Flexible querying of Volunteered Geographic Information for risk management

Paolo Arcaini[1] **Gloria Bordogna**[1] **Simone Sterlacchini**[1]

[1]CNR – Institute for the Study of the Dynamics of Environmental Processes (IDPA)

Abstract

The paper presents an approach to manage volunteered geographic information (VGI) to point out anomalous conditions of the environment to help administrators in charge of the governance and maintenance of the territory to plan mitigation and safeguard interventions. To this end they can formulate flexible queries on the VGI reports to analyze their contents. The novelty of the proposal is the search framework of VGI reports designed to support distinct needs, among which the assessment of VGI quality which is an important issue in such applications. Flexible queries are formulated and evaluated within a fuzzy database approach.

Keywords: environmental risk, Volunteered Geographic Information, fuzzy databases, flexible querying, VGI quality assessment

1. Introduction

Citizen science is a term currently used to describe the scientific efforts that are based on the contribution of citizen volunteers engaged in collecting crucial information for the project, which would not be possible to acquire based solely on the commitment of scientists [15]. These contributions can be collected due to the wide availability of computers and mobile devices connected to the Internet. Volunteers are asked to make contributions of various types, which include georeferenced reports, i.e., Volunteered Geographic Information (VGI) [13], on anomalous environmental conditions like in the project "*Did You Feel It?*"[1].

The many citizen science projects that collect contributions in the form of VGI use crowdsourcing platforms such as *Ushahidi* [18], accessible over the Internet, for acquiring, storing and representing the VGI reports, which are visualized by post-its on a map of the territory. To analyze the contents of the VGI reports one has to click on the post-its shown on the map, while querying facilities on the reports contents are generally missing. Nevertheless, in practical applications, VGI reports contents need to be analyzed and filtered with respect to several information needs, serving distinct purposes. For example, in our context, VGI provided by citizens has the objective of pointing out anomalous conditions of the territory and requests of mitigation and safeguard interventions. Analysing the reports contents can aid to plan mitigation interventions, to monitor areas that are susceptible to specific kinds of risks (such as landslides, wildfires, etc.), and finally to assess the quality of the VGI reports themselves which is one of the major concern when using VGI [10, 14]. In this paper we propose a set of flexible query types and their evaluation mechanism defined within the framework of fuzzy databases [4, 5, 12] in order to effectively analyze VGI. The motivation of this approach is that VGI reports are often affected by imprecision both in the georeference and content. Further, the operators in charge of managing the territory are not expert in formal query languages and thus they need practical ways to formulate their needs. Specifically, in Section 2 we summarize the application context and in Section 3 the system architecture and types of flexible queries. In Section 4 the data model is defined and in Section 5 the quality assessment method of VGI reports is presented. Section 6 reports some related works and Section 7 summarizes the main content and novelty of the proposal.

2. The application context of the proposal

The SISTEMATI project[2] is developing a distributed communication system to inform the operators responsible for the governance and maintenance activities of the territory on the anomalous situations that need assistance for the prevention of environmental and industrial risks in the pre-alert phases. This is done to prevent the occurrence of emergency situations. To this end, the project uses the geographic information provided by volunteer citizens (VGI) who are so involved in land management in the early stages of defining the priority areas of intervention. In local geoportals connected to the Internet the municipalities receive, record and display on a map VGI reports created by their citizens witnessed of abnormal situations. This local acquisition of VGI allows the municipalities to cross-check the VGI authorships with personal data in order to validate them. These reports, along with the date of their creation and the spatial position may include free text annotations, audio and video recordings of events, categories and risk levels ob-

[1]http://earthquake.usgs.gov/earthquakes

[2]http://www.idpa.cnr.it/sistemati.htm

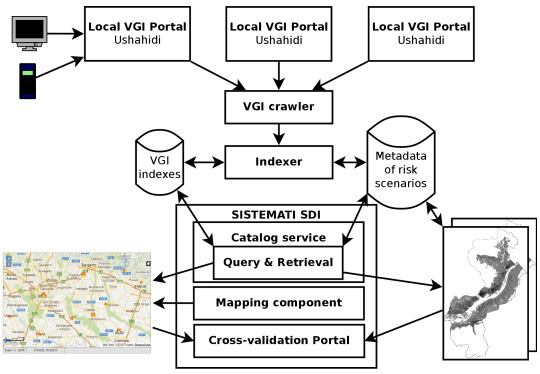

Figure 1: SISTEMATI framework

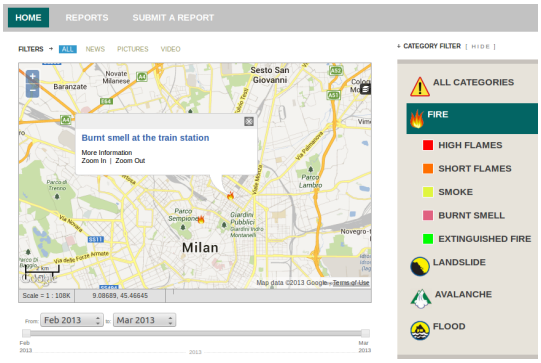

Figure 2: Municipal VGI report geoportal based on Ushahidi

served in the vicinity of the locations. These local geoportals are connected to a central Spatial Data Infrastructure (SDI) that provides querying facilities of VGI reports of all municipalities. The operator can carry out various searches specifying various kinds of selection conditions of the reports:

- *free text* to select reports requiring specific interventions, such as requests of repairing of crumbling roads, cleaning river banks, etc., in order to plan actions of safeguard and mitigation;
- *categories of users*, such as citizens, civil protection volunteers, etc., and *levels of risk*, such as reports of fires with sighting of flames, to monitor the response to an emergency;
- *space-time conditions*, to restrict the analysis to the VGI reports located in regions susceptible to risks, for example, to select reports in the vicinity of areas with a high density of potentially polluting industries;
- *space-time density* conditions for the selection of areas with a minimum number of reports that meet one or more of the above mentioned conditions.

The conditions on the space-time density of reports can be useful to validate the contents of the records themselves. In fact, one of the primary problems in the use of VGI is estimating the quality of the information that depends on several factors, such as the reputation of the source, the truthfulness and accuracy of the contents [10, 14]. A method for estimating the quality may be based on a comparison between the textual contents of the VGI reports close to each other both in space and in time. A high density of records that point out the same risk in a given region at the same time reinforces the truthfulness of their contents, and thus increases the level of attention of the operator responsible for the management of interventions in the area.

3. The SISTEMATI system architecture

We propose a distributed framework, named SISTEMATI, depicted in Figure 1, in which each municipality (at the local level) hosts an independent *Ushahidi* installation [18]. These local systems are connected to a central Spatial Data Infrastructure (SDI) (at the regional level) implemented as an extension of the *Geonetwork* [22] and *Geoserver* [9] open source systems.

Field	Mandatory
Latitude, Longitude	Yes
Date	Yes
Title	Yes
Description	Yes
User category	Yes
Risk classification	Yes
Images	No
Videos	No
Links to web-pages	No
Comments	No

Table 1: SISTEMATI VGI report fields

Citizens belonging to a municipality (hereafter named *users*) can connect to a local *Ushahidi* server and submit reports either about dangers they have witnessed or about requests of mitigation interventions (see Figure 2). For example, a user could signal the burnt smell he perceives at the railway station. This can be done by sending a report from his/her smartphone.

Each report contains at least the geographic footprints represented by either a georeferenced polygon or, in the simplest case, a point provided by coordinates (latitude and longitude) of the place where the risk has been observed, the date of the report submission, a title, a description, and a classification of the risk. Table 1 presents a complete description of the report fields we have identified in the personalization of the *Ushahidi* platform for the SISTEMATI project.

Users are categorized according to their role (e.g., unregistered citizens, citizen belonging to the civil register of the municipality, firemen, etc.). The distributed *Ushahidi* servers are connected through the Internet to the SISTEMATI SDI (compliant with INSPIRE SDI architecture[3]), which is made available at the staff in charge of managing the territory; the SDI consists of three main components: a

[3] http://inspire.jrc.ec.europa.eu/

VGI *crawler*, a *catalog service*, a *mapping component* and a *cross-validation* component. The VGI *crawler* periodically collects all the remote VGI reports hosted on the different municipality servers and builds metadata for them [17]. Such metadata, which comprise the textual content of the VGI and the centroid of the geographic footprints, are organized in a catalog and indexed in a collection of VGI indexes so that they can be queried by the administrative staff (*operators*) based on distinct selection conditions through the geoportal *catalog service* of the SDI. The SDI also allows accessing the contents of the VGI reports and their remote geographic footprints on the municipality servers by the *mapping component* through their links. Finally the SDI provides WFS and WMS services to allow accessing remote spatial data. In the following subsection we introduce the querying facilities.

3.1. Querying facilities

We want to allow the administrative operators in charge of the management of the territory to express in a simple way as well as to execute different kinds of queries over the VGI reports:

- *Spatial range queries*: one can demand to retrieve all the VGI reports whose geographic footprints overlap a specified region specified by a (fuzzy) Bounding Box (BB). We must permit the operator to execute also a search across the tiling regions identifying the administrative boundaries of the municipalities with the local installations of *Ushahidi*.
- *Content-based queries*: the textual fields of the reports (e.g., title, description, ...) must be searchable in full-text mode and retrieved based on the matching of their contents with respect to the operator textual query.
- *User category-based queries*: all the reports of a given user category (or a group of user categories) can be retrieved based on a controlled vocabulary of categories.
- *Risk level-based queries*: all the reports of a given risk type and risk level (or a group of risk levels) can be retrieved.
- *Temporal queries*: all the reports that have been submitted in a given temporal (fuzzy) range can be retrieved.
- *Spatial/temporal density queries*: the operator should be able to specify (fuzzy) density thresholds (e.g., about 10 reports/km^2, about 5 reports/hour, etc.) to select the reports belonging to regions (*spatial clusters*) and/or temporal intervals (*temporal clusters*) in which the density is greater than the (fuzzy) threshold. This can be useful to further analyze if the selected reports deal with the same events, risk, requests and to validate their quality as it will be better described in Section 5.

One may require that the results are retrieved with some tolerance of the selection conditions, which can be formulated by specifying linguistic terms defining soft constraints, and thus admitting degrees of satisfaction to the query (the relevance degree) [6]. These flexible queries are modeled by adopting the fuzzy database framework [4, 5, 12]. The retrieved reports can be presented in decreasing order of their relevance to the query or can be visualized by post-its with distinct size on the map, so that the greater the relevance, the bigger the post-it. Tolerant queries should be available for spatial, temporal-based and spatial/temporal density queries as defined in the following subsections [3].

3.1.1. Spatial range query

This type of query compares the centroid of the geographic footprint of VGI reports with the bounding box specified by the operator. The Bounding Box (BB) can be defined with a fuzzy boundary so that reports contained in the inner boundary have relevance value equal to 1, whereas those outside the outer boundary have a null relevance, and those between the two boundaries get a relevance value inversely proportional to their distance from the inner boundary. The fuzzy boundary can be specified by the operator who may draw the inner boundary by a bounding box on the map and may specify a tolerance on the distance from it where the reports become totally irrelevant, thus implicitly defining the outer boundary. A practical way is to choose a buffer with a given size δ and draw the BB. This is translated into the coordinates (x_{sw}, y_{sw}) of the SW point and the coordinates (x_{ne}, y_{ne}) of the NE point, with δ indicating the required tolerance of the query (distance from the BB); if the buffer is not specified, we assume $\delta = 0$, i.e., the query is crisp and only reports strictly contained in BB must be retrieved. Figure 3 shows an example of BB with the buffer indicating the tolerance δ.

We define the BB centroid vector as follows:

$$C = (x_c, y_c) = \left(\frac{x_{sw} + x_{ne}}{2}, \frac{y_{sw} + y_{ne}}{2} \right)$$

Being (x_r, y_r) the coordinates representing the geographic footprint of a report r, we introduce f as follows:

$$f(r, BB) = \max \begin{pmatrix} |x_r - x_c| - (x_c - x_{sw}), \\ |y_r - y_c| - (y_c - y_{sw}) \end{pmatrix}$$

$f(r, BB)$ is negative if r is contained in the BB, $f \in [0, \delta]$ if r belongs to the fuzzy boundary, and $f > \delta$ if r is outside both BB and the fuzzy boundary.

The degree of satisfaction (*Relevance Status Value* (RSV)) of a tolerant spatial range query is computed by the following function:

$$RSV = inBB(r, BB, \delta) = \begin{cases} 1 & if \quad f(r, BB) \leq 0 \\ \frac{\delta - f(r,BB)}{\delta} & if \quad 0 < f(r, BB) < \delta \\ 0 & otherwise \end{cases} \tag{1}$$

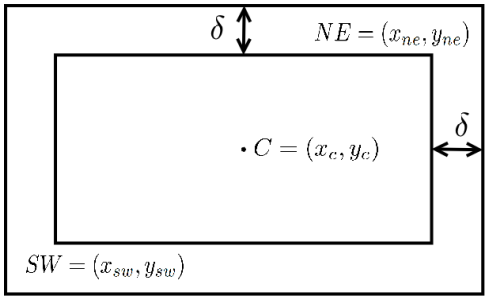

Figure 3: Bounding box with fuzzy boundary

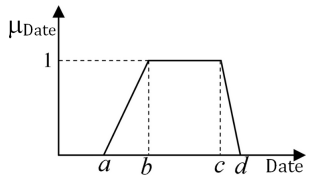

Figure 4: Soft temporal constraint

3.1.2. Temporal range query

The operator can formulate tolerant temporal queries by specifying linguistic values defining soft temporal constraints on the dates [8] of the reports with trapezoidal shaped membership functions μ_{Date} as the one shown in Figure 4.

Trapezoidal functions are defined by providing four values (a, b, c, d) with $a \leq b \leq c \leq d$ on the domain *Date*. The evaluation mechanism of these queries computes as RSV the degree of satisfaction of the soft constraint by the report date d as $RSV = \mu_{Date}(d)$, where μ_{Date} is defined as in formula 2 below.

3.1.3. Density-based query

The operator can formulate tolerant density-based queries by soft constraints on either the spatial density or temporal density of the reports. Thus the results of these queries are ranked reports, each satisfying the request to a given degree, according to the density of the clusters they belong to.

Notice that, as it will be described in Section 4, the reports are previously clustered and may belong to several clusters, each one with a distinct spatial, temporal and spatial-temporal density. The soft constraints are defined with piecewise linear membership functions $\mu_{Sdensity}$ and $\mu_{Tdensity}$, for the spatial and temporal densities respectively. The evaluation mechanism of these queries computes a RSV for each cluster of reports and, thus, each report may have several RSVs since it can belong to distinct clusters. We take the maximum of the RSVs to rank the reports.

3.1.4. Combined queries

To select the reports, the operator may specify combined queries by multiple soft constraints of distinct kind. In this case, the RSVs are combined by considering the simultaneous satisfaction of the constraints, thus by taking the minimum of the RSVs.

3.2. Visualizations facilities

At the regional level we also build a global VGI map displaying all the reports of the single municipalities. Each metadata of a single report contains a link to the global map that zooms in the area of the corresponding report: in this way a single report can be analyzed together with the (maybe related) surrounding reports. Each metadata also contains some fields of the original report, and a link to the original report with its geographic footprint on the *Ushahidi* municipality server. The central SDI also permits to graphically browse the reports, filtering them by risk level and user category.

4. Data model

In order to describe the structure of the reports organized into fields whose values are defined on distinct domains, and the operator types that can be used for querying them, we define the data model.

4.1. Basic data model

Let $U = \{u_1, \ldots, u_n\}$ be a set of municipalities, each hosting an installation of *Ushahidi*. From now on, we only consider a single $u \in U$.

Let $UC = \{uc_1, \ldots, uc_m\}$ be the set of user categories (e.g., *citizen*, *administrator*, *officer*, ...).

Let $RC = \{rc_1, \ldots, rc_s\}$ be the set of risk categories (e.g., *fire*, *flood*, *avalanche*, ...). Each risk category is further divided in levels of risk; for each risk category $rc_i \in RC$ we define its corresponding set of risk levels $RL_i = \{rl_{i1}, \ldots, rl_{ih}\}$. For example, for the risk category *fire*, we can identify the risk levels {*high flames*, *low flames*, *smoke*, *burnt smell*, *extinguished fire*}.

We define $RL = \cup_{i=1}^{s} RL_i$ as the set of all the risk levels of all the risk categories.

Let R_u be the set of reports of a single municipality $u \in U$ and $R = \cup_{u=1}^{n} R_u$ the set of all reports. Each report $r \in R$ is associated with the category uc_k of the user who submitted the report, and with a risk level rl_{ij}. We define the mappings:

$$getUc: R \rightarrow UC \qquad getRc: R \rightarrow RC \qquad getRl: R \rightarrow RL$$

that retrieve, from a report, the associated user category, risk category, and risk level, respectively. Note that the correct mapping *getRl* must guarantee that, for any $r \in R$, if $getRC(r) = rc_i$ (with $rc_i \in RC$), then $getRl(r) \in RL_i$.

4.2. Density classification model

Let D_{dens} be the domain of values of the densities; they can be number of reports over units of either time, or space, or both.

Let $CLASS$ be a set of classifications of reports in clusters with the same minimum density $\Delta \in D_{dens}$.

Function

$$dens : CLASS \rightarrow D_{dens}$$

indicates, for each classification, the minimum density of the clusters of the classification. We require the function to be injective, since we need to distinguish the classifications by their minimum density.

So, a classification $class_\Delta \in CLASS$ (with $dens(class_\Delta) = \Delta$) is defined as follows:

$$class_\Delta \subseteq 2^R \wedge \forall cl_1, cl_2 \in class_\Delta : cl_1 \cap cl_2 = \emptyset$$

where cl_1 and cl_2 are clusters of $class_\Delta$ and their densities are greater than or equal to Δ.

Notice that, given a classification $class_\Delta$, a single report may be not part of any cluster $cl \in class_\Delta$; all these reports are marked as *noise* reports of the classification and collected in the set $noise_\Delta$. So, the set $\{\{cl_i \in class_\Delta\}, noise_\Delta\}$ establishes a partition of R. Note that a classification for a given Δ may be empty, i.e., $noise_\Delta = R$.

Function:

$$getDensReports : CLASS \rightarrow 2^R$$

retrieves, for each classification, the reports clustered by the classification (i.e., $getDensReports(class_\Delta) = R \backslash noise_\Delta$).

4.3. Intensional definition

We now introduce the notation used to define the schema of a report:

- F the set of fields names in which a report can be structured;
- C={*spatial*, *temporal*, *textual*, *link*, *userCategory*, *riskLevel*} the set of possible kinds of the fields;
- D the set of possible primitive data types. The primitive data types contained in D can be *Boolean*, *String*, *Integer*, *Real*, *Date*, *Url* and enumerative domains (e.g., user category, risk category, risk level);
- $fc : F \rightarrow C$ a function that identifies, for each field name $n_i \in F$, a kind $c_j \in C$;
- $fd : F \rightarrow D$ a function that identifies, for each field name $n_i \in F$, a primitive data type $d_j \in D$;
- $fm : F \rightarrow \{1, *, +\}$ a function that signals the multiplicity of the field; "1" means that one single occurrence of the field is mandatory, "*" that 0 or more occurrences of the field can be present, and "+" that at least one occurrence of the field is required;
- OP is a set of operator types $\{OP_{text}, OP_{rel}, OP_{sel}, OP_{BB}, OP_{null}\}$ where
  - $op_i \in OP_{text}$ is defined for $D = String$ and gives values in [0,1]. It is based on statistical functions of the occurrences of terms in the reports as in information retrieval systems [20].
  - $op_i \in OP_{rel}$ is defined for a numeric domain D and gives values in [0,1]. It is the membership function of a soft constraint with trapezoidal shape defined by four values (a, b, c, d) with $a \leq b \leq c \leq d$ belonging to D. The definition of the function is as follows:

$$op_i(x) = \begin{cases} 0 & if \quad x \leq a \vee x \geq d \\ \frac{x-a}{b-a} & if \quad a < x < b \\ 1 & if \quad b \leq x \leq c \\ \frac{d-x}{d-c} & if \quad c < x < d \end{cases} \tag{2}$$

    Figure 4 shows an example of membership function of a trapezoidal soft constraint for the field *Date*.
  - $op_i \in OP_{sel}$ is defined for an enumerative domain: functions *getUc*, *getRc*, and *getRl* are of this type;
  - $op_i \in OP_{BB}$ is defined for a spatial bidimensional real domain $D \times D$ and gives values in [0,1]. One of such functions has been defined in the Section 3.1.1 by formula 1;
  - OP_{null} is the empty set of operations, i.e., no operation must be done on the field.

The intensional definition of a report, i.e., its schema, is a sequence of tuples such as:
$Ir = [< name \in F, kind \in C, value \in D, opType \in OP, m \in \{1, *, +\} > *]$

For example, in our application, the reports have the following schema:
$[< geofoot, spatial, Real \times Real, OP_{BB}, 1 >,$
$< date, temporal, Date, OP_{rel}, 1 >,$
$< user, userCategory, UC, OP_{sel}, 1 >,$
$< riskName, riskLevel, RL, OP_{sel}, 1 >,$
$< title, textual, String, OP_{text}, 1 >,$
$< description, textual, String, OP_{text}, 1 >,$
$< comment, textual, String, OP_{text}, * >,$
$< image, link, Url, OP_{null}, * >,$
$< video, link, Url, OP_{null}, * >,$
$< reference, link, Url, OP_{null}, * >]$

An instance of this schema is a concrete report containing values for the fields, respecting the defined data types and multiplicities. An example of concrete report is as follows:
[*(45.48561, 9.203826), Tue Mar 05 11:48:00 CET 2013, officer, burnt smell, "Burnt smell at the train station", "Strong burnt smell near the ticket office"*]

4.4. Metadata

We recall that, at the regional level, a *crawler* collects all the reports hosted on the different *Ushahidi* installations and builds metadata for them. Such metadata are indexed in a collection of *VGI indexes*.

We have identified different policies for building the metadata in order to more efficiently answer the queries specified in Section 3.1.

A first solution is to build a metadata m_r for each single report r, by collecting all contents of fields of

r such that the metadata schema is the following:

$$m_r = \left[\left\langle \begin{matrix} Ir.name \in F, \\ Ir.value \in D \end{matrix} \middle| \begin{matrix} Ir.opType \neq OP_{null} \wedge \\ Ir.kind \neq link \end{matrix} \right\rangle \right], ush(r)$$

where ush is a function that provides the URL of the report on the local *Ushahidi* platform. So, in the metadata of a report, we include all the fields contents that are searchable. Such an approach has the advantage of providing fine results to queries, since the single report can be individually identified and retrieved.

Another solution is to build metadata for a particular class (or set) of reports. For example, we can build metadata m_{ijk} for all local installations of *Ushahidi* and each risk category i, risk level j and user category k, as follows:

$$m_{ijk} = \left\{ m_r \middle| \begin{matrix} r \in R \wedge getRl(r) = rl_{ij} \wedge \\ getUc(r) = uc_k \end{matrix} \right\} \quad (3)$$

Instead, if we are interested in classifying reports by their user category uc_k, we can build metadata in the following way:

$$m_k = \{m_r | r \in R \wedge getUc(r) = uc_k\} \quad (4)$$

Note that these are only two of the possible metadata that can be generated for groups of reports. For example, to quickly answer density-based queries, we can build a single metadata for all reports belonging to a classification *class* with a given minimum spatial (Δ_s), temporal (Δ_t) or spatial-temporal (Δ_{st}) density:

$$m_\Delta = \{m_r | r \in R \wedge r \in getDensReports(class)\} \quad (5)$$

where $class \in CLASS$ and $dens(class) = \Delta$.

The solution of building metadata for groups of VGI reports has the advantage that user category-based, risk category-based, risk level-based queries and density-based queries can be answered very quickly, since the global metadata already contain all the requested reports.

4.5. Data structure for VGI retrieval

In order to efficiently evaluate the different kinds of queries described in Section 3.1, we must organize the metadata fields into distinct data structures to access the information optimized with respect to the kinds of queries.

In order to answer to *user category-based* and *risk level-based queries*, functions $getUc$ and $getRl$ can be used to generate metadata for the group of reports of same category and/or risk level, by applying formulae such as 3 and 4. As said previously, metadata already grouping reports by user category and/or risk level can be built in advance, so that reports belonging to the same group can be analyzed together. Since the cardinality of the categories and the kinds of risk and risk levels are small, these metadata can be organized in a table in alphabetical order.

Content-based queries, instead, can be handled by considering metadata for each single report and by building an inverted index for the textual fields as in information retrieval [20]. We have dealt with the indexing and querying of textual fields of the reports by using Lucene library [16].

For efficiently evaluating *spatial range queries*, the spatial field can be organized in a spatial data structure such as an R-tree optimized for answering range queries [7].

For *temporal-based queries*, the metadata can be ordered in increasing value of their field *Date*.

Spatial/temporal density queries can be handled by organizing metadata as shown in formula 5. To this end, in a preprocessing step, a clustering algorithm such as DBSCAN [11] is used to group reports according to their mutual spatial or temporal distance. The DBSCAN algorithm permits to generate classifications in which the reports are grouped into possibly non-convex clusters having a specified minimal spatio-temporal density. By several runs of DBSCAN, we can build distinct classifications $class_\Delta$ for different values of density Δ. Each classification is associated with a single metadata m_Δ. The operator can perform three kinds of density-based queries:

- *available clusters query*: the operator chooses one of the spatio-temporal density values suggested by the system, associated with the precomputed classifications; in this case the reports in the correspondent metadata m_{Δ_i} are retrieved;
- *exact threshold query*: the operator specifies a threshold Th on the spatio-temporal density and the framework retrieves the reports belonging to the correspondent classifications whose density is over the threshold; in this case we retrieve the reports linked in the metadata m_{Δ_i} with the minimal density above the threshold. Given the metadata $\langle m_{\Delta_1}, \ldots, m_{\Delta_n} \rangle$, ordered by the increasing order of Δ, we select m_{Δ_i} such that

$$\Delta_i \geq Th \wedge \forall j \in [1, i-1] : \Delta_j < Th$$

- *fuzzy threshold query*: the user specifies a soft constraint, for example expressed by linguistic terms such as *low density*, *medium density*, and defined with a piecewise membership function μ_{FTh} on the domain of the spatio-temporal density [0,1]. This can be done by specifying a trapezoidal membership function as shown in formula 2. Since a report r can belong to more classifications $class_{\Delta_i}$ (with $i = 1, \ldots, n$), with increasing density satisfying the fuzzy threshold, its degree of relevance is computed as the maximum satisfaction of the soft constraint by the distinct densities of the retrieved classifica-

tions $class_{\Delta_i}$:

$$RSV(r) = \max(\mu_{FTh}(\Delta_1), \ldots, \mu_{FTh}(\Delta_n))$$

5. Validating VGI reports

When dealing with VGI, there is the problem of validating its quality. Indeed a user may report a false warning for different reasons:

- (s)he misuses the platform (e.g., (s)he wrongly classifies a risk report by accidental clicking on the wrong checkbox);
- (s)he overestimates/underestimates an observed risk;
- (s)he could be a malicious user who reports false warnings on purpose.

In order to validate the provided information, we can use two different kinds of techniques, *ex-ante* and *ex-post*, distinguished by the time when the information is validated [2].

Ex-ante techniques aim to prevent the creation of low quality VGI by supporting the user in the creation of correct reports:

- only particular classes of trusted users can submit reports;
- a report form is provided to help the user in providing a correct information. For example, if the user wants to submit information about a fire, the system may show some examples to be used for comparison with the observed fire. In such a way the user is supported in identifying the risk level.

Ex-post techniques, instead, remove low quality reports or classify the reports after these have been produced:

- the user authority and category can be used to assign a value to the reports. Warnings provided by registered users are considered more trustworthy than those provided by unregistered users. Moreover, among the registered users, reports of expert users (e.g., firemen) could be more considered than those of volunteers;
- a risk forecasting system could be used to validate the reports. The risk forecasting system, given the current values of some dynamic parameters (e.g., temperature, humidity, ...), furnishes a risk map representing the scenario probability that a given natural/environmental disaster can happen in the area of interest. Its cross-validation with VGI can lead to two situations:
  - *Inconsistency*: VGI reports about risks occur within areas that are considered safe by the risk maps. A single risk report in a safe area may be considered unreliable. A high density of reports in a safe area, instead, demands for a control and possible revision of the risk map.
  - *Concurrency*: risk reports occur within areas considered risky by the risk maps. High density of VGI annotations confirms the reliability of the risk map.

6. Related work

The first contribution of our proposal is the use of VGI in emergency management. The definition of VGI can be applied to different scenarios in which the users contribute in different ways. In projects as BOINC [1], a computationally expensive process is parallelized in different sub-processes executed by the users that make available the computing time of their PCs. In other projects the users make some tasks in which humans outperform computers, as shape classification of galaxies in the Galaxy Zoo project [19]. In some other projects, instead, the users act as sensors, i.e., they perform some measurements on the object of interest: in the project "*Did you feel it?*"[4] and in the analogue Italian project "*Hai sentito il terremoto?*"[5], the users can provide information about their perception of an earthquake. They use VGI after the crisis has occurred, while we want to use it to prevent the occurrence of emergencies.

A second aspect of our proposal is to indicate a method for the quality assessment of VGI reports. The problem of validating VGI has been already tackled in literature. In [21] an approach for validating VGI is presented. The approach is based on the evaluation of the reputation of the source, on the consideration of the temporal and spatial distribution of the VGI items, and on the cross-referencing of different VGIs coming from different sources (e.g., Twitter messages and Flickr photos). In [14] the authors propose three approaches for validating VGI. In the crowd-sourcing approach the final users themselves validate the information. In the social approach some users are considered more trustworthy than others (e.g., since they provided several correct VGI in the past), and they have privileges (e.g., deleting pages or blocking user in Wikipedia) not granted to standard users. In the geographic approach, standard geographic rules that are known to be true are applied to identify VGI deviating from the rules. For example, if an area is known to be a natural reserve without business activities, a VGI indicating the presence of a restaurant in the area is considered unreliable. As far as we know, the proposal of validating VGI based on content and spatio-temporal analysis, performed by submitting queries to the database of VGI reports, is novel.

7. Conclusions and future work

Classical VGI managing systems usually provide basic filtering facilities to select subsets of VGI reports, e.g., those of a given category of users, category of

[4] http://earthquake.usgs.gov/earthquakes/dyfi/
[5] http://www.haisentitoilterremoto.it

risk, etc. As far as we know, none has still proposed a system for indexing and querying the contents of VGI reports to the aim of assessing their quality which is the main issue we have in mind with the current proposal. Specifically, the density-based queries have been defined in order to assess the trustfulness of VGI reports based on cross comparisons of the contents of the nearby geographic footprints of reports created in close dates. In this respect we propose the adoption of a flexible query language in order to allow the specification of tolerant selection conditions, thus yielding discriminated answers.

As future work, in the near future we plan to experiment the querying approach presented in this paper to existing Ushahidi installations of active projects, in order to have meaningful data that would permit us to draw some conclusions about the effectiveness of the proposed queries (in particular the density-based queries). In more advanced phases of the SISTEMATI project, instead, we will experiment our approach with the data provided by some partners of the project (local municipalities).

Acknowledgement

This work has been carried out within the project SISTEMATI "Geomatics Supporting Environmental, Technological and Infrastructural Disaster Management", funded by the Italian Ministry of Research jointly with Regione Lombardia.

References

[1] D. P. Anderson. Boinc: A system for public-resource computing and storage. In *5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004.

[2] G. Bordogna, L. Criscuolo, P. Carrara, and M. Pepe. An approach to assess the quality of volunteer geographic information for citizen science. *Information Sciences (submitted)*, 2012.

[3] G. Bordogna, M. Pagani, G. Pasi, and G. Psaila. Managing uncertainty in location-based queries. *Fuzzy Sets and Systems*, 160(15):2241–2252, 2009.

[4] G. Bordogna and G. Pasi. *Recent issues on fuzzy databases*, volume 53. Physica-Verlag HD, 2000.

[5] P. Bosc and J. Kacprzyk. *Fuzziness in database management systems.* Physica-Verlag, 1996.

[6] P. Bosc and H. Prade. An introduction to the fuzzy set and possibility theory-based treatment of flexible queries and uncertain or imprecise databases. In *Uncertainty management in information systems*, pages 285–324. Springer, 1997.

[7] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of the 23rd International Conference on Very Large Data Bases*, VLDB '97, pages 426–435, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[8] R. De Caluwe, G. De Tré, B. Van Der Cruyssen, F. Devos, and P. Maesfranckx. *Time management in fuzzy and uncertain object-oriented databases*, volume 39 of *Knowledge management in fuzzy databases*, pages 67–88. Physica-Verlag, 2000.

[9] J. Deoliveira. Geoserver: uniting the geoweb and spatial data infrastructures. In *Proc. of the 10th International Conference for Spatial Data Infrastructure, Trinidad*, 2008.

[10] R. Devillers, R. Jeansoulin, et al. *Fundamentals of spatial data quality.* ISTE London, 2006.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[12] J. Galindo, editor. *Handbook of Research on Fuzzy Information Processing in Databases.* IGI Global, 2008.

[13] M. F. Goodchild. Citizens as voluntary sensors: spatial data infrastructure in the world of web 2.0. *International Journal of Spatial Data Infrastructures Research*, 2:24–32, 2007.

[14] M. F. Goodchild and L. Li. Assuring the quality of volunteered geographic information. *Spatial Statistics*, 1:110–120, 2012.

[15] E. Hand. Citizen science: People power. *Nature*, 466(7307):685–687, Aug. 2010.

[16] E. Hatcher, O. Gospodnetic, and M. McCandless. Lucene in action, 2004.

[17] L. Litwin and M. Rossa. *Geoinformation Metadata in INSPIRE and SDI.* Lecture notes in geoinformation and cartography. Springer Berlin Heidelberg, 2011.

[18] O. Okolloh. Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. *Participatory learning and action*, 59(1):65–70, 2009.

[19] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg. Galaxy zoo: Exploring the motivations of citizen science volunteers. *Astronomy Education Review*, 9:010103, 2010.

[20] G. Salton. Automatic Text Processing: The Analysis, Transformation and Retrieval of Information by Computer, 1989.

[21] L. Spinsanti and F. O. Ostermann. Validation and relevance assessment of volunteered geographic information in the case of forest fires. In *ValGeo Workshop, Ispra*, 2010.

[22] J. Ticheler and J. U. Hielkema. Geonetwork opensource internationally standardized distributed spatial information management. *OSGeo Journal*, 2(1), 2007.