell_i do
        if neighbor.visible then
            receivedPower ← txPower × (SQR(waveLen /(4.0 ×valuePi ))× SQR(1/distance))
        end if
    end for
end procedure
```

5.2. Single knife-edge diffraction model

Diffraction appears around objects such as buildings, vegetation, etc.... The losses caused by these obstacles can be described by a *Knife Edge Diffraction* model. This is considered as a function of the path difference around the obstacles and could be explained by Fresnel zones. Diffraction loss of signal after a knife edge is calculated as a function of Fresnel parameter v.

$$v = h(\sqrt{\frac{2}{\lambda}(\frac{1}{t_t} + \frac{1}{d_r})}) \tag{10}$$

where h is the height of obstacle. d_t, d_r are the distances from the obstacle to emitter and receiver, respectively. Algorithm 3 demonstrates the computation of received power within a shadowing area because of a single knife edge obstacle.

$$L(v) = 6.9 + 20log(\sqrt{(v-0.1)^2+1} + v - 0.1) \tag{11}$$

In Algorithm 3, *receivedPower* is defines as the signal strength in dBm obtained at a receiver. In addition, *sensibilityPower* is the minimum value of signal strength in which condition a receiver can detect and demodulate a message successfully.

5.3. Okumura-Hata model

This is a dedicated radio propagation model to predict the path loss regarding geographic environments such as urban, suburban, and open areas [30,31]. The formula for Okumura-Hata model is given by

$$L = 69.55 + 26.16log_{10}(f) - 13.82log_{10}(h_b) - a(h_m) + (44.9 - 6.55log_{10}(h_b))log_{10}(d) - K \tag{12}$$

where

f_{MHz}: center carrier frequency of transmission band in MHz.
h_b: antenna height of base station in meter.
h_m: antenna height of mobile node in meter.
d_{km}: distance in kilometers.
$a(h_m)$ for each type of area and K see Table

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

Algorithm 3 Received signal power in Single Knife-Edge Diffraction Model

```
procedure DIFFRACTIONLOSS
    Input: fsplPower, h, d1, d2
    Output: diffPower
    v = h * SQRT((valuePi/2)×((1/d1) + (1/d2)))
    if v < 0 then
        diffLoss ← 0
    else
        if v < 2.4 then
            diffLoss ← 6 + (9 × v) + (1.27 × SQR(v))
        else
            diffLoss ← 13 + 20× log10(v)
        end if
    end if
    diffPower = fsplPower - diffLoss
end procedure
```

The model is valid for radios using carrier frequencies from 150 to 1500 MHz with an effective height of antennas from 30 m to 1000 m and distances ranging from 1 km to 100 km.

Table 2. List of $a(h_m)$ and K values for each type of area [30].

Type of area	$a(h_m)$	K
Open		$4.78[log_{10}(f)]^2 - 18.33 log_{10}(f) + 40.94$
Suburban	$[1.1 log_{10}(f) - 0.7]h_m - [1.56 log_{10}(f) - 0.8]$	$2[log_{10}(f/28)]^2 + 5.4$
Small city		0
Large city	$3.2[log_{10}(11.75h_m)]^2 + 4.97$	0

5.4. Correctness of the Different Radio Communication Models

The terrain was a mountain area investigated by applying the three models during simulations. The computation results are shown in Table 3. Figure A7 displays the coverage obtained by the model of *single knife-edge diffraction*, and effective coverage measured by a mobile.

Table 3. Comparison in number of points found in coverage from a total of 96431 points.

Model	Points
Free space path loss	19537
Single knife-edge diffraction	22305
Okumura-Hata	17621

6. Conclusion

To optimize communication ranges two major concerns are radio waves blocked by obstacles and propagation loss in proportion with a point to point distance. Terrains to be covered can be classified and partitioned as smooth, or rough according to metrics (Section 1.1). In smooth terrain

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

areas, radio transmission can be considered as near line-of-sight (NLoS). As a result, the issue of topologies can be almost neglected and the communication distance is expressed in relationship with power reduction along the path. Free space path loss (FSPL) equation was proposed for this aim (Section 5). For example, using LoRa technology for a point-to-point connection in NLoS condition, the distance for transceiving data is expected to reach hundreds of kilometers [32].

In complex terrain areas, heterogeneity of topologies causes serious impact on the quality of radio links, especially for low power and long range communication networks. Different propagation models allow representing the median of the expected path loss such as Longley-Rice model, ITU model, Okumura-Hata model, as described in Section 5.

Appendices give details on experimental measurements, with the communication distance checked to be around 5 km in complex urban area and up to 20 km in rural and shore ones. Experiment results shown Table A1 confirm the interest of a computer-aided approach. It may be concluded that communication in LoRa network is strongly dependent on the considered environment. Hence, the actual coverage prediction must take topographic complexity into account as a critical factor. An assessment of physical simulation was also conducted for flash flooding problem (Section 3). This algorithm was applied to the practical case of an intense rain found in June 3, 2018, and compared with flooding observations. The rain occurred in a rough terrain urban area and the simulation was able to retrieve major flooding places and level of water. There is a major interest in this kind of physical simulations that reveals places of interest for sensing and establish causality between events.

Table 4. A characterization of space and signal relations.

Datagrid	Model	Resolution (m)
SRTM30	2D/3D	30 [4]
SRTM90	2D/3D	90 [5]
Weather broadcast	2D/3D	3000[6]
LiDAR	3D	5[7]
Coastal ocean	2D/3D	1200 [8]
Earth magnetic	2D/3D	3700 [9]
Ocean acoustics	2D/3D	2500 [10]
Road traffic noise	2D/3D	0.5 [11]
Pickcell	2D/2.5D	30

The software developed in this project is easy to use up to cellular system generation. Implementing cellular transition rules, tuning and verifying these rules necessitate in-depth investigations. Several domains were investigated, from sound propagation to insect behaviors. Thus the methodology appears very general and flexible (see Table 4). Parallel programs implement described algorithms very efficiently (Tables 5 and 6). Efficiency is mandatory if space exploration strategies are to be developed.

4 https://lta.cr.usgs.gov/SRTM1Arc"https://lta.cr.usgs.gov/SRTM1Arc
5 https://blogs.esri.com/esri/arcgis/2015/06/26/terrain-3d-now-with-global-srtm-30-meter-content/""
6 https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/numerical-weather-prediction
7 https://content.meteoblue.com/en/research-development/data-sources/nmm-modelling/model-domain"
8 https://pdfs.semanticscholar.org/55be/a487827c28aaaf713017c499e4f33aed62fd.pdf
9 http://www.sciencedirect.com/science/article/pii/S0377042798002465
10 https://www.ngdc.noaa.gov/geomag/emag2.html
11 Peter Wille. Chapter 5: The Sea Floor - Natural Formations. In *Sound Images of the Ocean: in Research and Monitoring*. Springer-Verlag Berlin Heidelberg, 2005. ISBN 978-3-540-27910-5

Table 5. Execution times for three processes. There are 58725 cells in regular squared grid with cell size 3 x 3 pixels corresponding to actual area 115 x 115 meters.

Processes	Number of rounds	Time (s)
Compute ruggedness index	1	0.026
Flood simulation	32	0.103
Coverage prediction	261	0.211

Table 6. Execution times for flood simulations which are performed on Linux Ubuntu PCs with Intel® Core™ i3-4005U CPU @ 1.70GHz x 4, 8 GiB DDRAM, card NVIDIA GeForce 820M (96 CUDA Cores), Intel® Core™ i7 CPU 920 @ 2.67GHz x 8, 4 GiB DDRAM, card NVIDIA GeForce GTX 680 (1536 CUDA Cores) and Intel® Core™ i7-7700K CPU @ 4.20GHz x 8, 16 GiB DDRAM, card NVIDIA GeForce GTX 1070 (1920 CUDA Cores).

Resolution (pixels)	Number of cells	Execution time 820M	Execution time GTX 680	Execution time GTX 1070
3x3	58725	28.168 (ms)	6.5269 (ms)	1.3373 (ms)
5x5	21060	10.411 (ms)	2.1728 (ms)	518.64 (μs)
10x10	5226	2.5234 (ms)	454.59 (μs)	83.696 (μs)
15x15	2340	1.1155 (ms)	181.48 (μs)	65.916 (μs)
20x20	1287	602.47 (μs)	171.19 (μs)	62.085 (μs)

Acknowledgments:

City of Brest has supported a doctorate study that led to the initial design of tools shown figure .

Special thanks to France Ministry of Foreign Affairs (MAEDI) for the support of SAMES project in its STIC-Asia program, and to Vietnam MOET ministry for the support of doctorate studies in this project.

Significant student contributions helped the developments of high level tools and code generation targeting GPU accelerators. Combining several simulations using High Level Architectures was also investigated in this way.

High level software layers were developed on the Visualworks Smalltalk platform, while parallel processing was implemented using Kent Research Occam Compiler and NVIDIA CUDA tools.

All participants to the SAMES project have contributions to the project management, advising, developments and experiments [33], with special thank to Vincent Rodin. SAMES is a companion project of the Persepteur ANR project.

Appendix .1 A study in flash flood event

A case study for flash flooding prediction due to heavy rain during a storm on Monday, June 3, 2018, in Morlaix town, France [12]. The objective is for validating the accuracy of our proposed tool for flash flooding, especially for complex terrain areas.

Apart from data grid produced by QuickMap/NetGen/PickCell, to execute flooding simulation it is necessary the rainfall values measured in each hour during a storm, hurricane,

[12] http://www.letelegramme.fr/Finistere/morlaix/inundations-a-morlaix-evolution-de-la-situation-en-direct-03-06-2018-11980811.php

 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

etc. as input data (see the blue line in Figure A3. The data can be retrieved from the database of weather broadcast services such as meteofrance.com, meteoblue.com, etc.

Figure A1. Heavy flooding in center of Morlaix, France due to a storm in June 3, 2018 (Photo: Le Télégramme).

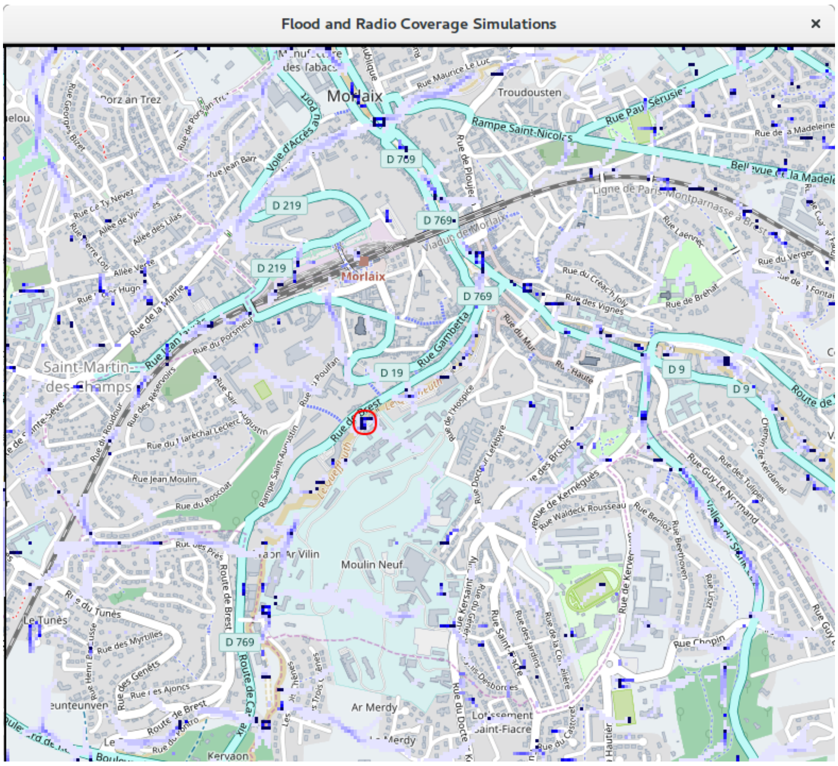

Figure A2. Simulation result for Morlaix, obtained by executing massively parallel processes on GPUs with thousands of CUDA cores. The experimental zone comprises of 58275 cells corresponding to actual area 3 x 3 km. A place of concern is indicated by a red circle This is *rue de Brest* near *Queffleuth* river (longitude: -3.8329839706421, latitude: 48.573767695437, elevation: 10.1 m) where water level went up to 60 cm according to *Le Télégramme* newspaper. The main flooding directions can be seen in light blue over the maps.

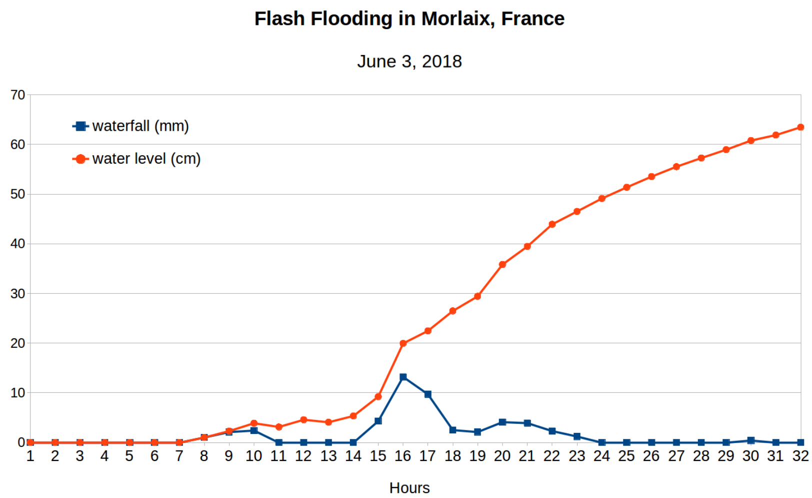

Figure A3. Chart shows rainfall levels during 32 hours from 00:00 June 3, 2018, to 08:00 June 4, 2018. It is noticed that the serious flooding often comes several hours after a storm or heavy rain. After the rain stopped, it requires a couple of hours or days to drain all water.

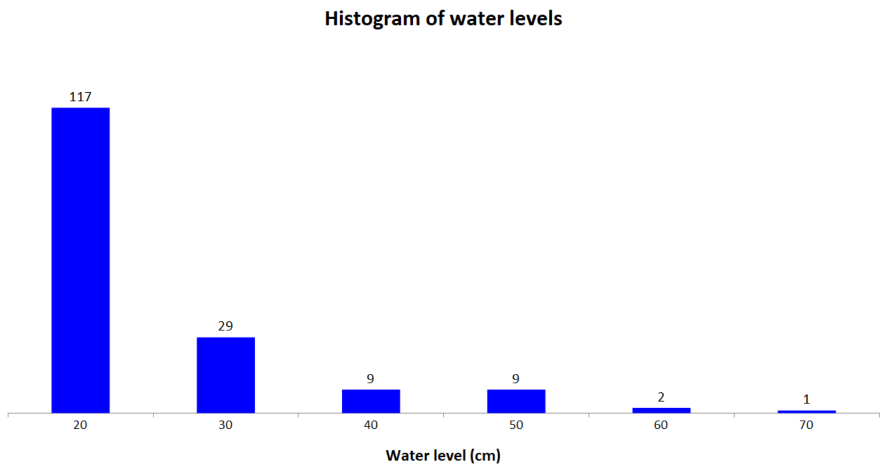

Figure A4. Even though the heavy rain occurred just during few hours (3:00 -5:00 PM), simulation result shows that there are 50 points where water levels are from 30 to 70 cm.

Our proposed simulator can be efficient to predict risk places where flooding due to heavy rain often occurs.

In fact, flood simulation is utilized the frame tool flow as what being used for radio signal propagation simulation and radio coverage prediction as explained in Section 3. For this study time step was one hour, we keep β at 20%, α_t rainfalls were taken from public website shortly after the episode.

It leverages further intensive studies for reducing the risk of natural disasters, especially in complex topographical areas. It gives many benefits for the human life of local residents, as well as economic activities by informing of the risks of their position regarding exceptional events.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1

Peer-reviewed version available at Sensors 2018, 18, 2323; doi:10.3390/s18072323

Appendix .2 Experimental measurements of radio coverages

Figure A5. A base station is equipped a LoRa board (emitter) and MacBook. This base station was deployed on the top of Roc'h Trevezel mountain, France. For receiver, another LoRa board is mounted on a car moving around.

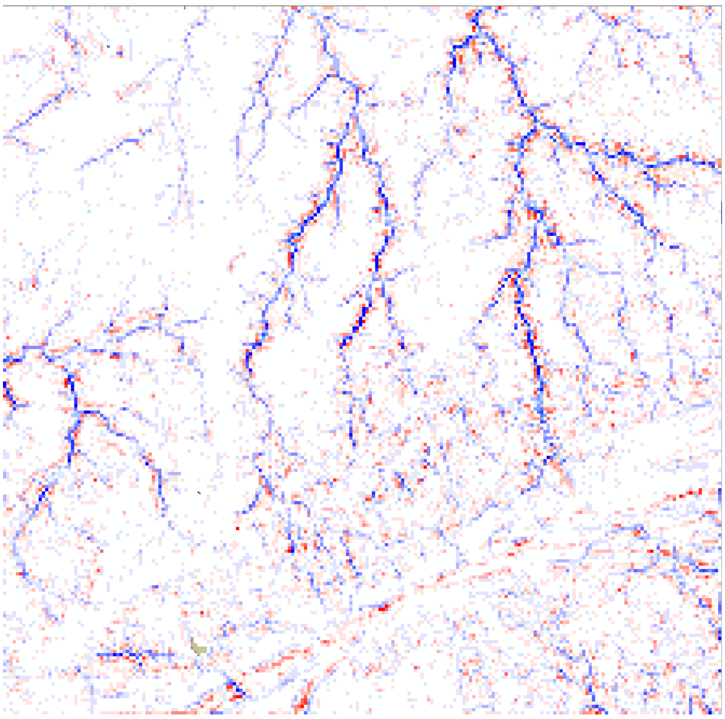

Figure A6. Terrain complexity analysis for the Roc'h Trevezel using TPI metric. This shows remarkable landform splits in the North-South direction. This provides critical references for network deployment aiming to enhancing the robustness and range of system and reducing the cost as well.

Some case studies were investigated with the credible installation position for *base stations* (BS) controlling hundreds of sensors. In each case, the signal strength was recorded with a picture

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 June 2018 doi:10.20944/preprints201806.0289.v1
Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

showing the BS environment. To evaluate the impact of transmission environment on reliability and robustness of radio links in the context of wide-range wireless networks, several places with different terrains in Bretagne region, France were selected for our intensive investigations. In the case as shown in Figure A7 have required a visit to *Arrée Mountains* in central Brittany. These mounts culminate at less than 400 m and are the site chosen for TV/Radio antenna. The car was driven in a loop, up to 10 km showing a good accuracy of simulation prediction.

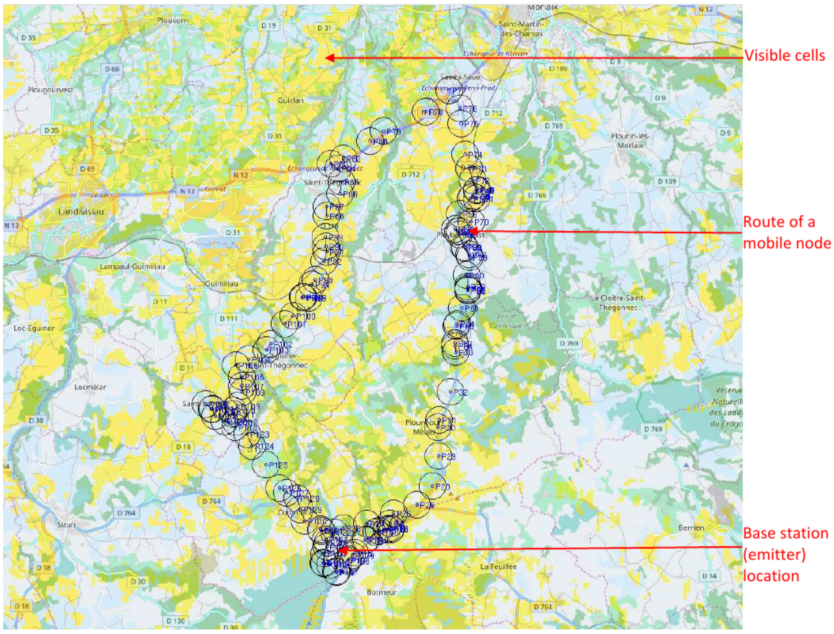

Figure A7. The base station (emitter) is located on top of the mountain (latitude: 48.4051306, longitude: -3.9077762, elevation: 345 m). A receiver is placed on a car (receiver) traveling around the area with many hills, valleys, and big trees. From simulation results, places (cells) are able to receive the radio signal highlighted in yellow color. Blue circles show points where the receiver got messages successfully in our actual measurement.

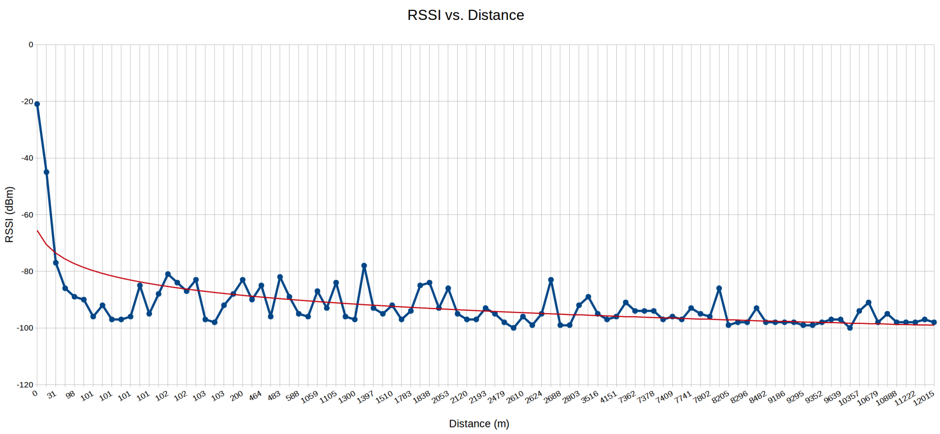

Figure A8. Obtained RSSI values for experiment in our actual measurement at the Roc'h Trevezel. RSSI values decay as a negative exponential function of distance. The complex terrains in this area cause serious impact on quality of radio links.

Table A1. Three areas with different complexity terrains. *Resolution* and *Actual size* columns give the size in pixels associated with actual size in meters of each cell in regular grid data. The maximum communication in each experiment is shown by column *Range*. *Match* and *Mismatch* column provides a number of visible points which are right matched and mismatched respectively between simulation results and obtained values in real measurements.

Area name	Description	Resolution	Actual size	Range	Match	Mismatch	% Error
Albert 1^{er}	Urban area	5×5 pixels	24 m	3 km	64	13	20.21%
Plougastel	River and its banks	5×5 pixels	96 m	9 km	129	4	3.10 %
Trevezel	mountain area	5×5 pixels	191 m	11 km	45	8	17.78%

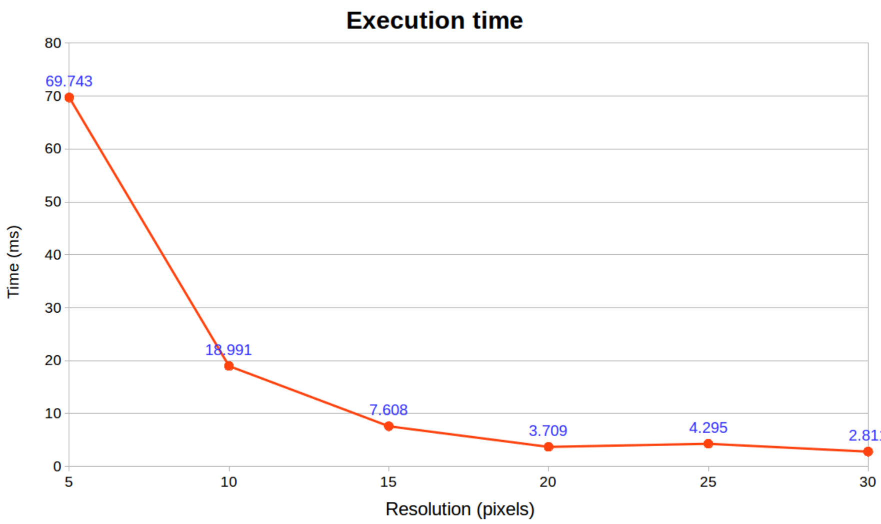

Figure A9. Chart of radio coverage execution times for different resolutions of cell from 5 to 30 pixels. This statistics was recorded for the performance on PC equipped Core i7-7700K CPU@4.20GHz x 8, 16 GiB DDRAM, card NVidia GeForce GTX 1070 (1920 CUDA cores).

References

1. Callaway, E.; Gorday, P.; Hester, L.; Gutierrez, J.A.; Naeve, M.; Heile, B.; Bahl, V. Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine* **2002**, *40*, 70–77.
2. Corporation, S. Sx1276/77/78/79 -137 MHz to 1020 MHz Low Power Long Range Transceiver. Rev. 4 - March 2015. Technical Report August, 2016.
3. Sigfox. Sigfox Technology Overview. https://www.sigfox.com/en/sigfox-iot-technology-overview, 2017. [Online; accessed on 15 December 2017].
4. Zhang, J.; Atkinson, P.M.; Goodchild, M.F. *Scale in Spatial Information and Analysis*; 2014; p. 362.
5. Riley, S.J.; DeGloria, S.D.; Elliot, R. A Terrain Ruggedness Index that Qauntifies Topographic Heterogeneity, 1999.
6. Jenness, J. Topographic Position Index (TPI) v. 1.2 **2006**. p. 42.
7. Huuskonen, A.; Saltikoff, E.; Holleman, I. The Operational Weather Radar Network in Europe. *Bulletin of the American Meteorological Society* **2014**, *95*.
8. EUMETSAT. Monitoring The Atmosphere, Ocean and Climate from Space - Transforming Our World. https://www.eumetsat.int/website/home/AboutUs/Publications/Brochures/index.html, 2017.

9. Online, W.W. Bejaia Historical Weather. https://www.worldweatheronline.com/bejaia-weather-history/bejaia/dz.aspx, 2017. [Online; accessed on 25 July 2017].
10. proj4.org. Proj4 documentation. https://proj4.org, 2018. [Online; accessed on 4 June 2018].
11. Wikipedia.org. Bejaia Historical Weather. https://en.wikipedia.org/wiki/Web_Mercator, 2018. [Online; accessed on 25 April 2018].
12. Lucas, P.Y. Modélisations, simulations, synthèses pour des réseaux dynamiques de capteurs sans fil. PhD thesis, Université de Brest (UBO), LabSTICC, UMR CNRS 6285, 2016.
13. Lucas, P.Y.; Van Long, N.H.; Truong, T.P.; Pottier, B., Wireless Sensor Networks and Satellite Simulation. In *Wireless and Satellite Systems: 7th International Conference, WiSATS 2015, Bradford, UK, July 6-7, 2015. Revised Selected Papers*; Pillai, P.; Hu, Y.F.; Otung, I.; Giambene, G., Eds.; Springer International Publishing: Cham, 2015; pp. 185–198.
14. Pridal, K.P. Google Summer of Code 2008, project GDAL2Tiles for OSGEO. http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/, 2011. [Online; accessed on 25 April 2018].
15. Neumann, J.V. *Theory of Self-Reproducing Automata*; University of Illinois Press: Champaign, IL, USA, 1966.
16. Frisch, U.; d'Humieres, D.; Hasslacher, B.; Lallemand, P.; Pomeau, Y.; Rivet, J.P. Lattice Gas Hydrodynamics in Two and Three Dimensions. *Complex Systems Publications* **1987**, *1*, 649–707.
17. Margolus, N.; Toffoli, T. *Cellular Automata Machines: A New Environment for Modeling*; MIT Press, 1987.
18. Lynch, N. *Distributed algorithms*; Morgan Kaufman, 1996.
19. Pountain, D.; May, D. *A tutorial introduction to Occam programming*; McGraw-Hill, Inc.: New York, NY, USA, 1987.
20. Iqbal, A.; Pottier, B. Meta-Simulation of Large WSN on Multi-core Computers. Proceedings of the 2010 Spring Simulation Multiconference; Society for Computer Simulation International, , 2010; SpringSim SCS Conference, p. 133.
21. Barnes, F.R.M. Blocking System Calls in KRoC/Linux. Communicating Process Architectures 2000; Welch, P.H.; Bakkers, A.W.P., Eds., 2000, pp. 155–178.
22. Dutta, H.; Failler, T.; Melot, N.; Pottier, B.; Stinckwich, S. An execution flow for dynamic concurrent systems: simulation of WSN on a Smalltalk/CUDA environment. Proceedings of SIMPAR 2010 Workshops Intl. Conf. on simulation, modeling and programming for autonomous robots; , 2010; Dynamic languages for robotic and sensors systems (DYROS), pp. 290 – 295.
23. Keita, E.B. Physical models and perception, contributions to sound analysis in urban environment. Phd thesis, Université de Bretagne occidentale - Brest, 2015.
24. Traore, M. Modélisation cellulaire et simulation physique: contribution à l'analyse de la dynamique de population des insectes ravageurs. PhD thesis, Université de Bretagne occidentale - Brest.
25. Météo-France. Le modéle á maille fine Arome. http://www.meteofrance.fr/prevoir-le-temps/la-prevision-du-temps/le-modele-a-maille-fine-arome, 2018.
26. Faroux, S.; Kaptué Tchuenté, A.T.; Roujean, J.L.; Masson, V.; Martin, E.; Le Moigne, P. ECOCLIMAP-II/Europe: a twofold database of ecosystems and surface parameters at 1 km resolution based on satellite information for use in land surface, meteorological and climate models. *Geoscientific Model Development* **2013**, *6*, 563–582.
27. Bresenham, J.E. Algorithm for Computer Control of a Digital Plotter. *IBM Syst. J.* **1965**, *4*, 25–30.
28. Rappaport, T.; (Firme), S.B.O. Wireless communications: principles and practice, 1996.
29. Seybold, J.S. *Introduction to RF Propagation*; 2005; pp. 1–330, [0712.0689].
30. Rappaport, T. *Wireless Communications: Principles and Practice*, 2nd ed.; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2001.
31. Seybold, J.S. *Introduction to RF Propagation*; 2005; pp. 1–330, [0712.0689].
32. Manuka. Balloon project. http://www.instructables.com/id/Introducing-LoRa-/, 2015. [Online; accessed on 25 July 2017].
33. SAMES group, F.; Vietnam.; Indonesia. Stic-Asia Modeling for Environment Simulation (SAMES). http://sames.univ-brest.fr, 2016. [Online; accessed on 31 August 2017].

Peer-reviewed version available at *Sensors* **2018**, *18*, 2323; doi:10.3390/s18072323

Sample Availability: Sample simulation on Linux : ask authors