Identifying and Describing Streets of Interest

Dimitrios Skoutas
IMIS, Athena R.C., Greece
dskoutas@imis.athena-innovation.gr

Dimitris Sacharidis
TU Wien, Austria
dimitris@ec.tuwien.ac.at

Kostas Stamatoukos
IMIS, Athena R.C., Greece
kstamatoukos@imis.athena-innovation.gr

ABSTRACT

The amount of crowdsourced geospatial content on the Web is constantly increasing, providing a wealth of information for a variety of location-based services and applications. This content can be analyzed to discover interesting locations in large urban environments which people choose for different purposes, such as for entertainment, shopping, business or culture. In this paper, we focus on the problem of identifying and describing Streets of Interest. Given the road network in a specified area, and a collection of geolocated Points of Interest and photos in this area, our goal is to identify the most interesting streets for a specified category or keyword set, and to allow their visual exploration by selecting a small and spatio-textually diverse set of relevant photos. We formally define the problem and we present efficient algorithms, based on spatio-textual indices and filter and refinement strategies. The proposed methods are evaluated experimentally regarding their effectiveness and efficiency, using three real-world datasets containing road networks, POIs and photos collected from several Web sources.

1. INTRODUCTION

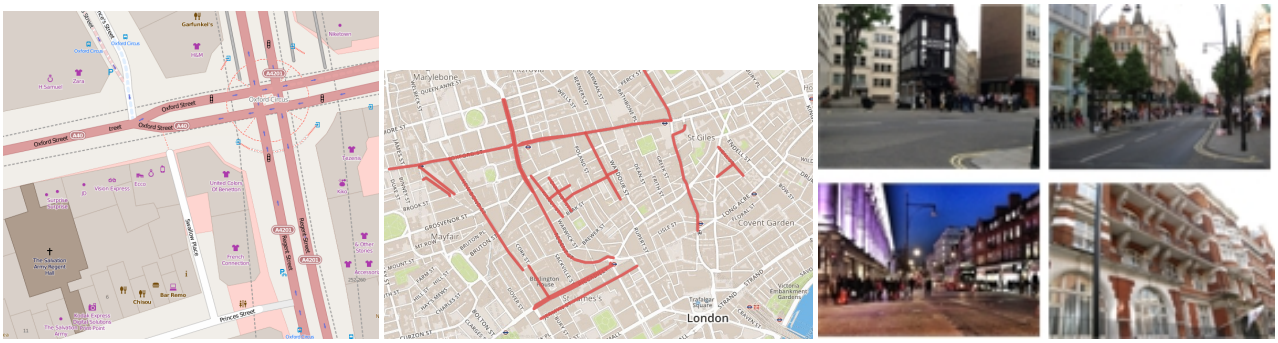
A large amount of user-generated content is becoming available on the Web daily, with increasingly large portions of it being associated with geospatial information. Typical examples include maps of road networks and other spatial features available on OpenStreetMap and Wikimapia, information about *Points of Interest* (POIs) from Wikipedia and Foursquare, geotagged photos from Flickr and Panoramio. This creates a valuable resource for discovering and exploring locations and areas of interest, with numerous applications in location-based services, geomarketing, trip planning, and other domains. In this paper, we advocate the use of *street* as the elemental area of interest in modern cities, and tackle two complementary problems, *identifying* and *describing* them.

Regarding the first task, there has been substantial work in identifying a *single POI*, based on spatial and textual criteria. More precisely, spatio-textual similarity queries, aim at retrieving POIs that are both spatially close to a given location and textually relevant to a given set of keywords specifying an information need [9]. Additional metadata associated to the POIs, such as ratings, comments, "likes" or check-ins, can be considered to weigh the *importance* of each POI when computing the ranking. The proximity of a POI to other relevant POIs has also been considered as a factor indicating importance [5]. Furthermore, in Location-Based Social Networks, information about social connectivity is also considered in determining the importance of POIs (and users) [4, 2].

A line of work more related to our first task deals with discovering a set of nearby and topically related POIs, that designate a *Region of Interest*. This is a more challenging problem, and proposals mainly differ in the way regions are formed. The majority of past research addresses the *maximum range sum problem*, where the region is defined as either a rectangle of fixed length and width [21, 24, 10], or a disk with fixed radius [10], and the objective is to find the one that maximizes an aggregate importance score on the topically relevant POIs it contains. Other works do not enforce a constraint on region shape or size, and rather implement density-based clustering of POIs [20, 19].

All aforementioned works assume that POIs are located in a Euclidean space. However, particularly in urban environments, it is often more realistic and useful to consider the underlying road network. Grouping together nearby POIs makes little sense if the actual travel distance between them is large, e.g., when they are located in opposite banks of a river. Surprisingly, little work is done in this frontier. In [7], the authors look for a connected subgraph of the road network that maximizes an aggregate score on the relevant POIs that are included, subject to a constraint on its total length.

Nonetheless, such an approach also has shortcomings. First, the fact that there is no control on the *subgraph type* may result in returning oddly-shaped regions that are hard to inspect and not particularly meaningful in a user exploration setting. Additionally, the approach favors *POI quantity* over density. More often than not, there exists a single popular street with a high density of POIs. Using the formulation of [7], such a street would be in the result but accompanied by several other smaller adjacent streets that happen to have at least one relevant POI. Similarly, looking for *connected components* may lead to discovering artificial links among important streets for the sole purpose of ensuring connectivity. Another limitation is that [7] assumes POIs are conveniently situated on the road network as vertices. In reality, however, the situation is much different. Figure 1(a) shows the map of a popular corner (Oxford Str. and Regent Str.) in the center of London, and also depicts various types of POIs. It should be apparent that there is no straightforward mapping of POIs to the road network vertices. Instead, it is more natural to "assign" POIs to edges (streets) but not necessarily in an exclusive manner. For example a POI (e.g., the clothing shop in Figure 1(a)) near the corner should be associated with both intersecting streets. Moreover, a POI (e.g., the photo shop in Figure 1(a)) farther from the streets but inside a corner building should also contribute to the importance of the main crossing streets.



(a) Street segments and POIs (b) Top-20 Streets of Interest (c) Photo description of Oxford Street

Figure 1: Illustrative example for shopping streets in the center of London.

Motivated by these observations, we formulate the problem of *identifying Streets of Interest* (SOIs). Briefly, given some textual information (keywords, categories) the goal is to identify the streets (more accurately the street segments) that have a large density of relevant POIs around them. An example for the center of London is illustrated in Figure 1(b), where the top 20-SOIs are highlighted with red. Notice that the returned streets are not connected via non-interesting streets. Moreover, our ranking approach naturally allows for an exploratory search of the area. To efficiently retrieve a ranked list of SOIs, we propose an algorithm inspired from top-k query processing that operates on top of spatio-textual indices.

Although very helpful, identifying the main SOIs for the user's keywords is only the first step towards exploring a larger area. Typically, the user then needs to find more information and gain more insights about those results that are discovered and suggested. For this purpose, perhaps the easiest and most effective way is by providing visual information; thus, a valuable source is the numerous relevant photos that can be found in various Web sources, such as Flickr and Panoramio. The challenge that arises then is how to select a small set of results to present, in order to avoid overloading the user, especially when using a mobile device with limited resources in terms of bandwidth, screen size and battery, while still providing enough information. This can be achieved by *diversifying* the results to present, so that more and different information can be conveyed with fewer results.

To that end, the second task we address in this paper refers to the selection of a concise and spatio-textually diverse set of relevant photos for describing the discovered SOIs. We follow the general diversity principles from information retrieval [16, 8], and formulate a *spatio-textual SOI diversification* problem. In particular, we introduce spatial and textual measures of *relevance* and *diversity*, and seek to extract a small set of photos that act as an informative summary of a given SOI (see the example in Figure 1(c) for a 4-photo summary of Oxford Str). As this is a computationally hard problem, we turn into heuristic methods supported by appropriate spatio-textual indices.

The main contributions of our work can be summarized as follows:

- We formally define the top-k SOI query, and we present an efficient algorithm for its evaluation.
- We present spatio-textual relevance and diversity criteria for selecting subsets of available photos to describe SOIs, and we propose an efficient approximation algorithm for their computation.
- We present the results of an experimental evaluation of our proposed methods, using real-world datasets containing road networks, POIs and photos from several major Web sources, covering the areas of three different European capital cities.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 formally introduces the problem of finding k-SOIs, and presents an efficient algorithm for their computation. Section 4 presents measures for spatio-textual relevance and diversity, and an efficient approximation algorithm for selecting diversified subsets of relevant photos to describe SOIs. Finally, Section 5 presents the results of our experimental evaluation, while Section 6 concludes the paper.

2. RELATED WORK

This section reviews existing approaches for spatio-textual POI retrieval and diversification.

2.1 Ranking Points and Areas of Interest

Numerous works have focused on discovering and ranking points or areas of interest, based on various definitions and criteria. The main differences involve the following aspects: (a) whether the focus is on single POIs or whole areas, i.e. sets of POIs; (b) whether the problem involves nearby search around a given query location or rather browsing and exploration within a whole area; (c) whether the aim is to maximize the number or the total score (e.g. relevance or importance) of the POIs enclosed in the discovered area or to minimize some cost function (e.g. distance or travel time) on a set of POIs that suffice for covering the query keywords.

The majority of existing works focuses on the ranking of single POIs. Location-aware top-k text retrieval queries have been studied in [11]. Given the user location and a set of keywords, this query returns the top-k POIs ranked according to both their spatial proximity and their textual relevance to the query. For the efficient evaluation of such queries, a hybrid indexing approach was proposed, integrating the inverted file for text retrieval and the R-tree for spatial proximity querying. Further variations on spatio-textual queries and indexes have been extensively studied [9]. Top-k spatial keyword queries have been studied also in [23], with distances being calculated on the road network instead of the Euclidean space. A different perspective for ranking POIs is taken in [5], where the importance of a POI takes into account the presence of other relevant nearby POIs.

Queries involving sets of spatio-textual objects have been investigated in [6, 27]. Given a set of keywords and, optionally, a user location, the goal is to identify sets of POIs that collectively satisfy the query keywords while minimizing the maximum distance or the sum of distances between each other and to the query.

More recently, other works have focused on discovering regions

of interest w.r.t. a specified category or set of keywords, where the importance of a region is determined based on the number or the total weight of relevant POIs it contains. In [19], density-based clustering is applied to identify regions with high concentration of POIs of certain categories, collected and integrated from several Web sources. A method for extracting scenic routes from geotagged photos uploaded on sites such as Flickr and Panoramio is presented in [3]. Discovering and recommending regions of interest based on user-generated data is also addressed in [20]. The quality of a recommended area is determined based on the portion of the contained POIs that can be visited within a given time budget. Other variations of queries for discovering interesting regions include the subject-oriented top-k hot region query [21] and the maximizing range sum query [10]. The region is defined by a rectangle or circle with a maximum size constraint, and the goal is to maximize the score of the relevant POIs contained in it.

The most closely related work to our approach is [7], which proposes the length-constrained maximum-sum region query. Given a set of POIs in an area and a set of keywords, this query computes a region that does not exceed a given size constraint and that maximizes the score of the contained POIs that match the query keywords. The query assumes an underlying road network, in which the POIs are included as additional vertices, and the returned region has the form of a connected subgraph of this network with arbitrary shape. The problem is shown to be NP-hard, and approximation algorithms are proposed.

This problem is similar to our setting; in both cases, the goal is to discover interesting parts of a road network that are associated with large number of POIs relevant to a given set of keywords. However, in [7], the result is a single connected subgraph of the road network, maximizing the score of contained points, while our method returns a ranked list of streets that are not necessarily connected and are ordered according to their density w.r.t. POIs relevant to the query. Moreover, we additionally consider the problem of describing these discovered streets by means of a diversified set of photos, which is not addressed in [7] or any of the other works mentioned before.

Finally, in a different line of research, other works have applied probabilistic topic modeling on user-generated spatio-textual data and events to associate urban areas with topics and patterns of user mobility and behavior [18, 15].

2.2 Search Results Diversification

Information retrieval engines often try to improve the utility of the search results by taking into account not only their relevance to the user's query, but also their dissimilarity, offering thus a range of alternatives, which comes handy in situations where the true intent of the user is unknown or many highly similar objects exist. Stated in an abstract manner, the content-based diversification problem is to determine a set of objects that maximizes an objective function with two components, the relevance and the diversity.

While there exist many different formulations (refer to [16, 13] for classification), the most well-known is the MaxSum problem, where the goal is to maximize the weighted sum of two components, the total relevance of objects, and the sum of pairwise diversities among the objects. Similar to other diversification problems, MaxSum is NP-hard as it is related to the dispersion problem [22]. Therefore, various greedy heuristics are proposed. Typically, they incrementally construct the diversified result set by choosing at each step the object that maximizes a certain scoring function. The most well-known function is the *maximal marginal relevance* (mmr) [8]. An evaluation of various object scoring functions and different heuristics can be found in [26].

Since [8], several works addressed other diversification problems, such as taxonomy/classification-based diversification [1], [25] or multi-criteria diversification [12]. Another related work is the coverage problem [14], where the goal is to select a set of diverse objects that cover the entire database.

3. IDENTIFYING STREETS OF INTEREST

We first formulate the problem of identifying interesting streets, and then present our proposed approach.

3.1 Problem Definition

A road network is a directed graph $G = (\mathcal{V}, \mathcal{L})$, where the set of vertices $\mathcal{V}$ contains street intersections or breakpoints in streets, and the set of links $\mathcal{L}$ contains street segments (between intersections or breakpoints) represented as line segments. Each vertex $v \in \mathcal{V}$ is associated with its coordinates (x_v, y_v). The length $len(\ell)$ of a segment $\ell \in \mathcal{L}$ is computed as the Euclidean distance between its endpoints. We also consider the set of streets $\mathcal{S}$, where each street $s \in \mathcal{S}$ comprises a set of consecutive segments (a simple path on G). Each segment $\ell \in \mathcal{L}$ belongs to a unique street s, and we denote this relationship by $\ell \in s$. The length $len(s)$ of street s is the sum of the length of its segments.

Moreover, we define an additional data source $\mathcal{P}$, being a set of POIs. Each POI $p \in \mathcal{P}$ is defined by a tuple $p = \langle (x_p, y_p), \Psi_p \rangle$, where (x_p, y_p) are the coordinates of the POI, and Ψ_p is a set of keywords describing this POI (e.g., keywords derived from its name, description, tags). The distance $dist(p, \ell)$ of a POI p to a line segment ℓ is defined as the minimum Euclidean distance between POI location (x_p, y_p) and any point on ℓ. Accordingly, the distance of POI p to a street s is the minimum distance of p to any segment of s, i.e., $dist(p, s) = \min_{\ell \in s} dist(p, \ell)$.

To measure the interest of a street segment w.r.t. a given set of keywords, we use the notion of *mass*, which refers to the number of relevant POIs that exist in its proximity. Then, we rank segments according to their *mass density*, to account also for the different lengths of each segment. We formally define these concepts below.

DEFINITION 1 (SEGMENT MASS). *For a given set of keywords Ψ and a distance threshold ϵ, the* mass *of segment ℓ is the number of POIs within distance ϵ that contain at least one keyword from Ψ:*

$$mass(\ell \,|\, \Psi, \epsilon) = |\{p \in \mathcal{P} : dist(p, \ell) \leq \epsilon \;\&\; \Psi_p \cap \Psi \neq \emptyset\}|.$$

Note that this definition can be straightforwardly adapted in the case that POIs have different weights.

DEFINITION 2 (SEGMENT INTEREST). *The* interest *of segment ℓ is its mass density, i.e., the ratio of ℓ's mass over the size of the area within distance ϵ around ℓ:*

$$int(\ell \,|\, \Psi, \epsilon) = \frac{mass(\ell \,|\, \Psi, \epsilon)}{2\epsilon\, len(\ell) + \pi\epsilon^2}.$$

Given this definition for the interest of a segment, there exist several alternatives for defining the interest of an entire street. Here, we use a simple definition, as stated below.

DEFINITION 3 (STREET INTEREST). *Given Ψ and ϵ, the interest of a street s is the maximum interest among its segments, i.e.:*

$$int(s \,|\, \Psi, \epsilon) = \max_{\ell \in s} int(\ell \,|\, \Psi, \epsilon). \quad (1)$$

Based on these definitions, a k-SOI query returns the k most interesting streets.

Problem 1. [k-SOI] Assume a set of streets $\mathcal{S}$, forming a road network G, and a set of POIs $\mathcal{P}$. The k-Streets of Interest (k-SOI)

query $q = \langle \Psi, k, \epsilon \rangle$, where Ψ is a set of keywords, k a positive integer, and ϵ a distance threshold, returns a set of k streets S^k such that for each $s' \notin S^k$ it holds that $int(s' \mid \Psi, \epsilon) \leq \min_{s \in S^k} int(s \mid \Psi, \epsilon)$.

3.2 The SOI Algorithm

In what follows, we present the SOI (Streets Of Interest) algorithm. We assume a given query $q = \langle \Psi, k, \epsilon \rangle$; for brevity, Ψ and ϵ are omitted when it is clear from the context.

3.2.1 Methodology and Indices

The SOI algorithm for processing k-SOI queries operates in a manner reminiscent of top-k processing algorithms [17]. It progressively examines segments of streets and POIs until it can establish that the k-SOI can be determined solely from the information already collected. Specifically, SOI maintains a *seen lower bound* LB_k on the interest of the k best streets encountered so far, and an *unseen upper bound* UB on the interest of any street for which no segment has been encountered yet. As more segments are considered, LB_k progressively increases, while UB progressively decreases. When LB_k becomes not smaller than UB, the examination stops, since the k-SOIs are those streets with interest not smaller than LB_k.

Under the aforementioned strategy, there are two issues to address: (1) how to compute the seen lower bound LB_k and the unseen upper bound UB, and (2) how to expedite the termination condition, $LB_k \geq UB$.

We address the former first, as its solution gives intuition about the latter. Based on the definition of street interest (Equation 1), we can compute bounds on the interest of a street s by considering directly the segments. The following lemma suggests a method to compute LB_k and UB; their exact definition is presented later.

LEMMA 1. *Consider a subset of segments $\mathcal{L}_{seen} \subseteq \mathcal{L}$. Then, for a street s it holds that:*

$$int(s) \geq \max_{\ell \in s \cap \mathcal{L}_{seen}} int(\ell), \qquad \text{if } s \cap \mathcal{L}_{seen} \neq \emptyset$$

$$int(s) \leq \max_{\ell \in \mathcal{L} \setminus \mathcal{L}_{seen}} int(\ell), \qquad \text{if } s \cap \mathcal{L}_{seen} = \emptyset.$$

PROOF. For the first case observe that $s \cap \mathcal{L}_{seen} \subseteq s$, and thus $\max_{\ell \in s \cap \mathcal{L}_{seen}} int(\ell) \leq \max_{\ell \in s} int(\ell) = int(s)$. For the second case observe that $\mathcal{L} \setminus \mathcal{L}_{seen} \supseteq s$, and thus $\max_{\ell \in \mathcal{L} \setminus \mathcal{L}_{seen}} int(\ell) \geq \max_{\ell \in s} int(\ell) = int(s)$. □

Consider a seen street s, meaning that one of its segments has been encountered, i.e., $s \cap \mathcal{L}_{seen} \neq \emptyset$. The first case of Lemma 1 implies that a lower bound on the interest of s can be extracted from the largest interest of its seen segments, or, more practically, from the largest lower bound on the interest of any of its seen segments.

On the other hand, consider an unseen street s, i.e., $s \cap \mathcal{L}_{seen} = \emptyset$. The second case of Lemma 1 implies that an upper bound on the interest of s can be extracted from the largest possible interest among unseen segments, or, more practically, from an upper bound on the largest possible interest among unseen segments.

In other words, Lemma 1 directly addresses the former issue. However, it also suggests how to address the latter. Suppose we have encountered a set of segments $\mathcal{L}_{seen}$. What segments should it contain so as to increase the chances of satisfying the termination condition? To obtain a high seen lower bound LB_k, the first case of Lemma 1 suggests putting in $\mathcal{L}_{seen}$ segments with high interest. Moreover, to obtain a small unseen upper bound UB, the second case of Lemma 1 suggests leaving out from $\mathcal{L}_{seen}$ segments with low interest. Therefore, the algorithm should try to visit segments with large interest first.

As it is impractical to directly retrieve segments by interest (that would require precomputation for all possible k-SOI queries, i.e., for arbitrary ϵ, Ψ), we need a way to identify promising segments having large interest. To this end, we employ the following data structures.

- A *spatial grid index* with arbitrary cell size storing all POIs. Within each cell c, there is a *local inverted index* on the set of keywords among the cell POIs. The entry for keyword ψ is a list of POIs sorted increasingly on POI id.
- A *global inverted index* on the set of all keywords. The entry for keyword ψ is a list of $\langle c, numPOIs \rangle$ entries sorted decreasingly on $numPOIs$, which is the number of POIs within cell c that contain keyword ψ.
- A *cell-to-segment map* that stores for each grid cell the segments that pass through it. At query time when ϵ is known, the map is augmented to contain for each cell all segments that are within distance ϵ. We denote the augmented list for cell c as $\mathcal{L}_\epsilon(c)$.
- A *segment-to-cell map* that stores for each segment the grid cells that it intersects. At query time when ϵ is known, the map is augmented to contain for each segment all cells that are within distance ϵ. We denote the augmented list for segment ℓ as $\mathcal{C}_\epsilon(\ell)$.
- A *list of segments* sorted increasingly on their length.

Note that since street segments and POIs are relatively static, these data structures can be created and maintained offline.

3.2.2 Algorithm Description

In what follows, we discuss the case of a query specifying a single keyword, i.e., $\Psi = \{\psi\}$. Intuitively, we look for segments that have large $mass$ and small len. Thus, given the above data structures, we look for segments that (1) are close (within distance ϵ) to cells with large number of relevant POIs (satisfying ψ), (2) are close to many cells, and (3) have small len. The first two factors combined contribute to the $mass$, while the third directly to len. Therefore, the algorithm considers segments according to the following three ranked source lists constructed, in part, at query time.

SL_1**:** Contains all cells sorted decreasingly on the number of POIs with keyword ψ. This is essentially the list of the global inverted index for keyword ψ.

SL_2**:** Contains all segments sorted decreasingly on the number of cells within distance ϵ to them (the cells each segment intersects when enlarged by ϵ).

SL_3**:** Contains all segments, sorted increasingly on their length.

The algorithm proceeds iteratively, considering in each iteration either the next cell from SL_1 or the next segment from SL_2 or SL_3. Each source list can be accessed in a round robin fashion; the correctness of our method is not affected by the access strategy. In practice, we alternate between SL_1 and SL_3, trying to balance the number of segments considered from each source; each cell access results in the access of multiple segments, while each segment access causes the visit of multiple cells. We only access segments via the second source SL_2 in the case that a few segments with a large number of neighboring cells exist.

During the processing of k-SOI, a segment can be in three possible states. Initially, a segment is *unseen*, meaning that the algorithm

has not considered it via any source. An efficient algorithm would leave many segments in the unseen phase. Then, when a segment is first retrieved, it is put into the *partial* state, meaning that some, but not all, POIs near it that satisfy the query have been accounted for. For each partial segment ℓ, we maintain two pieces of information: (1) the count $mass^-(\ell)$ of relevant POIs seen so far that satisfy the query, and (2) a list $toVisit$ indicating which neighboring cells (and consequently their POIs) to visit. Based on these, we can compute a lower bound on ℓ's interest as:

$$int(\ell) \geq int^-(\ell) = \frac{mass^-(\ell)}{2\epsilon\, len(\ell) + \pi\epsilon^2}.$$

Once all relevant POIs have been processed (equivalently, all neighboring cells have been visited), the segment is in the *final* state, where its exact interest is known.

Algorithm 1 presents the pseudocode for the `SOI` algorithm. During initialization, `SOI` prepares the three source lists. Particularly, it builds SL_1 by examining the global inverted index (lines 1–3); for the simple case of one keyword ψ, SL_1 is essentially the inverted list $I[\psi]$. Moreover, source list SL_3 corresponds to the list of segments, while SL_2 is extracted from the augmented segment-to-cell map (lines 4–7).

Then, `SOI` proceeds to the main filtering phase (lines 8–24), where segments from the source lists are examined until the termination condition $LB_k \geq UB$ holds; initially LB_k and UB are set to zero and infinity, respectively (line 9).

Assume that cell c via source list SL_1 is to be accessed (lines 11–15). For this cell we examine the segments that are within distance ϵ (lines 13–14), and for each segment we determine the number of POIs with keyword ψ that are within distance ϵ so as to update their interest score given the contents of cell c. Specifically, we employ the cell-to-segment map to determine all segments $\mathcal{L}_\epsilon(c)$ that are within distance ϵ to cell c (line 13). For each such segment ℓ, we invoke the procedure `UpdateInterest` (line 14), whose pseudocode is also depicted. After the procedure returns, we set the next source list to consider according to round robin (line 15).

Procedure `UpdateInterest` first checks whether cell c has been visited for ℓ and immediately returns if so. Otherwise, it removes c from the $toVisit$ list. Then it visits the local inverted index of cell c and retrieves the list for keyword ψ. For each POI p in the list, `UpdateInterest` checks whether it is within distance ϵ to segment ℓ, and if true increments $mass^-(\ell)$ by one.

Now, assume that segment ℓ is to be accessed (lines 16–21), either via source list SL_3 or SL_2. Its exact interest will be determined, changing its state to final. Using the segment-to-cell map, we visit sequentially all neighboring cells of ℓ, and invoke procedure `UpdateInterest` (lines 18–19). A cell visited during some segment access, may be again visited due to another segment's access or via a direct cell access via source list SL_1. The next step is to properly set the next source list to consider (lines 20–21).

After each access, cell or segment, algorithm `SOI` checks whether the termination condition applies. It first computes an upper bound UB on the interest of any *unseen* segment (line 22). Let $top(SL_1)$, $top(SL_2)$, $top(SL_3)$ denote the top items in the corresponding sources lists that are to be accessed next. Due to the second case of Lemma 1, it computes the unseen interest upper bound as:

$$UB = \frac{top(SL_1) \cdot top(SL_2)}{2\epsilon\, top(SL_3) + \pi\epsilon^2}.$$

`SOI` also computes a lower bound LB_k on the interest of the k-SOIs based on the first case of Lemma 1 (lines 23–24). It maintains all seen segments in a ranked list $\mathcal{L}_{seen}$ sorted decreasingly on the interest lower bound $int^-()$. Let $\mathcal{L}_{seen}[i]$ denote the first i items

Algorithm 1: Algorithm `SOI`

```
   Input: network G, streets S, query q = ⟨Ψ, k, ϵ⟩
   Output: k-SOIs S^k
   ▷ build source list SL_1
1  foreach cell c that has an entry in I[ψ] for some ψ ∈ Ψ do
2      |P_Ψ(c)| ← min{|P_c|, Σ_{ψ∈Ψ} I[ψ][c]}
3      insert entry ⟨c, |P_Ψ(c)|⟩ in SL_1
   ▷ build source lists SL_3, SL_2
4  foreach segment ℓ do
5      insert entry ⟨ℓ, len(ℓ)⟩ in SL_3
6      |C_ϵ(ℓ)| ← |c ∈ C | dist(c, ℓ) ≤ ϵ|
7      insert entry ⟨ℓ, |C_ϵ(ℓ)|⟩ in SL_2
   ▷ filtering phase
8  SL ← SL_1                                          ▷ next source list
9  LB_k ← 0; UB ← ∞
10 while UB > LB_k do
11     if SL = SL_1 then
12         c ← pop(SL)                                ▷ retrieve cell
13         foreach ℓ ∈ L_ϵ(c) do                      ▷ ℓ within distance ϵ to c
14             UpdateInterest(ℓ, c, Ψ)
15         SL ← SL_2                                  ▷ set next source list
16     else
17         ℓ ← pop(SL)                                ▷ retrieve segment
18         foreach c ∈ C_ϵ(ℓ) do                      ▷ ℓ within distance ϵ to c
19             UpdateInterest(ℓ, c, Ψ)
20         if SL = SL_2 then SL ← SL_3                ▷ set next source list
21         else SL ← SL_1
22     UB ← top(SL_1)·top(SL_2) / (2ϵ top(SL_3)+πϵ^2)
23     μ ← min_i : |{s | ∃ℓ ∈ L_seen[i], ℓ ∈ s}| = k
24     LB_k ← int^-(ℓ_μ)
   ▷ refinement phase
25 foreach ℓ ∈ L_seen do
26     foreach c ∈ C_ϵ(ℓ) do                          ▷ ℓ within distance ϵ to c
27         UpdateInterest(ℓ, c, Ψ)
28 return S^k ← extract k-SOIs from L_seen
```

Procedure UpdateInterest(ℓ, c, Ψ)

```
1 if c ∉ ℓ.toVisit then return                        ▷ c is already visited for ℓ
2 remove c from ℓ.toVisit
  ▷ traverse lists c.I(ψ), ∀ψ ∈ Ψ synchronously
3 foreach p ∈ ⋃_{ψ∈Ψ} c.I(ψ) do
4     mass^-(ℓ) ← mass^-(ℓ) + 1
```

in the list. Then, LB_k is set to the interest lower bound of the μ-th ranked segment ℓ_μ provided that μ is the smallest index such that the segments of $\mathcal{L}_{seen}[\mu]$ belong to k distinct streets, i.e.,

$$LB_k = int^-(\ell_\mu), \text{ for } \mu = \min_i : |\{s \mid \exists \ell \in \mathcal{L}_{seen}[i], \ell \in s\}| = k.$$

The accesses on source lists stop as soon as $UB \leq LB_k$. At that point, it is guaranteed that the result to k-SOI can be extracted from the segments in $\mathcal{L}_{seen}$. To identify the streets with the top-k interest, a refinement phase begins (lines 25–28). The exact interest of each segment in $\mathcal{L}_{seen}$ is computed by invoking `UpdateInterest` as necessary. The extraction of the streets with the highest interest, i.e., the k-SOI, is then straightforward.

A final note concerns the case of multiple keywords Ψ in the query. The `SOI` algorithm changes in only two places. The first is when source list SL_1 is built. It is necessary to account for POIs that have any keyword among those in Ψ. `SOI` looks within the global inverted index, for each entry $I[\psi][c]$ corresponding to the entry for cell c in the list for keyword ψ. This entry contains the count of POIs in cell c that have keyword ψ. Adding these counts for all keywords provides an upper bound to the number of POIs within c that have any keyword among Ψ. The minimum of this number and the total number $|\mathcal{P}_c|$ of POIs in the cell (line 2) is then inserted into SL_1. The second change is in the UpdateInterest procedure. To compute for segment ℓ the exact number of POIs within cell c that satisfy Ψ, lists $c.I[\psi]$ for each $\psi \in \Psi$ are tra-

versed in parallel; recall that the lists are sorted by POI id. For each encountered POI, the mass of ℓ is incremented.

4. DESCRIBING STREETS OF INTEREST

Having identified the k-SOIs in the road network, the next step is to provide summarized information to describe them. Section 4.1 formalizes the problem, while Section 4.2 describes our solution.

4.1 Problem Definition

We begin by formalizing the problem, and then present details about the measures considered.

4.1.1 Problem Statement

In the following, we assume an additional data source $\mathcal{R}$, being a set of geo-tagged photos. Each photo $r \in \mathcal{R}$ is defined by a tuple $r = \langle (x_r, y_r), \Psi_r \rangle$, specifying its location and a set of keywords (its tags); the distance of a photo to a segment or a street is defined as in the case of a POI.

To describe a SOI, we exploit its related photos. For each street s, these are the photos that are located within distance ϵ, i.e. $\mathcal{R}_s = \{r \in \mathcal{R} : dist(r, s) \leq \epsilon\}$. However, the size of $\mathcal{R}_s$ can typically be quite large. Thus, the problem is to select a relatively small subset of k photos ($k \ll |\mathcal{R}_s|$) to present as an overview for the street s. To avoid redundancy and repetition, we formulate the problem as a MaxSum diversification problem, where a subset of items is selected from a set in such a way as to maximize both their relevance to a given need and their pairwise dissimilarity. More formally, the problem can be defined as a bi-criteria optimization problem, aiming at optimizing an objective function $\mathcal{F}$ that comprises a *relevance* component and a *diversity* component [16, 26].

Let R^k be a subset of $\mathcal{R}_s$ of size k, and let $rel(R^k)$ and $div(R^k)$ be two functions that measure, respectively, the relevance and the diversity of the contents of R^k. Then, the problem is to select among all possible subsets R^k the one that maximizes the function $\mathcal{F}$ defined as follows:

$$\mathcal{F}(R^k) = (1 - \lambda) \cdot rel(R^k) + \lambda \cdot div(R^k) \tag{2}$$

where $\lambda \in [0, 1]$ is a parameter determining the tradeoff between relevance ($\lambda = 0$) and diversity ($\lambda = 1$).

Problem 2. [SOI Diversification] Given a street s with an associated set of photos $\mathcal{R}_s$, where each photo has a geolocation and a set of keywords, select a subset R^k of $\mathcal{R}_s$ containing k photos such that the objective function $\mathcal{F}$ is maximized, i.e.:

$$R^k = \underset{R \subseteq \mathcal{R}_s, |R|=k}{\arg\max} \mathcal{F}(R) \tag{3}$$

We proceed to define the functions $rel(R^k)$ and $div(R^k)$ for our problem. Note that the value of k throughout this section refers to the number of photos describing a street, and is thus unrelated to the value of k in Section 3 which refers to the number of SOIs.

4.1.2 Spatio-Textual Relevance and Diversity

The function $\mathcal{F}$ described above provides a generic criterion for selecting a subset of items that are both relevant and diverse w.r.t. a given query. In our context, this needs to take into account both the spatial and the textual description of a street. In particular, the *spatial aspect* of a street s is determined by the locations of its associated photos $\mathcal{R}_s$. The *textual aspect* of s is captured by a keyword frequency vector Φ_s, which describes the strength of each keyword associated with s; we denote as Ψ_s the set of keywords with non-zero frequency in Φ_s. Note that there are many ways to derive the keyword frequency vector of a street; for example, we can extract it directly from a textual description, or from the keywords of its neighboring POIs and/or photos.

The relevance and diversity functions of a set R^k of photos should thus capture both aspects. Assuming a weight parameter $0 \leq w \leq 1$ between the two aspects, we define:

$$rel(R^k) = \frac{w}{k} \sum_{r \in R^k} spatial\_rel(r) + \frac{1 - w}{k} \sum_{r \in R^k} textual\_rel(r) \tag{4}$$

and

$$div(R^k) = \frac{2w}{k(k-1)} \sum_{r, r' \in R^k} spatial\_div(r, r') + \frac{2(1-w)}{k(k-1)} \sum_{r, r' \in R^k} textual\_div(r, r'). \tag{5}$$

Notice that the relevance of set R^k is defined as the sum of the spatial and textual relevance of each photo $r \in R^k$, whereas the diversity of R^k is the sum of the pairwise spatial and textual diversity over all photo pairs $r, r' \in R^k$. To balance the different number of summands in $rel(R^k)$ and $div(R^k)$, we normalize them using the fractions $\frac{1}{k}$ and $\frac{2}{k(k-1)}$, respectively. Next, we define the four functions that account for spatio-textual relevance and diversity.

Spatial relevance and diversity. For a point query, the spatial distance of an item to the query point would typically constitute a natural way to measure relevance. However, in our case, ranking the photos of a street according to their distance from it does not generally provide an indicative criterion for judging relevance. Thus, we select instead a different criterion, based on spatial coverage. The intuition is that high density of photos in an area can be considered as an indication of "importance"; thus, the selection of photos should be biased towards those areas. Accordingly, we define the spatial relevance of a photo based on the number of other photos contained in its neighborhood. Furthermore, we divide this number to the total number of photos associated with the street, in order to obtain a normalized value in the range $[0, 1]$.

DEFINITION 4 (SPATIAL RELEVANCE). *Assuming a radius ρ for the neighborhood of a photo r, we define the spatial relevance of r w.r.t. the street s as:*

$$spatial\_rel(r) = \frac{|\{r' \in \mathcal{R}_s : dist(r, r') \leq \rho\}|}{|\mathcal{R}_s|}. \tag{6}$$

The spatial diversity of a pair of photos r, r' can be defined by means of their spatial distance. For normalization, we divide the distance of the pair with $maxD(s)$, which is the largest possible distance between any two photos associated with s. Value $maxD(s)$ is computed as the length of the diagonal of the minimum bounding rectangle, extended with a buffer of size ϵ, that encloses s. Thus:

DEFINITION 5 (SPATIAL DIVERSITY). *We define the spatial diversity of two photos r, r' associated with street s as:*

$$spatial\_div(r, r') = \frac{dist(r, r')}{maxD(s)}. \tag{7}$$

Textual relevance and diversity. The textual relevance of a photo r measures the similarity of its textual description Ψ_r to the textual aspect of street s as captured by its keyword frequency vector Φ_s.

DEFINITION 6 (TEXTUAL RELEVANCE). *The textual relevance of a photo r to street s is defined as:*

$$textual\_rel(r) = \frac{\sum_{\psi \in \Psi_r} \Phi_s(\psi)}{\|\Phi_s\|_1}, \quad (8)$$

where $\|\Phi_s\|_1 = \sum_{\psi \in \Psi_s} \Phi_s(\psi)$ is a normalization term.

Finally, we define textual diversity by means of the Jaccard distance.

DEFINITION 7 (TEXTUAL DIVERSITY). *We define the textual diversity of two photos r, r' as the Jaccard distance of their sets of keywords, i.e.:*

$$textual\_div(r, r') = 1 - \frac{|\Psi_r \cap \Psi_{r'}|}{|\Psi_r \cup \Psi_{r'}|}. \quad (9)$$

4.2 The ST_Rel+Div Algorithm

Next, we present the ST_Rel+Div (Spatio-Textual Relevance and Diversity) algorithm. We first describe the methodology and index used, and then we present how the algorithm selects the diversified subset of photos more efficiently.

4.2.1 Methodology and Indices

The ST_Rel+Div algorithm follows the standard greedy methodology for solving MaxSum diversification problems. It builds the diversified set R^k iteratively, selecting at each step the photo that maximizes the maximum marginal relevance function mmr (see Section 2). Suppose that set R has been constructed, where $|R| < k$. Then, the photo from $\mathcal{R}_s \setminus R$ to be inserted next is the one that maximizes the function:

$$mmr(r) = (1-\lambda) \cdot rel(r) + \frac{\lambda}{k-1} \cdot \sum_{r' \in R} div(r, r'). \quad (10)$$

Functions $rel(r)$ and $div(r)$ are as defined in Section 4.1.2 taking into account the spatial and textual aspects.

The main issue with applying this heuristic methodology in our context is the computational complexity of mmr. A naïve implementation would compute a large number of spatial and textual photo-to-street relevances and photo-to-photo diversities. In particular, computing mmr at each iteration, requires $O(|\mathcal{R}_s|)$ computations for its first component, and $O(|R||\mathcal{R}_s|)$ for the second. Even though some of these computations need not be repeated across iterations, the total computational cost can be prohibitive, especially when the set $\mathcal{R}_s$ is large.

Consequently, the goal of ST_Rel+Div is to efficiently evaluate each component of the objective function mmr towards retrieving the best candidate at each iteration. For this purpose, we construct an index as described in the following. We use an index structure that combines a spatial grid with inverted indices in each cell. Each cell $c_{i,j}$ in the grid has side length $\frac{\rho}{2}$, and contains the following information:

- a list of the photos in the cell, denoted as $c_{i,j}.\mathcal{R}$
- an inverted index $c_{i,j}.I$, where the terms are the keywords appearing in the photos in this cell, and each postings list $c_{i,j}.I[\psi]$ contains those photos that have the keyword ψ (we denote as $c_{i,j}.\Psi$ the set of keywords present in $c_{i,j}.I$)
- the maximum ($c_{i,j}.\psi_{max}$) and minimum ($c_{i,j}.\psi_{min}$) number of keywords for the photos in this cell.

Next, we show how this index is used to derive lower and upper bounds for each of the components of the objective function mmr. Note that the described index, although similar to the grid index used in Section 3.2, is distinct, indexing a different dataset, i.e. the set of photos instead of POIs.

4.2.2 Computing Bounds

Using the index, we can derive, for any of the photos within a cell, upper and lower bounds for each of the components of the mmr function. The ST_Rel+Div algorithm exploits these bounds while computing the mmr function in order to iterate over the cells instead of individual photos and to prune the search space more quickly, identifying the next candidate that optimizes the mmr criterion. In particular, we need to derive lower and upper bounds for the following: (a) spatial and textual relevance of a cell to a street and (b) spatial and textual diversity of a cell to a photo. We elaborate on each below.

Cell spatial relevance. Consider a cell $c_{i,j}$. The spatial relevance of a photo r w.r.t. street s is defined according to Equation 6. Moreover, recall that the length of each side of a cell in the spatial grid is $\frac{\rho}{2}$. Hence, each photo $r \in c_{i,j}.\mathcal{R}$ covers at least all other photos in the same cell and at most all photos that are no more than two cells away. Accordingly, we derive the following lower and upper bounds for the cell-to-street spatial relevance:

$$spatial\_rel^-(c_{i,j}) = \frac{|c_{i,j}.\mathcal{R}|}{|\mathcal{R}_s|}, \quad (11)$$

$$spatial\_rel^+(c_{i,j}) = \frac{\sum_{\Delta i, \Delta j \in [-2,2]} |c_{i+\Delta i, j+\Delta j}.\mathcal{R}|}{|\mathcal{R}_s|}. \quad (12)$$

Cell textual relevance. Consider a cell c. The textual relevance of a photo $r \in c.\mathcal{R}$ w.r.t. street s is defined according to Equation 8. We seek to construct keyword sets $\Psi^-(c \mid s), \Psi^+(c \mid s)$, which are subsets of $c.\Psi$, such that when they substitute set Ψ_r in Equation 8, the obtained values bound the textual relevance of any photo $r \in c.\mathcal{R}$. In other words, we obtain the following bounds of $textual\_rel(r)$:

$$textual\_rel^-(c) = \frac{\sum_{\psi \in \Psi^-(c \mid s)} \Phi_s(\psi)}{\|\Phi_s\|_1}, \quad (13)$$

$$textual\_rel^+(c) = \frac{\sum_{\psi \in \Psi^+(c \mid s)} \Phi_s(\psi)}{\|\Phi_s\|_1}. \quad (14)$$

We next describe how to build the keyword sets $\Psi^-(c \mid s), \Psi^+(c \mid s)$. Based on the information stored in the index, each photo $r \in P(c)$ may contain at least $c.\psi_{min}$ and at most $c.\psi_{max}$ keywords. Hence, sets $\Psi^-(c \mid s), \Psi^+(c \mid s)$ should obey these cardinality constraints.

For set $\Psi^-(c \mid s)$, we should choose the fewest possible keywords from $c.\Psi$, i.e., $c.\psi_{min}$, and make sure they have as low frequencies in Φ_s as possible. Therefore, we select up to $c.\psi_{min}$ keywords from $c.\Psi$ that do not appear in Ψ_s, and, if necessary (so as to satisfy the minimum cardinality constraint), we additionally select keywords from $c.\Psi$ with the lowest frequencies in Φ_s.

On the other hand, for set $\Psi^+(c \mid s)$, we select up to $c.\psi_{max}$ keywords from $c.\Psi$ that also appear in Ψ_s, and if necessary (so as to satisfy the minimum cardinality constraint), we arbitrarily choose additional keywords from $c.\Psi$.

Cell-to-photo spatial diversity. Assume a grid cell c and a photo r. The lower and upper bounds of the spatial diversity between r and any photo $r' \in c.\mathcal{R}$ are determined by respective bounds on the distance of r and cell c. Therefore, we have:

$$spatial\_div^-(c,r) = \frac{mindist(r,c)}{maxD(s)}, \tag{15}$$

$$spatial\_div^+(c,r) = \frac{maxdist(r,c)}{maxD(s)}, \tag{16}$$

where functions $mindist(r,c)$ and $maxdist(r,c)$ return the minimum and maximum distance, respectively, between r and any point within cell c.

Cell-to-photo textual diversity. Assume a grid cell c and a photo r. We determine the lower and upper bounds of the textual diversity between r and any other photo $r' \in c.\mathcal{R}$. We follow a similar rationale as for the cell textual relevance, and construct keyword sets $\Psi^+(c\,|\,r)$, $\Psi^-(c\,|\,r)$ so that when they substitute $\Psi_{r'}$ in Equation 9 they provide a lower and an upper bound, respectively, on the textual diversity of any photo $r' \in c.\mathcal{R}$.

For the lower bound for the textual diversity, we constuct set $\Psi^+(c\,|\,r)$ maximizing the common keywords with Ψ_r. Specifically, we insert in $\Psi^+(c\,|\,r)$ up to $c.\psi_{max}$ common keywords, and if necessary, we additionally insert keywords from $c.\Psi$ until we obtain at least $c.\psi_{max}$ keywords in $\Psi^+(c\,|\,r)$. Therefore, the textual diversity of any $r' \in c.\mathcal{R}$ is lower bounded by:

$$textual\_div^-(c,r) = 1 - \frac{|\Psi^+(c\,|\,r) \cap \Psi_r|}{|\Psi^+(c\,|\,r) \cup \Psi_r|} = \\ = \begin{cases} 1 - \frac{|c.\Psi \cap \Psi_r|}{|\Psi_r| + c.\psi_{min} - |c.\Psi \cap \Psi_r|}, & \text{if } |c.\Psi \cap \Psi_r| < c.\psi_{min} \\ 1 - \frac{\min(|c.\Psi \cap \Psi_r|, c.\psi_{max})}{|\Psi_r|}, & \text{if } |c.\Psi \cap \Psi_r| \geq c.\psi_{min} \end{cases} \tag{17}$$

For the upper bound for the textual diversity, we constuct set $\Psi^-(c\,|\,r)$ minimizing the common keywords with Ψ_r. Specifically, we insert in $\Psi^-(c\,|\,r)$ up to $c.\psi_{min}$ keywords from $c.\Psi$ that are not in Ψ_r, and if necessary, we insert additional keywords from $c.\Psi$ until we obtain at least $c.\psi_{min}$ keywords. Therefore, the textual diversity of any $r' \in c.\mathcal{R}$ is upper bounded by:

$$textual\_div^+(c,r) = 1 - \frac{|\Psi^-(c\,|\,r) \cap \Psi_r|}{|\Psi^-(c\,|\,r) \cup \Psi_r|} = \\ = \begin{cases} 1 - \frac{c.\psi_{min} - |c.\Psi \setminus \Psi_r|}{|r.\Psi| + |c.\Psi \setminus \Psi_r|}, & \text{if } |c.\Psi \setminus \Psi_r| < c.\psi_{min} \\ 1, & \text{if } |c.\Psi \setminus \Psi_r| \geq c.\psi_{min} \end{cases} \tag{18}$$

4.2.3 Algorithm Description

Algorithm 2 shows the pseudocode for the `ST_Rel+Div` algorithm, which incrementally builds the result set R^k by adding, at each step, the next best photo, denoted as $next\_r$, that maximizes the objective function mmr defined in Equation 10. However, the distinguishing factor of `ST_Rel+Div`, compared to a naïve algorithm that directly computes the mmr function for each photo, is that instead of evaluating individual photos, it first considers entire grid cells. Specifically, at each step, in the filtering phase (lines 4–9), it iterates over the cells and computes for each cell the lower and upper bound of the objective function mmr (lines 5–7), by applying the corresponding bounds presented in 4.2.2.

Then, any cell having an upper bound that is lower than the lower bound of another cell is discarded (line 9). The remaining cells are organized in a priority queue ordered descending on their mmr upper bound. Only the photos belonging to the remaining cells are processed in the refinement phase (lines 10–17). For each examined photo, its exact value for the objective function is calculated (line 13). During this process, if the upper bound of a cell is lower than the value computed for an examined photo, this cell is also discarded (lines 14–17). The process continues until the priority queue contains no cells.

Algorithm 2: Algorithm `ST_Rel+Div`

Input: Set of all relevant photos $\mathcal{R}_\ell$, integer k, the `ST_Rel+Div` index $\mathcal{I}$
Output: Diversified subset R^k

1 $R^k \leftarrow \emptyset$
2 $\mathcal{C} \leftarrow$ the grid cells in $\mathcal{I}$
▷ select next candidate
3 **while** $|R^k| < k$ **do**
▷ filtering phase
4 $B_{min}, B_{max} \leftarrow \emptyset$ ▷ maps to store cell bounds
5 **foreach** cell $c \in \mathcal{C}$ **do**
6 $B_{min}(c) \leftarrow mmr(c)^-$ ▷ using bounds in
7 $B_{max}(c) \leftarrow mmr(c)^+$ ▷ Section 4.2.2
8 $mmr\_min \leftarrow \max_{c \in \mathcal{C}} B_{min}(c)$
9 $C \leftarrow \{c : B_{max}(c) \geq mmr\_min\}$ ▷ list of candidate cells
▷ refinement phase
10 **while** $C \neq \emptyset$ **do**
11 $c \leftarrow top(C)$ ▷ visit next cell with largest $B_{max}(c)$
12 **foreach** $r \in c.\mathcal{R}$ **do**
13 $v \leftarrow mmr(r)$ ▷ compute exact value
14 **if** $v > mmr\_min$ **then**
15 $mmr\_min \leftarrow v$ ▷ refine bound
16 $C \leftarrow C \setminus \{c : B_{max}(c) < mmr\_min\}$
17 $next\_r \leftarrow r$
18 $R^k \leftarrow R^k \cup next\_r$ ▷ add next best photo
19 **return** R^k

Table 1: Datasets used in the evaluation.

Dataset	Num of segm.	Min segm. length (m)	Max segm. length (m)	Num of POIs
London	113,885	0.93	5,834.71	2,114,264
Berlin	47,755	0.06	6,312.96	797,244
Vienna	22,211	1.35	9,913.42	408,712

5. EXPERIMENTAL EVALUATION

We have conducted an experimental evaluation using real-world data comprising road networks, POIs and photos. The datasets cover the areas of three European capital cities, London, Berlin and Vienna, and were collected from various Web sources, in particular: (a) road networks from OpenStreetMap, (b) POIs from DBpedia, OpenStreetMap, Wikimapia and Foursquare, and (c) photos from Flickr and Panoramio. Table 1 presents statistics about the datasets. All algorithms were implemented in Java and experiments were run on a machine with an Intel Core i7 2400 CPU and 8GB RAM.

The primary focus of this paper is to propose efficient algorithms for the tasks of identifying and describing SOIs, as defined in Sections 3.1 and 4.1. A detailed *performance* study is presented in Section 5.2. Nonetheless, it is also very important to gauge the *effectiveness* of our methods in achieving their goals. Therefore, in Section 5.1, we present the results of an empirical study of both tasks, identification and description.

5.1 Effectiveness Study

The goal of this section is to highlight the effectiveness of our methods in identifying and describing streets of interest.

5.1.1 Identifying Streets of Interest

We focus on a particular SOI retrieval scenario: determine streets in Berlin that are interesting for "shopping". As ground truth, we

(a) Alte/Neue Schönhauser Straße site

(b) Kurfürstendamm site

(c) Friedrichstraße site

Figure 2: Main shopping sites in Berlin and their most important streets.

Table 2: Comparison of identified top SOIs for "shops" in Berlin.

Top-10 SOIs	Source #1	Source #2
1. **Neue Schönhauser Straße**	**Tauentzienstraße**	Kurfürstendamm
2. Rosenthaler Straße	Fasanenstraße	**Tauentzienstraße**
3. Mäusetunnel	**Friedrichstraße**	**Potsdamer Platz**
4. **Münzstraße**	**Alte/Neue Schönhauser Straße**	**Friedrichstraße**
5. **Potsdamer Platz Arkaden**	**Münzstraße**	**Alte/Neue Schönhauser Straße**
6. **Friedrichstraße**		
7. Mulackstraße		
8. **Alte Schönhauser Straße**		
9. Weinmeisterstraße		
10. **Tauentzienstraße**		

assume two authoritative Web sources about top shopping destinations in Berlin[1] [2]. Each source provided a (non-ranked) list of 5 streets, displayed in the two last columns in Table 2.

In our SOI algorithm, we set the query parameters to Ψ={"shop"}, $k = 10$, $\epsilon = 0.0005^\circ \approx 55m$, i.e., looking for the top 10 streets that have a large concentration of "shop"-related POIs within 55 meters from them. The ranked list is shown in the first column of Table 2. Streets that are common among the 10-SOIs and the two sources are shown in bold. For both sources, we retrieve 4 out of 5 streets; therefore, our method has a recall (at rank 10) of 0.8.

A closer inspection of the results though, suggests that our method has actually better recall (and precision). All streets included in Table 2 belong to one of four main shopping sites in Berlin, near Alte/Neue Schönhauser Straße, near Kurfürstendamm, near Friedrichstraße, and near Potsdamer Platz. Figure 2 illustrates the first three areas on the map (the last one is a public square that has no important adjacent shopping streets) and highlights the streets included in Table 2. Green color indicates that the street is in the 10-SOIs and in at least one of the source lists (true positives); orange means it is in the 10-SOIs but not in any source (false positives); blue indicates it is in a source but not in the 10-SOIs (false negatives).

It should be apparent that all orange streets near Alte/Neue Schönhauser Straße are actually valid results as they are adjacent to the main streets in the respective area, and further have lots of little shops in their vicinity. Similarly, the orange street near Friedrichstraße is also valid as it is a pedestrian undergound tunnel with shops. Regarding the blue streets in the Kurfürstendamm site (which is analogous to Champs-Élysées in Paris), we should note that Kurfürstendamm appears in the 20-SOIs. The reason they are ranked lower is that in their vicinity the density of shops is lower, as they essentially house big luxury brands. This could be addressed by exploiting additional metadata to assign different weights to the POIs.

[1] http://www.tripadvisor.com/Travel-g187323-s405/Berlin:Germany:Shopping.html

[2] http://www.globalblue.com/destinations/germany/berlin/top-five-shopping-streets-in-berlin

5.1.2 *Describing Streets of Interest*

To study the effectiveness of our method in selecting appropriate photos for describing a particular SOI, we focus on the popular Oxford Street in London and seek for 3 photos. Since deriving the ground truth from the collected crowd-sourced data is not possible, we choose to empirically test the result of our method against simpler techniques that select photos based on spatial, textual information or their combination, and consider relevance, diversity, or their combination. More precisely, we compare our method ST_Rel+Div that combines spatio-textual relevance and diversity, against 8 other techniques depicted in Table 4. Symbols S and T denote, respectively, that spatial and textual information is considered, while Rel and Div indicate, respectively, that relevance and diversity are taken into account; our method considers all factors and information, hence its name ST_Rel+Div.

For a visual inspection of the results, Figure 3 shows the 3-photo summary of Oxford Street, according to methods S_Rel, T_Rel and ST_Rel+Div, respectively. In the first method, all selected photos are located outside HMV, an entertainment retailing company, and are in fact near-duplicates. The reason for this seems to be that the particular location attracts a large number of photos, due to the release of popular movies, music albums and other similar events, thus creating a high density spot. For the second method, all results are photos from a particular demonstration that took place along Oxford Street. This bias was introduced due to the high frequency of the corresponding tags, thus resulting in a higher rank for photos having those tags. On the other hand, observe that in our method (Figure 3(c)) the result comprises different kinds of photos, achieving both high relevance and diversity: one photo outside HMV, another from the aforementioned demonstration, and a third photo showing a view of the street undergoing construction work.

For a quantifiable measure of effectiveness, we use the objective function of Equation 2 ($\lambda = 0.5$, $w = 0.5$), that provides a balanced score reflecting the relevance and diversity of both spatial and textual information included in the photo summary. For the top SOI in the considered cities, Table 3 presents the scores achieved by each method; the value is normalized with respect to that of the

Table 3: Objective scores (Equation 2 after normalization).

Method	London	Berlin	Vienna
S_Rel	0.831	0.726	0.508
S_Div	0.923	0.982	0.961
S_Rel+Div	*0.982*	0.953	0.911
T_Rel	0.708	0.367	0.219
T_Div	0.831	0.811	0.895
T_Rel+Div	0.949	0.848	0.919
ST_Rel	0.776	0.367	0.279
ST_Div	0.913	*0.986*	*0.961*
ST_Rel+Div	**1.000**	**1.000**	**1.000**

ST_Rel+Div method. In all cities, our method achieves the highest normalized score (shown with bold), often by a large margin (up to 4.5x). It is worth mentioning that there is no clear runner-up, as S_Rel+Div is second best for London, while ST_Div is for Berlin and Vienna (second highest values shown with italics).

5.2 Performance Study

We next evaluate the efficiency of the proposed methods.

5.2.1 *Identifying Streets of Interest*

Methods. First, we evaluate the performance of our proposed SOI algorithm for solving k-SOI. To the best of our knowledge, no approaches have been proposed in previous works for the specific problem addressed (see Problem 1). Hence, we compare the performance of SOI to a baseline implementation, denoted as BL. Specifically, BL uses only the spatial grid index to efficiently compute the interest of every segment, and then determines the k-SOIs.

Parameters. Throughout our experiments, we set the distance threshold ϵ to a fixed value ($\epsilon = 0.0005^\circ \approx 55m$). We study the effect of the number k of SOIs requested, and the number $|\Psi|$ of keywords in the query. To construct the keyword set, we select the first $|\Psi|$ keywords among $\{religion, education, food, services\}$. The resulting accumulated number of relevant POIs is shown in Table 4. In each experiment, we vary one parameter while setting the other to its default value ($k = 50$, $|\Psi| = 3$).

Table 4: Relevant POIs according to $|\Psi|$.

Dataset	$\|\Psi\| = 1$	$\|\Psi\| = 2$	$\|\Psi\| = 3$	$\|\Psi\| = 4$
London	10,445	32,682	113,211	202,127
Berlin	1,969	10,506	47,950	78,310
Vienna	1,678	7,660	25,695	41,484

Metrics. The objective of our evaluation is to analyze the performance of SOI compared to BL. Therefore, we measure the total execution time of both methods. For SOI we further break it down into time spent during list construction, filtering, and refinement.

Results. Figure 4 presents the evaluation results on the three datasets for all settings considered. Note that the bars for SOI are divided into the time spent in each of the three phases discussed.

The value of k has a small effect on all algorithms. Particularly for SOI the execution time slightly increases with k. SOI outperforms BL by a factor around 2.1–3.2 for London, 1.6–2.1 for Berlin, and 1.1–2.5 for Vienna. An important observation is that our method is more efficient for larger datasets, such as London (see Table 1).

The value of $|\Psi|$ has no effect in BL. On the other hand, the execution time of SOI increases with $|\Psi|$ as more POIs become relevant (see Table 4) and thus more cells and segments have to be visited. For example in London, the number of cells SOI visits increases from 5% to 13% of the total cells. As a result, SOI outperforms BL by a factor that varies from 1.1 up to 18.

(a) Spatial relevance (S_Rel).

(b) Textual relevance (T_Rel).

(c) Spatio-textual relevance and diversity (ST_Rel+Div).

Figure 3: Selected photos under different criteria.

Note that the selected keywords are quite general; they only serve for benchmark purposes in extreme settings. For example, when $|\Psi|$=4, we are essentially trying to rank streets that have churches, schools, restaurants or various services within their neighborhood. As a result, around 60% of all street segments are relevant (SOI manages to prune half of them). In practice, the user would pose more selective keywords.

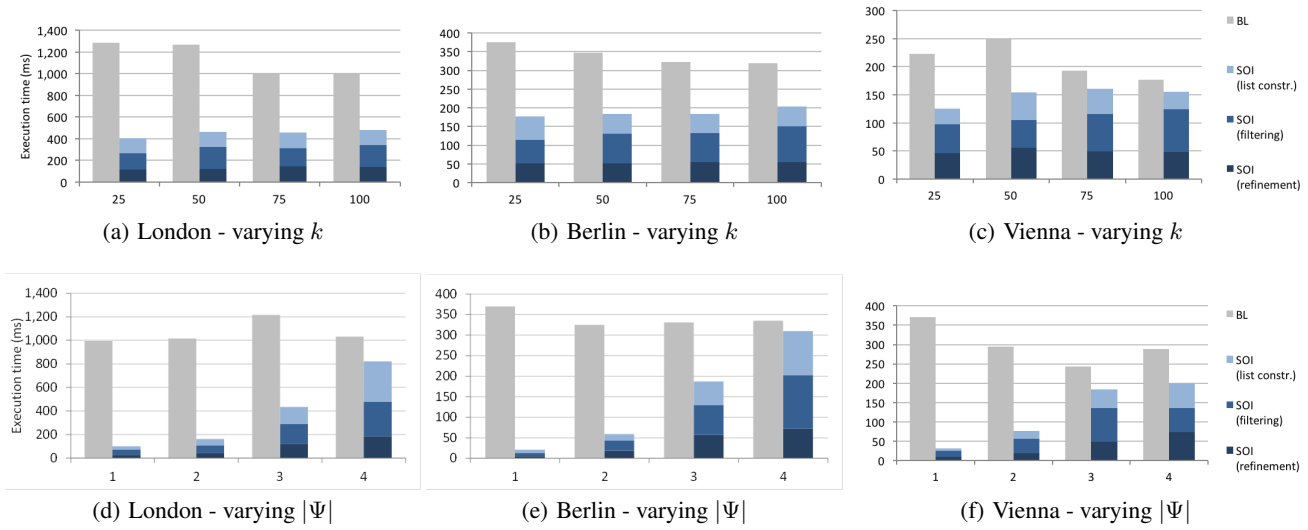
5.2.2 *Describing Streets of Interest*

Methods. Next, we evaluate the performance of the ST_Rel+Div algorithm compared to a baseline method (BL) which, similarly to ST_Rel+Div, constructs the diversified result set iteratively, but examining all photos in each iteration instead of operating on the grid cells and using the bounds presented in Section 4.2.

Parameters. We fix the values of the distance parameters to $\epsilon = 0.0005^\circ$ and $\rho = 0.0001^\circ$, and we vary: (a) the number k of photos requested (default k=20), (b) the weight λ between relevance and diversity (default λ=0.5), and (c) the weight w between the spatial and textual components (default w=0.5). For each dataset, we randomly selected one of the returned k-SOIs in the previous experiments. The number of nearby photos for the cases of London, Berlin and Vienna was, respectively, 6572, 788, and 1584.

An interesting discussion concerns the selection of an appropriate value for parameter λ.[3] We can think of the relevance-diversity trade-off as follows. In order to increase the diversity of the result set (the *return*), we have to sacrifice its relevance (the *investment*). Typically, diversity starts to increase quickly in the beginning (when relevance is still high), but its rate slowly decreases, meaning that a greater reduction in relevance is required to achieve

[3]The other parameter in Equation 2, w, is application dependent.



(a) London - varying k (b) Berlin - varying k (c) Vienna - varying k

(d) London - varying $|\Psi|$ (e) Berlin - varying $|\Psi|$ (f) Vienna - varying $|\Psi|$

Figure 4: Experimental results for the `SOI` algorithm.

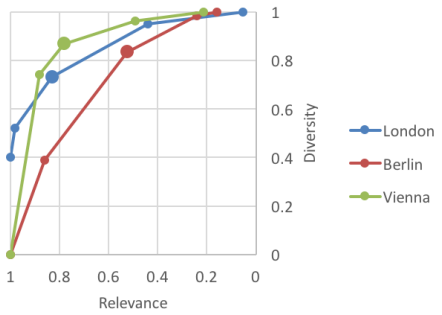


Figure 5: Trade-off between relevance and diversity ($w = 0.5$).

the same increase in diversity. So the goal is to figure out an acceptable investment that is "value for money".

Figure 5 depicts the normalized relevance (Equation 4) and diversity (Equation 5) scores of the constructed photo summary for the top SOI in the three cities for various values of λ; note that the relevance axis is reversed. As we go from bottom left to top right, the value λ increases from 0 to 1 in increments of 0.25, and thus relevance decreases while diversity increases. The larger marker indicates the value $\lambda = 0.5$. In all cities, λ values around 0.5 achieve the best relevance-diversity trade-off. For example, in Vienna $\lambda = 0.5$ suggests that by sacrificing 0.22 units of normalized relevance we achieve a diversity of 0.87 normalized units. These findings justify our selection of 0.5 as the default value for λ.

Metrics. As previously, we compare the total execution time of the `ST_Rel+Div` algorithm and the `BL` method.

Results. The results are shown in Figure 6. The pruning achieved by `ST_Rel+Div` via bounds computed for the grid cells drastically reduces the execution time in all experiments. `ST_Rel+Div` outperforms `BL` by a factor that varies from 2 up to 64.

Moreover, it is worth noticing that `ST_Rel+Div` has response times of less than a second, in contrast to the `BL` method that typically requires several seconds to compute the results, thus being unsuitable for online exploration. In fact, the execution time is much higher for London, due to the fact that the selected segment in that case has a much higher number of associated photos, while the inverse holds for Berlin.

For both algorithms, execution time increases with k, as more iterations are performed; however, `ST_Rel+Div` shows much better scalability due to the pruning. These differences in performance also remain consistent when varying the parameters λ and w.

6. CONCLUSIONS

In this paper, we have addressed the problem of finding and exploring *Streets of Interest* based on Points of Interest and photos characterized by geolocation and keywords. The problem addressed is twofold. Given a set of keywords, we first rank streets according to relevant nearby POIs. To that end, we define an interest score for a street, and we present an efficient algorithm that returns the top-k interesting streets. Then, we select for each discovered street a small, diversified set of photos. We formulate this as a diversification problem for spatio-textual objects, and we present an efficient algorithm that performs a greedy search using a spatio-textual grid to speed up the selection of candidates. Our experimental results on real-world data from several Web sources show that the proposed algorithms drastically reduce the computation time, allowing for online discovery and exploration of interesting parts of the road network. In the future, we plan to enhance the diversification criteria with visual features extracted from the photos, as well as to provide route recommendations based on the discovered streets of interest.

Acknowledgements

This work was partially supported by the EU Projects GEOSTREAM (FP7-SME-2012-315631) and City.Risks (H2020-FCT-2014-653747).

7. REFERENCES

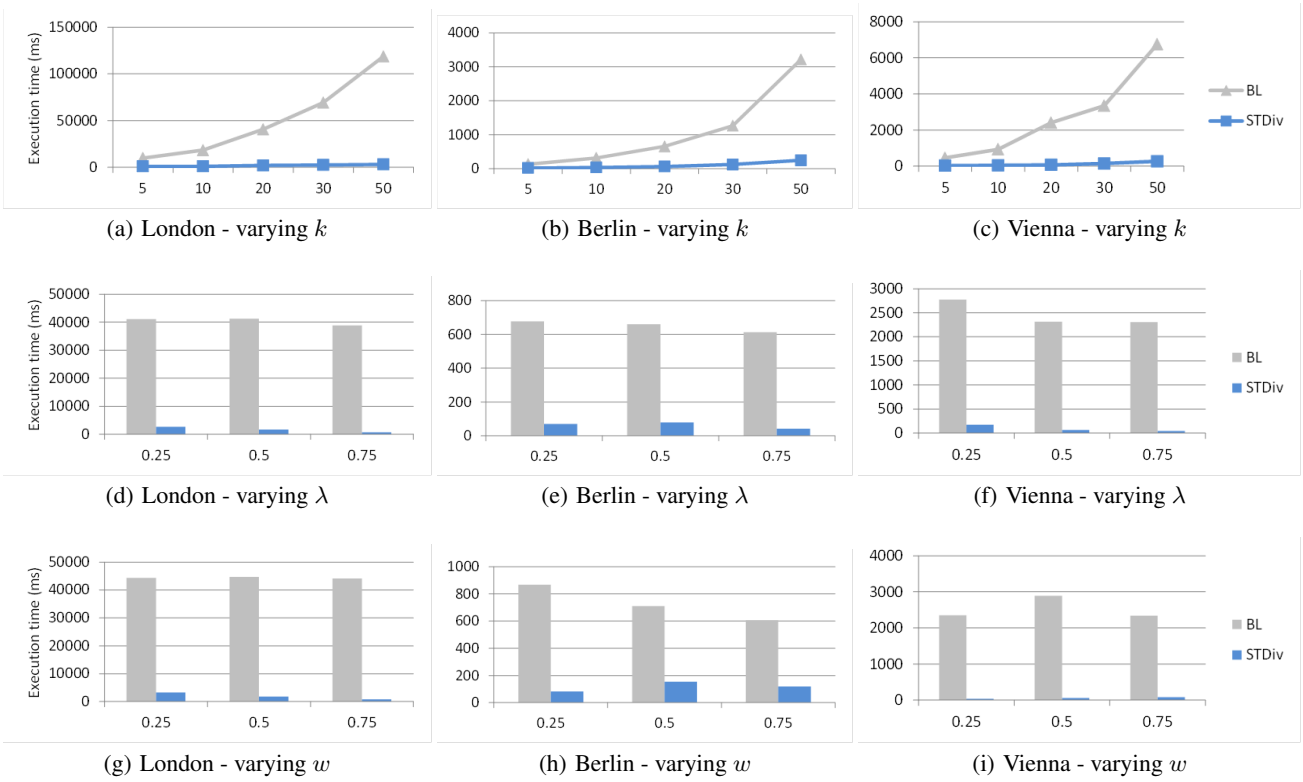
[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[2] R. Ahuja, N. Armenatzoglou, D. Papadias, and G. J. Fakas. Geo-social keyword search. In *SSTD*, pages 431–450, 2015.

[3] M. Alivand and H. H. Hochmair. Extracting scenic routes from VGI data sources. In *GEOCROWD*, pages 23–30, 2013.

[4] N. Armenatzoglou, S. Papadopoulos, and D. Papadias. A general framework for geo-social query processing. *PVLDB*, 6(10):913–924, 2013.

[5] X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. *PVLDB*, 3(1):373–384, 2010.



(a) London - varying k (b) Berlin - varying k (c) Vienna - varying k

(d) London - varying λ (e) Berlin - varying λ (f) Vienna - varying λ

(g) London - varying w (h) Berlin - varying w (i) Vienna - varying w

Figure 6: Experimental results for the `ST_Rel+Div` algorithm.

[6] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *SIGMOD*, pages 373–384, 2011.

[7] X. Cao, G. Cong, C. S. Jensen, and M. L. Yiu. Retrieving regions of interest for user exploration. *PVLDB*, 7(9):733–744, 2014.

[8] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[9] L. Chen, G. Cong, C. S. Jensen, and D. Wu. Spatial keyword query processing: An experimental evaluation. *PVLDB*, 6(3):217–228, 2013.

[10] D. Choi, C. Chung, and Y. Tao. Maximizing range sum in external memory. *TODS*, 39(3):21:1–21:44, 2014.

[11] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[12] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen. Multi-dimensional search result diversification. In *WSDM*, pages 475–484, 2011.

[13] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010.

[14] M. Drosou and E. Pitoura. Disc diversity: result diversification based on dissimilarity and coverage. *PVLDB*, 6(1):13–24, 2012.

[15] L. Ferrari, A. Rosi, M. Mamei, and F. Zambonelli. Extracting urban patterns from location-based social networks. In *LBSN*, pages 9–16, 2011.

[16] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[17] I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4), 2008.

[18] F. Kling and A. Pozdnoukhov. When a city tells a story: urban topic analysis. In *SIGSPATIAL*, pages 482–485, 2012.

[19] G. Lamprianidis, D. Skoutas, G. Papatheodorou, and D. Pfoser. Extraction, integration and analysis of crowdsourced points of interest from multiple web sources. In *GEOCROWD*, pages 16–23, 2014.

[20] D. Laptev, A. Tikhonov, P. Serdyukov, and G. Gusev. Parameter-free discovery and recommendation of areas-of-interest. In *SIGSPATIAL*, pages 113–122, 2014.

[21] J. Liu, G. Yu, and H. Sun. Subject-oriented top-k hot region queries in spatial dataset. In *CIKM*, pages 2409–2412, 2011.

[22] S. Ravi, D. Rosenkrantz, and G. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[23] J. B. Rocha-Junior and K. Nørvåg. Top-k spatial keyword queries on road networks. In *EDBT*, pages 168–179, 2012.

[24] Y. Tao, X. Hu, D. Choi, and C. Chung. Approximate MaxRS in spatial databases. *PVLDB*, 6(13):1546–1557, 2013.

[25] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient computation of diverse query results. In *ICDE*, pages 228–236, 2008.

[26] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174, 2011.

[27] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *ICDE*, pages 688–699, 2009.