Modeling and Querying Uncertain Spatial Information for Situational Awareness Applications [*]

Dmitri V. Kalashnikov, Yiming Ma, Sharad Mehrotra, Ramaswamy Hariharan, Carter Butts

Information and Computer Science
University of California, Irvine, USA

ABSTRACT

Situational awareness (SA) applications monitor the real world and the entities therein to support tasks such as rapid decision-making, reasoning, and analysis. Raw input about unfolding events may arrive from variety of sources in the form of sensor data, video streams, human observations, and so on, from which events of interest are extracted. Location is one of the most important attributes of events, useful for a variety of SA tasks. In this paper, we propose an approach to model and represent (potentially uncertain) event locations described by human reporters in the form of free text. We analyze several types of spatial queries of interest in SA applications. Our experimental evaluation demonstrates the effectiveness of our approach.

Categories and Subject Descriptors

H.2.8 [**Information System**]: Database Management—*Spatial databases and GIS*

General Terms

Algorithms, Performance

Keywords

Modelling, Retrieval, Uncertain, Probability

1. INTRODUCTION

In this paper, we study the problem of representing and querying uncertain location information about real-world events that are described using free text. As a motivating example, consider the excerpts from two real reports filed by *Port Authority Police Department* (PAPD) Officers who participated in the events of September 11th, 2001:

1. “ *. . . the PAPD Mobile Command Post was located on West St. north of WTC and there was equipment being staged there . . .* ”

2. “ *. . . a PAPD Command Truck parked on the west side of Broadway St. and north of Vesey St. . . .* ”

These two reports refer to the same location of the same command post – a point-location in the New York, Manhattan area. However, neither the reports specify the exact location of the events, nor do they mention the same street names. We would like to represent such reports in a way that it enables efficient evaluation of spatial queries and analyses. For instance, the representation must enable us to retrieve events in a given geographical region (e.g., around World Trade Center). Likewise, it should enable us to determine similarity between reports based on their spatial properties; e.g., we should be able to determine that the above reports might refer to the same location.

Our primary motivation in studying the afore-mentioned problem comes from designing database solutions to support applications where the real world is being monitored (potentially using a variety of sensing technologies) to support tasks such as situation assessment and decision-making. Such Situational Awareness (SA) applications abound in a variety of domains including homeland security, emergency response, command and control, process monitoring/ automation, business activity monitoring, to name a few. Our particular interest lies in the domain of emergency response and security. We already alluded to the usefulness of spatial reasoning over free text in the example above. Such solutions are useful in a variety of other application scenarios in emergency response. For instance, such a system could support real-time triaging and filtering of relevant communications and reports among first responders (and the public) during a crisis. In our project, we are building SA tools to enable social scientists and disaster researchers to perform spatial analysis over two such datasets: (1) the transcribed communication logs and reports filed by the first responders after the 9/11 disaster, and (2) newspaper articles and blog reports covering the S.E. Asia Tsunami disaster. We believe that techniques such as ours can benefit a very broad class of applications where free text is used to describe events.

Our goal in this paper is to represent uncertain locations specified in reports to allow for effective execution of analytical queries. Clearly, merely storing location in the database as free text is not sufficient either to answer spatial queries or to disambiguate reports based on spatial locations. For example, spatial query such as ‘retrieve events near WTC’,

[*] This research was sponsored by NSF Award number 0331707 and 0331690 to the RESCUE project [23].

ACM-GIS’06, November 10–11, 2006, Arlington, Virginia, USA.
Copyright 2006 ACM 1-59593-529-0/06/0011 ...$5.00.

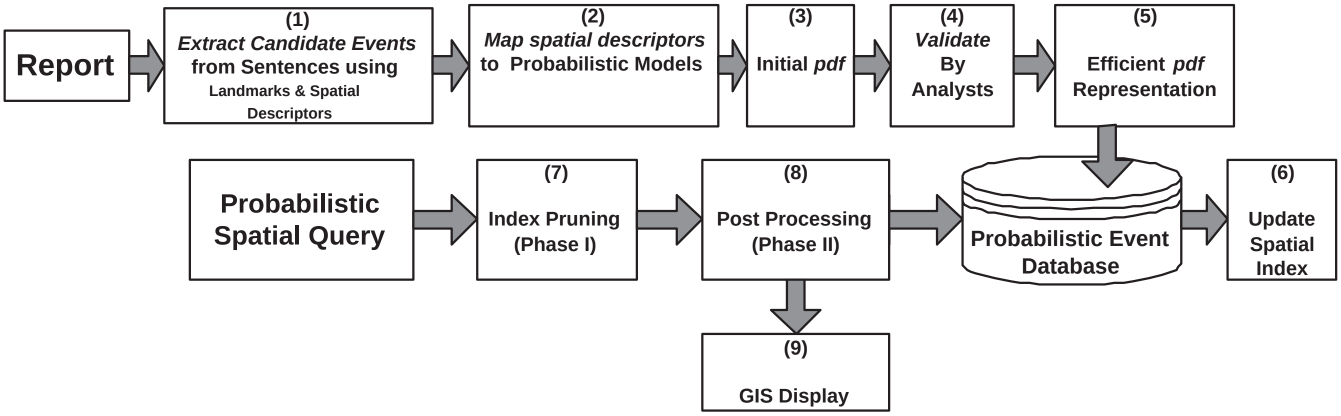


Figure 1: SAT Components

free text	*s-expression*
'near WTC'	`near`(WTC)
'on West St., north of WTC'	`on`(West St.) $\wedge$ `north`(WTC)

Figure 2: Examples of s-expressions.

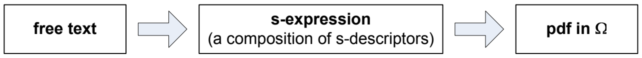


Figure 3: Free text location $\mapsto$ pdf

based on keywords alone, can only retrieve the first report mentioned earlier.

To support spatial analyses on free text reports, we need to project the spatial properties of the event described in the report onto the domain. In this paper, we model uncertain event locations as random variables that have certain probability density functions (*pdfs*) associated with them. We develop techniques to map free text onto the corresponding pdf defined over the domain.

Our approach is based on the assumption[1] that people report event locations based on certain *landmarks*. Let $\Omega \subset {}^2$ be a 2-dimensional physical space in which the events described in the reports are immersed. Landmarks correspond to significant spatial objects such as buildings, streets, intersections, regions, cities, areas, etc. embedded in the space. Spatial location of events specified in those reports can be mapped into *spatial expressions* (s-expressions) that are, in turn, composed of a set of *spatial descriptors* (s-descriptors) (such as `near`, `behind`, `infrontof`, etc) described relative to landmarks. Usually, the set of landmarks, the ontology of spatial descriptors, and the precise interpretations of both are domain and context dependent. Figure 2 shows excerpts of free text referring to event locations and the corresponding spatial expressions. These expressions use ***WTC*** and ***West St.*** as landmarks. While the locations of landmarks are precise, spatial expressions are inherently uncertain: they usually do not provide enough information to identify the exact point-locations of the events.

Our approach to representing uncertain locations described in free text consists of a two-step process, illustrated in Figure 3. First, a location specified as a free-text is mapped into the corresponding s-expression, which in turn is mapped to its corresponding pdf representation. Given such a model, we develop techniques to represent, store and index pdfs to support spatial analysis and efficient query execution over the pdf representations.

The **primary contribution** of this paper is an approach to mapping uncertain location information from free text into the corresponding pdfs in the domain Ω (Section 4).

The rest of the paper is organized as follows. In Section 2 we present an overall overview for our end-to-end SAT system for creating spatial awareness from text. Section 3 discusses related work. The main contribution of this paper, the approach for mapping textual locations into their probabilistic representations, is described in Section 4. We then discuss which aspects should guide the design of queries for SA applications and define some of the probabilistic queries that we use in our experiments. We empirically evaluate modeling aspects of our approach in Section 6. Finally, we conclude in Section 7.

2. SYSTEM OVERVIEW

Development of an end-to-end approach for spatial awareness from raw textual input must address several practical challenges. First, a mathematical *model* should be chosen or devised to allow to represent and manipulate with uncertain location information. Second, references to locations should be identified in raw reports, parsed and *extracted*. Third, a process should be devised for *mapping* the extracted textual locations into their representations in the chosen model. Next, a database *representation* should be designed, that would allow to efficiently store the uncertain locations in the database. The requirements of SA applications should be analyzed and the desired functionality should be reflected in choosing the set of spatial queries to be used by those applications. Queries are also might need to be adjusted to work with the given uncertainty model. Algorithms and auxiliary data structures will need to be designed for fast processing of the queries. Finally, SA applications would require convenient and intuitive interfaces for visualization and querying.

Figure 1 illustrates the major components of the prototype SAT system (Situational Awareness from Textual input) we have developed [19]. In this paper, we cover only

[1] We have validated this claim through a careful study of a variety of crisis related data sets we have collected in the past. This is also addressed in [15, 26].

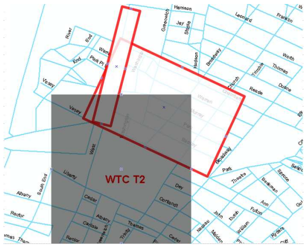


Figure 4: WTC: Data and Query

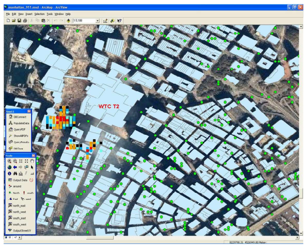


Figure 5: GIS Interface

one component, crucial to the whole process of creating spatial awareness. That component maps textual input into the probabilistic model and it is described in detail in Section 4. We next briefly overview the functionality of other SAT components.

Model. We use the spatial probabilistic model developed in [3, 4, 7], which will be discussed in more detail in the subsequent sections.

Extraction. There are many exciting research challenges that are related to extracting locations from text. This is a well studied problem that has received wide attention due to its importance in a variety of applications [1,29]. Nowadays, many of the state of the art extraction software packages, such as GATE and REX, already have a built-in functionality that allows for certain types of location extraction. As a rule, they provide the flexibility to further enhance this functionality, for instance by introducing new extraction rules and incorporating more domain knowledge, e.g. from Gazetteers. In the context of SAT, such tools need to be enhanced to also extract basic spatial descriptors, such as `near`, `on`, etc. Some of the extraction challenges arise due to ambiguous names of locations, e.g. a location specified only as 'Washington' can be used for both, WA state or 'Washington, D.C.', leading to ambiguity. Frequently context is used to help the disambiguation process or such issues are resolved using the analyst supervision.

Mapping. The process of mapping the extracted spatial locations into their corresponding probabilistic representations will be discussed in detail in Section 4. The analyst oversees this process to correct errors that arise from the mapping process and also to resolve extraction ambiguities. We integrate the mapping process as a toolkit to the standard GIS system as illustrated in Figure 5. For example, our extraction and modeling tools can automatically determine the uncertainty regions of the two reports in Section 1, and display them as in Figure 4.

Database Representation. In the context of SA applications, such as [19, 22], we need to be able to represent pdfs of complex and irregular shapes in the database. In this case, pdfs are continuous functions in 2-dimensional space, which need to be stored in the database. Various representations of pdfs has been explored in the past, including using histograms and representing pdfs as a mixture of other well-known distributions, e.g. Gaussians. Our solution is to first conceptually represent pdfs as histograms. Histograms, in turn, are represented as quad trees, with probability-summarization statistics attached to each node of the quad trees. These summarization statistics, precomputed in advance, allow for very effective query processing, since they help to avoid the corresponding costly integration operations during the runtime. We have also designed several lossy compression algorithms that allow to compress the quad-trees, resulting in overall storage improvement and query speedup due to reduced disk I/Os [18].

Probabilistic Spatial Query Component. SAT provides a query component to support common spatial query types. For example, using a spatial region query, an analyst can express a query such as "find all the events, the location of which are around WTC". Figure 4 visually illustrates this query (shaded region). The SAT system will compute the probability of the events to be inside this region, and filter away low probability events. In Section 5 we focus on the query requirement and semantics of different query types for SA applications.

Indexing and query processing. A system, that provides situational awareness on uncertain data, is of practical value only if it can demonstrate fast query processing capabilities and can scale to large domains. To achieve that, components 7 and 8 in Figure 1 handle indexing and efficient processing of spatial queries. The related techniques are covered in detail in [18].

3. RELATED WORK

In this section, we review relevant work on spatial modeling. We will use the probabilistic model for spatial uncertainty developed in [3–7], because it has the following properties:

- *Formality.* It builds on the formal probability theory.
- *Practicality.* It has been implemented in practice.
- *Generality.* This model is capable of handling (probabilistic versions of) many different types of spatial queries, as opposed to retrieval (selection) queries only.
- *Effectiveness.* Existing solutions that employ this model are known to be effective and scalable.

In the probabilistic model, an uncertain location ℓ is treated as a continuous random variable (r.v.) which takes values $(x, y) \in \Omega$ and has a certain probability density function (pdf) $f_\ell(x, y)$ associated with it. Interpreted this way, for any spatial region R, the probability that ℓ is inside R is computed as $\int_R f_\ell(x, y)dxdy$.

In general, spatial uncertainty has been explored both in the GIS and in database literature. The GIS literature has traditionally focused on qualitative approaches to representing uncertain spatial information [10, 11, 17, 25]. Spatial relations are classified as topological relations (e.g., disjoint, overlap), direction relations (e.g., North, South), ordinal relations (e.g., inside, contain), and distance relations (e.g., far, near). Geospatial ontologies have been explored in [2, 16]. Uncertain spatial information has been explored in the context of moving objects in [27]. In [24] authors proposed a probabilistic spatial data model that captures positional uncertainty arising due to imprecise data collection (e.g., such as GPS). In [28] the data model quantifies uncertainty arising from spatial analysis such as the discretization of thematic attributes.

Another related work is on georeferencing and spatial retrieval of documents, e.g. [29]. If we view the spatial domain as a uniform grid of cells, their modeling task can be formulated as follows. Given a document, for each cell in the grid, determine the number of times this cell is covered by the regions, mentioned in the document. Our modeling task is different: given a description of an event, for each cell,

Relation Class	**Descriptors**			
Topological	`indoor/inside`	`outdoor/disjoint`	`meet`	`at/on/equal`
Cardinal Direction	`north`	`east`	`south`	`west`
Orientation	`behind`	`in_front_of`	`to_the_left_of`	`to_the_right_of`
Distance	`within_dist`	`near`	`far`	`around`

Figure 6: Examples of S-descriptors.

Landmark Class	Name	Shape	Type	Area/footprint	Length	Height	City
Building	Y	Polygon	Housing/business/government	m^2	NA	story/meter	Y
Street	Y	Polyline	Highway/major/minor	NA	meter	NA	Y
Street Intersection	Y	Point	Highway/major/minor	NA	NA	NA	Y

Figure 7: Examples of Landmark Objects

determine the probability that the event happened in this cell.

Finally, there has been some theoretic work, e.g. [9], on modeling spatial uncertainty in text using heuristics and fuzzy logic techniques.

Mapping text into probabilistic model. While we employ an existing probabilistic model, the process of *mapping* textual locations into the corresponding representations in a probabilistic model has not been studied before. Such a mapping is one of the pivotal steps in developing the end-to-end awareness system, we cover it Section 4.

We have above summarized the existing body of research on spatial uncertainty most related to our paper. Other concepts/techniques (e.g., histograms, quad-trees, indexing), which are related to our work as well, will be discussed in this paper when the need arises.

4. MODELING LOCATION UNCERTAINTY

We model uncertain locations as continuous random variables that have certain pdfs associated with them. When processing a report about an event, our goal is to determine $f(x, y|report)$: the location of the event, given the information contained in that report. A report might contain several types of information that can influence $f(x, y|report)$. We focus on a frequent case where this density is context-invariant and in the form $f(x, y|s, t)$. Here s is an s-expression and t is the type of the event. We first consider how to compute $f(x, y|s)$. After that we will consider $f(x, y|s, t)$.

For instance, in the report '*A traffic accident near World Trade Center*', we have $s = \texttt{near}(WTC)$ and $t =$ '*traffic accident*'. Let us observe that, among all types of information mentioned in the report, s narrows down the possible location of the event most significantly. Then, we can employ the event type, 'traffic accident', to refine our answer further by observing that an event of that type is more likely to occur on a road than somewhere else.

Our approach first extracts s and t from the report (Section 4.1). S-expression s is a composition of s-descriptors $\mathcal{D}_1, \ldots, \mathcal{D}_n$. S-descriptors are less complex than s-expressions, and can be mapped into the corresponding pdfs (Section 4.2). The desired pdf $f(x, y|s, t)$ is computed by combining the pdfs $f(x, y|\mathcal{D}_i)$ and $f(x, y|t)$ (Section 4.3).

4.1 Mapping free text onto s-expression

Mapping of free-text locations into s-expressions has been studied before in the context of spatial ontologies. Even though spatial ontologies is *not* a focus of this paper, we summarize some of the related concepts to explain our approach.

The basic idea is that each application domain $\mathcal{A}$ has, in general, its own spatial ontology $\mathcal{D}(\mathcal{A})$. The ontology defines what constitutes the landmarks in $\mathcal{A}$. It also defines the set of basic s-descriptors $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ and ways to compose them, such that any free-text location from $\mathcal{A}$ can be mapped onto a composition of s-descriptors. The four major classes of s-descriptors are topological relations (e.g., disjoint, inside) [10], cardinal direction relations (e.g., north, west) [12], orientation relations (e.g., left of, right of) [14], and distance relations (e.g., near, around) [13]. Examples of landmarks and s-descriptors are provided in Figures 6 and 7. Each s-descriptor is of the form $\mathcal{D}_i(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m)$: it takes as input $m \in$ landmarks, where m is determined by the type of s-descriptor. Figure 2 shows examples of free text referring to event locations and the corresponding s-expressions. Some descriptors may not take any parameters, e.g. an ontology may use the concepts of `indoor` and `outdoor`, to mean 'in some building' and 'not in any building'.

One can define multiple types of s-expressions, such as `AND`-, `OR`-, `NOT`-expressions. However, there are only two common scenarios in practice: (1) s-expression consists of a single instantiated s-descriptor and (2) s-expression is an `AND`-expression. An `AND`-expression arises when the same location ℓ is described using n different descriptions $s_1, s_2, \ldots, s_n$, which we denote as:

$$s = s_1 \wedge s_2 \wedge \cdots \wedge s_n.$$

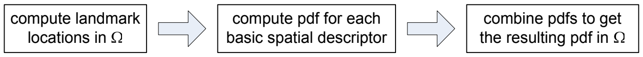
As an example, assume a person is asked 'where are you?' to which he replies 'I am near building A *and* near building B', which corresponds to the s-expression $\texttt{near}(A) \wedge \texttt{near}(B)$.

Let us note that representing event location using s-expression requires first extracting them from text. Although extracting spatial properties is complex in general, when ontologies and domains are fixed, the task becomes relatively simpler. The analyst manually supervises the extraction process to correct exceptional situations and errors.

4.2 Pdf for a single s-descriptor

Merely having locations represented as spatial expressions is still not sufficient. We also need to be able to *project* the meaning of each s-expression onto the domain Ω. We achieve this by (a) computing the projection (i.e., the pdf) of each individual s-descriptor in the s-expression; and (b) combining the projections, as illustrated in Figure 8.

Let us first understand how a basic s-descriptor can be

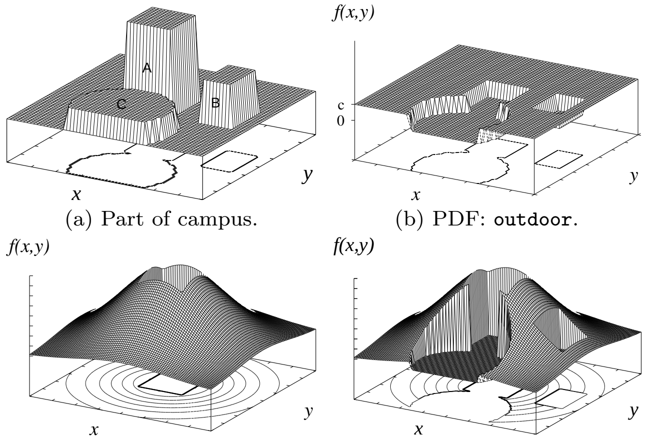


compute landmark locations in Ω

compute pdf for each basic spatial descriptor

combine pdfs to get the resulting pdf in Ω

Figure 8: Combination of s-descriptors ↦ pdf.



(a) Part of campus.

(b) PDF: `outdoor`.

(c) PDF: $\texttt{near}(A)$.

(d) PDF: $\texttt{outdoor} \wedge \texttt{near}(A)$.

Figure 9: Part of campus and various pdfs.

projected into Ω in an automated fashion, i.e. not manually. In Section 4.3, we will demonstrate how to compose those projections to determine the pdfs for s-expressions. Let us note that other components of the overall approach for creating spatial awareness from text are *independent* from a particular algorithm for mapping basic s-descriptors into pdfs. This section presents only one such algorithm, which can be treated as a guideline for creating customized density functions suited for a particular domain.

To illustrate the steps of the algorithm more vividly, consider a simple scenario demonstrated in Figure 9(a). This figure shows a portion of a university campus with three buildings A, B, and C. We will illustrate the concepts with the help of two descriptors: `outdoor` and $\texttt{near}(A)$. Notice that, in general, at run time the algorithm might need to compute the pdf for $\texttt{near}(\mathcal{L})$ for any landmark $\mathcal{L}$. If it is desirable to automate this pdf computation, intuitively, one should avoid manually predefining a separate pdf per each known landmark $\mathcal{L}$ in the domain in advance, since there can be many landmarks. Instead, a more preferable approach is to design a single generic pdf-generating procedure for all possible landmarks. Given any landmark $\mathcal{L}$, such a procedure will generate the desired pdf based only on the relevant *properties* of the landmark, such as its footprint and height.

That is, one method for determining the pdf $f(x, y|\mathcal{D})$ for any s-descriptor $\mathcal{D}(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m)$ is to make reasonable assumptions about the functional form of $f(x, y|\mathcal{D})$ based on the properties of the landmarks it takes as input. Those assumptions can be refined or rejected later on, e.g. using Bayesian framework [8].

For instance, we can define the pdf $f(x, y|\texttt{outdoor})$ for the s-descriptor `outdoor` as having the uniform distribution everywhere inside the domain Ω except for the footprints of the buildings that belong to Ω, as illustrated in Figure 9(b). That is $f(x, y|\texttt{outdoor}) = c$ for any point $(x, y) \in \Omega$ except when (x, y) is inside the footprint of a building, in which case $f(x, y|\texttt{outdoor}) = 0$. The real-valued constant c is determined from the constraint $\int_\Omega f(x, y|\texttt{outdoor})dxdy = 1$.

Another example of an s-descriptor is $\texttt{near}(A)$, which means somewhere close to the landmark A (the closer the better), but not inside A. Let us observe that, unlike the density for `outdoor`, the pdf for $\texttt{near}(A)$ is clearly *not uniform*. Rather, a more reasonable density can be a variation of the truncated-Gaussian density, centered at the center of the landmark, with variance determined by the spatial properties of the landmark A (its height, the size of its footprint). Also, since the location cannot be inside A, the values of that density should be zero for each point inside the footprint of the landmark, as illustrated in Figure 9(c).

Using the above procedure, we can construct pdfs for arbitrarily complex s-descriptors in an automated fashion. Naturally, there can be exceptions from the rule and for a limited number of landmarks the above described general procedure might not produce good pdfs. The analyst should handle those cases beforehand, simply by manually assigning pdfs for the corresponding s-descriptors in advance. In general, the analyst can supervise the automated pdf construction process as well.

4.3 The pdf of a spatial expression

Now that we know how to map s-descriptors into the corresponding pdfs, let us consider how to compute the pdf for an `AND-` s-expression. Let us first assume an `AND`-expression s consists of only two sub-expressions: $s = s_1 \wedge s_2$, as in $\texttt{near}(A) \wedge \texttt{near}(B)$. Our goal is to derive $f(x, y|s_1, s_2)$ from already known $f(x, y|s_1)$, $f(x, y|s_2)$, and $f(x, y)$. Here, $f(x, y|s_1, s_2)$ is the pdf of the event location, given the report contains s_1 and s_2. The density $f(x, y)$ is the *global prior* which tells us where an event is likely to occur in the absence of any knowledge about the event. To derive $f(x, y|s_1, s_2)$ we first apply Bayes formula:

$$f(x, y|s_1, s_2) = \frac{\mathrm{P}(s_1, s_2|x, y) f(x, y)}{\mathrm{P}(s_1, s_2)},$$

where $\mathrm{P}(s_1, s_2|x, y)$ is the probability to observe s_1 and s_2 in a report, given the location is (x, y). While the presence of s_1 in a report is clearly not independent from the presence of s_2, it is reasonable to assume that they are *conditionally independent given the location.* In other words, it holds that:

$$\mathrm{P}(s_1, s_2|x, y) = \mathrm{P}(s_1|x, y)\,\mathrm{P}(s_2|x, y),$$

but

$$\mathrm{P}(s_1, s_2) \neq \mathrm{P}(s_1)\,\mathrm{P}(s_2).$$

For example, if buildings A and B are very close to each other, things that are 'near A' will also tend to be 'near B' and thus the two are dependent. However, once we know the location (x, y), we do not need to know whether this location is 'near A' to decide whether it is 'near B' and vice versa. Using the assumption of conditional independence, we have:

$$f(x, y|s_1, s_2) = \frac{\mathrm{P}(s_1|x, y)\,\mathrm{P}(s_2|x, y) f(x, y)}{\mathrm{P}(s_1, s_2)}.$$

By applying Bayes formula we compute:

$$\mathrm{P}(s_1|x, y) = \frac{f(x, y|s_1)\,\mathrm{P}(s_1)}{f(x, y)},$$

$$\mathrm{P}(s_2|x, y) = \frac{f(x, y|s_2)\,\mathrm{P}(s_2)}{f(x, y)}.$$

Thus

$$f(x,y|s_1,s_2) = \frac{f(x,y|s_1)f(x,y|s_2)}{f(x,y)} \cdot \frac{\mathrm{P}(s_1)\,\mathrm{P}(s_2)}{\mathrm{P}(s_1,s_2)}. \qquad (1)$$

We can assume that the global prior $f(x,y)$ is uniform, or make a weaker assumption that it is locally uniform, that is, it is not uniform in general but look uniform inside smaller regions in Ω. Let us observe that if U_1 is an uncertainty region for $f(x,y|s_1)$ and U_2 for $f(x,y|s_2)$, then an uncertainty region for $f(x,y|s_1,s_2)$ can be computed as:

$$U_{1\wedge 2} = U_1 \cap U_2.$$

For the global prior that is uniform, or locally uniform in $U_{1\wedge 2}$, Eq. (1) can be written as:

$$f(x,y|s_1,s_2) = f(x,y|s_1)f(x,y|s_2) \cdot c,$$

or, equivalently, as:

$$f(x,y|s_1,s_2) \propto f(x,y|s_1)f(x,y|s_2).$$

Here, c is a real-valued constant that depends on s_1 and s_2 but does not depend on x and y. To compute c, observe that by definition of an uncertainty region, the true event location is somewhere inside $U_{1\wedge 2}$. Consequently, $f(x,y|s_1,s_2)$ integrates to 1 over $U_{1\wedge 2}$ and thus the value of c is:

$$c = \frac{1}{\int_{U_{1\wedge 2}} f(x,y|s_1)f(x,y|s_2)dxdy}.$$

Let us note that if the integral in the denominator integrates to zero, this constant is undefined. The latter corresponds to an inconsistent definition of a location, such as in 'near L.A., California and London, England'. Thus

$$\begin{cases} f(x,y|s_1,s_2) = f(x,y|s_1)f(x,y|s_2) \cdot \frac{1}{I} & \text{if } I \neq 0; \\ f(x,y|s_1,s_2) \text{ is undefined} & \text{if } I = 0, \end{cases} \qquad (2)$$

$$\text{where } I = \int_{U_{1\wedge 2}} f(x,y|s_1)f(x,y|s_2)dxdy.$$

Similarly, for a general s-expression $s = s_1 \wedge \cdots \wedge s_n$ it holds that:

$$f(x,y|s_1,\ldots,s_n) \propto f(x,y|s_1) \times \cdots \times f(x,y|s_n).$$

Incorporating event type. Let us observe that the event type t can be viewed as another `AND` condition in $f(x,y|s,t)$ and we can apply the above deduction to derive that:

$$f(x,y|s,t) \propto f(x,y|s)f(x,y|t).$$

If the event type does not provide any new information where the event could occur, then we assume $f(x,y|t)$ is (locally) uniform. However, often $f(x,y|t)$ is not uniform and can help us to reduce the uncertainty further. For example, we know that t = *'home robbery'* implies an event happened at a home and not in the middle of a street. It is interesting to observe that $f(x,y|t)$, in essence, serves as a *local prior*: it tells us where an event of a *given type* is likely to occur in the absence of other knowledge.

The above formulas allow us to derive the exact density $f(x,y|s,t)$ for the types of s-expressions studied in this section, so that we can answer all types of probabilistic spatial queries. As an example of a pdf that results from an `AND`-expression, consider a possible pdf for `outdoor` $\wedge$ `near`(A) shown in Figure 9(d), in the context of the university campus scenario from Figure 9(a). Finally, observe that in SA domains pdfs can be complex and can have highly irregular forms, which do not lend themselves to simple Gaussian or uniform approximation. Thus, special methods for representing and storing pdfs should be devised [18]. The representation of data is normally determined by the nature of queries that are executed on top of the data. In the next section we take up the types of spatial queries that need to be supported by SA applications.

5. SPATIAL QUERIES

SA applications for crisis response, in general, should provide support for all standard types of spatial queries, such as range, NN, spatial join, and so on. Such applications, however, have several salient features that must be taken into account in the context of query design and processing. They are (1) uncertainty in data; (2) the need for fast query response; and (3) the need for triaging capabilities. In crisis situations, *triaging capabilities* can play a decisive role in reducing the amount of information the analyst should process. Those capabilities operate by restricting the size of query result sets and filtering out, or, *triaging*, only the most important results, possibly in a ranked order. The rest of the section lists the definitions of various region-based queries. For more detailed examples of using those queries the reader is referred to [18], which is an attendant publications also published in ACM GIS 2006.[2]

A fundamental type of query that should be supported by SA applications is a *range* or *region* query, such as "find all the events, the location of which can be inside a given region". Those queries can be formally defined as:

Definition 1 *Given a region (range) R and a set of objects $A = \{a_1, a_2, \ldots, a_n\}$, a basic* **region (range) query (RQ)** *returns all the elements in A whose probability of being inside R is greater than zero.*

The analytical formula for computing the probability that a location $\ell \sim f_\ell(x,y)$ is located inside a region R is:

$$\mathrm{P}(\ell \in R) = \int_R f_\ell(x,y)dxdy. \qquad (3)$$

Definition 2 *A probabilistic query is a* **detached-probability** *query if it returns elements without the probabilities associated with them. A query is an* **attached-probability** *query, denoted as p-, if its result is a set of tuples where each tuple consists of an element and the probability associated with this element.*

By default, any spatial query is a detached-probability query.

Definition 3 *Given a threshold p_τ, query τ-Q is said to be query Q* **with the threshold semantics** *if on the same input as Q it returns all the elements from the result set of Q whose associated probabilities are greater than p_τ.*

Let us now introduce another concept that SA application should support to enable triaging capabilities. In certain situations, it can be hard for the analyst to determine the

[2] Similar queries have also been studied in [3, 4].

right value of the threshold p_τ. As an alternative, it can be desirable to draw conclusions based on only the top-k ranked results. To support this functionality, we define:

Definition 4 *Given $k \in$, query k-Q is said to be query Q* **with the top-k semantics** *if on the same input as Q it returns $\min(k, |S|)$ elements from the result set S of Q whose associated probabilities are the highest in S.*

One can combine various query semantics to specify queries with the desired functionality. For example, there can be $p\tau$-RQ or $pk\tau$-RQ queries.

6. EXPERIMENTAL EVALUATION

In this section, we experimentally study the effectiveness of the proposed approach. We ran all our experiments on a P4-2GHz PC with 1GB RAM.

6.1 Experimental Setup

We use a real geographic dataset for the New York, Manhattan area, including area around World Trade Center. The original dataset is in vector format. It contains information about the buildings (polygons), streets/roads (line segments), and street intersections (points). A 400 × 400 virtual grid is overlaid on top of this dataset, with 10 × 10 m^2 cells, covering 4 × 4 km^2.

Uncertain location data for testing the accuracy is derived based on the 164 reports filed by NYPD Officers. From these reports, we use the probabilistic event modeling process introduced in Section 4 to construct the pdfs attached to the 2359 events. Our supervised location extraction tool has a knowledge base of all the street and building names in the area. It extracts the landmarks (buildings, street intersections, and streets in our case) automatically, and zooms into the corresponding text and highlights the s-descriptors. Our tool also provides an initial suggestion of combining related s-descriptors at sentence level to form an s-expression. The analyst can either accept or reject the suggestion; the latter offers an opportunity to the analyst to combine them differently. Once an s-expression is determined, our supervised modeling tool models the involved s-descriptors separately and generates the overall models for the s-expression based on the mechanisms introduced in Section 4. Under the analyst's supervision, the extraction tool can extract all the s-expressions from the text. However, even with human help, certain level of modeling error can still occur during the process of converting s-expressions to pdf. In our experiments, we show that even with certain level of modeling errors, our modeling process still outperforms the naive solutions significantly.

We then manually select 50 RQ queries based on the point of interests (POIs) in the area. The size of the queries varies from 10 × 10 to 100 × 100 vcells.

6.2 Experiments

The proposed modeling approach has two sources that can introduce errors, and thus the accuracy of the approach should be quantified. Firstly, there can be standard errors associated with extracting spatial location from text. We focus on the second type of error, most related to this paper, that arises due to creating the model $f(x, y|\mathcal{D})$ for various instantiated s-descriptors $\mathcal{D}$.

In Section 4 we have described a process for deriving $f(x, y|\mathcal{D})$ for an s-descriptor $\mathcal{D}$. That process employs properties of landmarks and generates pdf $f_{est}(x, y|\mathcal{D})$ as its *estimation* of the *true* desired pdf $f(x, y|\mathcal{D})$. Even though this process is supervised by the analyst, who might pick good generic models, in the end the parameters for the model (derived from landmark properties) will not be 100% accurate, leading to discrepancies between $f_{est}(x, y|\mathcal{D})$ and $f(x, y|\mathcal{D})$.

Naturally, it is difficult to assess this discrepancy exactly, because the true $f(x, y|\mathcal{D})$ is unknown to us. However, it is possible to analyze the effect of errors in parameters. For that, we first generate the pdf $f(x, y|\mathcal{D})$ according to some realistic distribution and assume it is the true pdf. Then, we choose $f_{est}(x, y|\mathcal{D})$ as the distribution for $f(x, y|\mathcal{D})$ but with manually disturbed parameters. Then we measure the precision/recall quality of RQs and PRQs where instead of $f(x, y|\mathcal{D})$ we use $f_{est}(x, y|\mathcal{D})$.

We will also compare our approach against a baseline method, where the analyst represents each uncertain location ℓ by simply drawing its uncertainty region U_ℓ, without specifying its pdf $f_\ell(x, y)$. Using that representation, the answer to a RQ with the range R and probabilistic threshold p_τ can be computed by measuring the fraction of the overlap between U_ℓ and R and comparing it to p_τ. The comparison to the baseline will show whether there is actually a merit for keeping $f_\ell(x, y)$ for ℓ, or whether U_ℓ is sufficient.

We will focus on two types of modeling errors that can arise during the construction of the model $f_\ell(x, y|\mathcal{D})$. In the first case, the analyst is overly confident in determining the event location and the resulting $f_{est}(x, y|\mathcal{D})$ is too "tight". Similarly, in the second case the analyst is not confident enough in locating the event and the resulting $f_{est}(x, y|\mathcal{D})$ is too loose. We simulate the first type of error by reducing the variance σ used in $f_\ell(x, y|\mathcal{D})$ by certain percentage (-80%, -50%, -20%). Similarly, for the second case we increase the σ by certain percentage (20%, 50%, 100%). Figure 10 show the comparisons of the modeling accuracy using the well-known F1 measure (the harmonic mean of the precision and recall). To compute the F1 measure, we first run the queries on true model $f_\ell(x, y)$ at different threshold levels (i.e., 20%, 50% and 80%). That gives us the ground truth, which we use in computing $F1$ for estimated models $f_{est}(x, y)$. Figure 10 shows that when the error is small, e.g. 20%, the accuracy of the estimated model approaches the optimal F1 measure – equal to 1 – at all threshold levels. We also can observe that the baseline method (U_ℓ) performs poorly, especially when the threshold levels are high (e.g., 80%), which demonstrates that U_ℓ is inadequate during the situations where high threshold levels are needed for prioritizing and triaging events.

7. CONCLUSION

In this paper, we presented our approach for building spatial awareness from textual input. We focus on the practical aspects of modeling and query design. We also demonstrated the effectiveness of our solution. This paper also opens up a variety of interesting follow up research problems on better automated extraction and modeling techniques. We plan to address those problems as our future work.

8. REFERENCES

[1] Gate – general architecture for text engineering. In *NLP, University of Sheffield*, 2006.

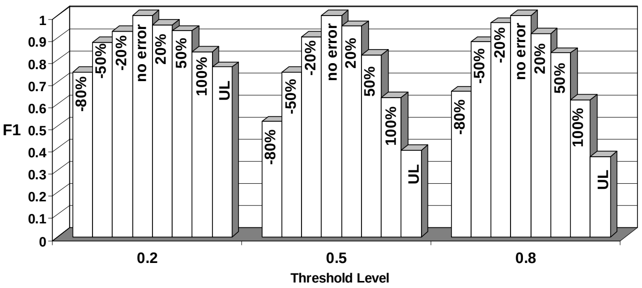


Figure 10: The effect of errors.

[2] I. Arpinar, A. Sheth, and C. Ramakrishnan. *Handbook of Geographic Information Science*. Blackwell Publsh., 2004.

[3] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proc. of ACM SIGMOD International Conference on Management of Data (ACM SIGMOD 2003)*, San Diego, CA, USA, June 9–12 2003.

[4] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 16(9), Sept. 2004.

[5] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluation of probabilistic queries over imprecise data in constantly-evolving environments. *Information Systems Journal*, 2006. to appear.

[6] R. Cheng, S. Prabhakar, and D. V. Kalashnikov. Querying imprecise data in moving object environments. In *Proc. of the 19th IEEE International Conference on Data Engineering (IEEE ICDE 2003)*, Bangalore, India, March 5–8 2003.

[7] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. of VLDB*, 2004.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[9] S. Dutta. Approximate spatial reasoning. *IEA/AIE*, 1, 88.

[10] M. Egenhofer and J. Herring. A mathematical framework for the definitions of topological relationships. In *Int'l Symp. on Spatial Data Handling*, 1990.

[11] A. Frank. Ontology for spatio-temporal databases. In *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, 2003.

[12] A. U. Frank. Qualitative spatial reasoning with cardinal directions. In *In Proceedings of the Austrian Conference on Artificial Intelligence*, 1991.

[13] A. U. Frank. Qualitative spatial reasoning about distance and directions in geographic space. In *Journal of Visual Languages and Computing*, 1992.

[14] C. Freksa. Using orientation information for qualitative spatial reasoning. In *In Proceeding of International Conference on The-ories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 1992.

[15] R. Golledge. *Wayfinding behaviour*. The Johns Hopkins University Press, 1999.

[16] K. Hiramatsu and F. Reitsma. Georeferencing the semantic web: ontology based markup of geographically referenced information. In *EuroSDR/EuroGeographics*, 2004.

[17] W. Kainz, M. Egenhofer, and I. Greasley. Modeling spatial relations and operations with partially ordered sets. In *Int'l J. of GISs, 7(3):215-229*, 1993.

[18] D. V. Kalashnikov, Y. Ma, S. Mehrotra, and R. Hariharan. Index for fast retrieval of uncertain spatial point data. In *Proc. of Int'l Symposium on Advances in Geographic Information Systems (ACM GIS 2006)*, Arlington, Va, USA, November 10–11 2006.

[19] D. V. Kalashnikov, Y. Ma, S. Mehrotra, R. Hariharan, N. Venkatasubramanian, and N. Ashish. SAT: Spatial Awareness from Textual input. In *Proc. of International Conference on Extending Database Technology (EDBT 2006)*, Munich, Germany, March 26–30 2006.

[20] D. V. Kalashnikov, S. Prabhakar, and S. Hambrusch. Main memory evaluation of monitoring queries over moving objects. *Distributed and Parallel Databases, An International Journal (DAPD)*, 15(2):117–135, Mar. 2004.

[21] D. V. Kalashnikov, S. Prabhakar, S. Hambrusch, and W. Aref. Efficient evaluation of continuous range queries on moving objects. In *Proc. of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002)*, Aix en Provence, France, September 2–6 2002.

[22] S. Mehrotra, C. Butts, D. V. Kalashnikov, N. Venkatasubramanian, K. Altintas, H. Lee, A. Meyers, J. Wickramasuriya, R. Hariharan, Y.Ma, R. Eguchi, and C. Huyck. CAMAS: A citizen awareness system for crisis mitigation. In *Proc. of ACM SIGMOD International Conference on Management of Data (ACM SIGMOD 2004), demo paper*, Paris, France, June 13–18 2004.

[23] S. Mehrotra, C. Butts, D. V. Kalashnikov, N. Venkatasubramanian, R. Rao, G. Chockalingam, R. Eguchi, B. Adams, and C. Huyck. Project RESCUE: challenges in responding to the unexpected. In *Proc of SPIE*, volume 5304, pages 179–192, Jan. 2004.

[24] J. Ni, C. Ravishankar, and B. Bhanu. Probabilistic spatial database operations. In *SSTD*, 2003.

[25] D. Papadias and T. Sellis. Qualitative representation of spatial knowledge in two-dimensional space. In *VLDB J*, 94.

[26] M. Sorrows and S. Hirtle. The nature of landmarks for real and electronic spaces. *Spatial Information Theory*, 1661, 99.

[27] G. Trajcevski and O. Wolfson. Managing uncertainty in moving objects databases. In *ACM TODS, 29(3)*, 2004.

[28] T. Windholz, K. Beard, and M. Goodchild. Data quality: A model for resolvable objects. In *Advances in Spatial Data Quality, Taylor-Francis*, 2001.

[29] A. Woodruff and C. Plaunt. *GIPSY: Georeferenced Information Processing SYstem*. 1994.