Hybrid Quantized Resource Descriptions for Geospatial Source Selection

Stefan Kufer
University of Bamberg
D-96047 Bamberg, Germany
stefan.kufer@uni-bamberg.de

Andreas Henrich
University of Bamberg
D-96047 Bamberg, Germany
andreas.henrich@uni-bamberg.de

ABSTRACT

Location nowadays is an important aspect of the Web. One scenario in this respect are archives or collections of geo-tagged media items. More concretely, we can think of collections in the arts and humanities available via OAI-PMH (a protocol for metadata harvesting) on the web or web accessible personal media archives maintained in a peer-to-peer manner. In such scenarios for search problems, source selection becomes an important aspect. For example, we would like to access only those collections containing media items in a certain geospatial region (maybe we are interested in images from Shanghai only). Here, the geospatial search criterion allows for a high selectivity. What is needed in such a scenario are expressive and nevertheless compact representations or descriptions of the "geospatial footprint" of each collection. A minimum bounding rectangle would be a trivial but not very accurate option. Generally, summarization techniques for this purpose can be distinguished into three categories, geometric approaches, space partitioning approaches and hybrid approaches. In this work, we present novel hybrid techniques, which mostly apply a set of approximating minimum area rectangles for subspace description together with quantization techniques in order to increase the selectivity of the summaries and, at the same time, keep the storage requirements small.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Spatial databases and GIS – Summarization; Top-k retrieval in databases; Distributed retrieval; Peer-to-peer retrieval; H.3.3 [**Information Search and Retrieval**]: Selection process

General Terms

Algorithms, Performance

Keywords

Geographic Information Retrieval, Distributed Information Retrieval, Source Selection, Summarization, Quantization

1. INTRODUCTION

In various Web-based scenarios, source selection—and especially geospatial source selection—is an important aspect. As an example, assume a peer-to-peer based image sharing community. The single users provide personal media archives consisting of geo-tagged images. To allow for community-wide search facilities without a central instance, a PlanetP [1] like mechanism is assumed where each peer is maintaining summaries of all other peers in the community. In accordance with the desired search criteria, the summaries have to form a footprint of the data or documents maintained by each peer.

With respect to the geospatial queries considered in this paper, the scenario requires compact and expressive descriptions of the geospatial footprint of the images maintained by a peer[1]. Typical query types are region queries or a k-nearest-neighbor (kNN) queries. With a region query, all peers which could potentially maintain media items—images in our scenario—falling into the query region (a query polygon for example) have to be contacted. To perform kNN queries, as usual an algorithm which is implemented as range query with decreasing query radius is used. This algorithm will be described together with the peer ranking schemes in section 2.3.

In the present paper, we propose new hybrid quantized resource descriptions for the sketched scenario. These summaries have to describe the geospatial positions of the images of a peer (i.e. a set of points) as accurate as possible with small storage space requirements, because the summaries have to be stored on all peers and they have to be distributed in the network via a rumor spreading mechanism. The presented techniques extend work presented by Kufer et al. in [2] and [3]. In [2], techniques distinguishable into geometric approaches and space partitioning approaches are examined, whereas in [3] first hybrid approaches combining features of the aforementioned classes are introduced.

[1]Of course, each image can also be described by additional criteria, such as text, timestamps or low-level content features. These descriptions could be aggregated criteria-wise for all images a peer maintains, building a summary for each criterion. Hence, more sophisticated queries, such as searching for images picturing the Shanghai Tower in the sunset, could be supported. Nevertheless, for this paper, we solely focus on the low-dimensional geographic aspect.

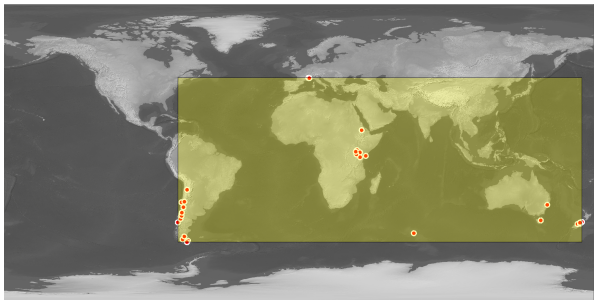

Figure 1: The simple MBR approach. Red dots denote the sample peer's data points.

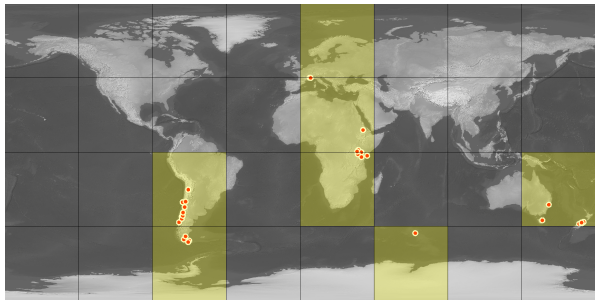

Figure 2: Grid$_r$ approach with 32 subspaces ($r = 4$). The highlighted cells contain data points.

In section 2, the different geospatial summarization techniques are presented. The existing approaches are briefly recapitulated in section 2.1. On that basis, the new hybrid techniques are described in section 2.2. The approaches are evaluated in section 3. Related work and potential application domains for geospatial resource descriptions will be discussed in section 4. Finally, section 5 concludes the paper.

2. RESOURCE DESCRIPTION FOR GEOGRAPHIC QUERIES

In our scenario, every peer maintains a set of images as media items. Each image has been enhanced with a geographic footprint, that is a single pair of latitude/longitude-coordinates. These geo-coordinates get mapped into a plane using the plate carrée projection, hence treating the lat/long-coordinates as x/y-coordinates (data points) in a two-dimensional Cartesian coordinate system. Consequently, the Euclidean distance is used for distance calculations. In Blank et al., [4], distance measures which are better suited for distance calculations between two points on the surface of the earth have been investigated, but showed no noticeable changes compared to the utilization of the Euclidean distance, which is computationally much more efficient. Thus, the resource description problem is reduced to encoding a set of two-dimensional data points effectively (accurate description) and efficiently (compact storage).

2.1 Categories of Summary Types

Previous work ([2, 3]) has investigated several techniques distinguishable into three categories.

2.1.1 *Geometric Approaches*

For the first class of summary types, the geometric approaches, one or multiple geometric shapes enclosing all of a peer's data points get calculated to represent the "point cloud" of the peer. Since the computation of approximated, concise representations of complex forms is a standard problem in a lot of computer science domains [5], many appropriate algorithms exist and are applicable for this type of summaries. One of the most basic techniques in this domain is the Minimum Bounding Rectangle (MBR) which encloses all data points in a rectangle of minimal size (see Fig. 1). Other examples would be a simple sphere or a small set of rectangles [5].

2.1.2 *Space Partitioning Approaches*

For space partitioning approaches, the data space gets globally segmented into a certain number of subspaces, identifiable per ID. Since the segmentation is global, it is the same for all peers. Thus, information about data occupation for the different subspaces (also called cells) can be stored in a peer's summary. For each cell, this information can be the number of data points the peer administers in this cell. An alternative are binary values whether a cell is "occupied" by data points or not, allowing for a finer space partitioning when using the same storage. A simple space partitioning technique is to lay a regular grid onto the data space and store binary information about cell occupancy. We will address this approach as Grid$_r$, r setting the number of grid rows, the number of columns being $2 \cdot r$ (see Fig. 2 for a visualization).

2.1.3 *Hybrid Approaches*

Hybrid approaches use geometric shapes and space partitioning in a two-stage approach. Depending on what is used first, two subclasses can further be distinguished. For the first subclass, in the primary step the data points will be approximated by one or multiple geometric shapes, before applying some space partitioning method to the base shapes. A simple example is an MBR which gets enhanced with an MBR-interior grid (MBRGrid$_r$). For the second subclass, first some sort of space partitioning is conducted, before possibly applying geometric shape computations for single subspaces (depending on the cell occupancy). A straightforward approach is to impose a regular grid onto the data space and calculate the (quantized) MBR for each cell occupied with data points (GridMBR$_r^b$ approach, b being the number of bits used for one MBR value, see Fig. 3; the more bits used, the higher the MBR accuracy).

In general, to achieve space efficiency, approaches of the first subclass encode binary information for subspace occupancy allowing for a finer partitioning, while approaches of the second subclass make use of quantization techniques to represent geometric shapes calculated for occupied single subspaces with fewer bits and a reasonable granularity.

2.2 Novel Hybrid Approaches

In the following, we introduce multiple novel hybrid approaches that are evaluated in section 3. We utilize quantization in order to reduce summary sizes not only for the novel techniques using space partitioning first (for these, we make use of quantization in greater extent), but also for the novel

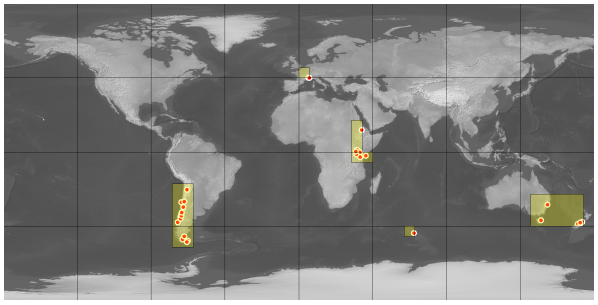

Figure 3: GridMBR$_r^b$ approach with $r = 4$ and $b = 3$, featuring quantized MBRs for each occupied cell.

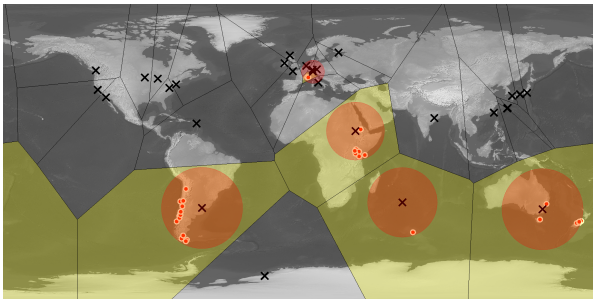

Figure 4: DFS$_n^b$ approach with 32 subspaces and 3 bits used for quantization ($n = 32$, $b = 3$).

technique employing geometric shapes first, hence for both subclasses of hybrid approaches.

2.2.1 *DFS$_n^b$*

The Ultra fine-grained summaries (UFS$_n$) evaluated in [2] and [3] apply a Voronoi-like space partitioning (based on n predetermined reference points distributed via rumor spreading) onto the data space. They store binary information for each cell indicating if the peer maintains data falling into that cell. In order to obtain good results, this space partitioning has to be adjusted to the data distribution of the data collection. Thus, reference points have to be chosen accordingly (see section 3.1 for details about origin and number of reference points).

The novel Distance fine-grained summaries (DFS$_n^b$) add the encoding of quantized distance information to the Voronoi-like space partitioning. In the first step, the data space globally gets segmented based on the n predetermined reference points. Each of a peer's data points is assigned to the cell of the reference point being closest to it. In a second step, for each cell, the maximum distance between the cell's reference point and any of its associated data points is calculated.

In order to address space efficiency, the calculated distances get quantized. Using b bits for encoding quantized distance information, we can distinguish 2^b different quantization steps, ranging from 0 to a certain threshold d_p^{max}.

Since the intra-cell distances may vary greatly between different peers, it is not suggestive to employ a global threshold for all peers, but to use an individual threshold d_p^{max} for each peer p, which gets encoded into the peer's summary. This also eliminates potential update problems of a global solution. Thus, the following is applied for every peer: For each occupied Voronoi cell, the maximum distance between its reference point and any of its associated data points in cell p is calculated. The biggest value over all these distances is taken as the peer's individual threshold d_p^{max}. See Fig. 4 for a visualization of DFS$_n^b$ summaries.

A peer's summary is represented by a bit vector. Voronoi cells not containing any data point are represented with 0, cells containing at least one data point are encoded with 1 followed by the quantized distance information for this cell.

2.2.2 *GridMAR$_r^{b,k}$*

For GridMAR$_r^{b,k}$, the data space globally gets segmented by imposing a regular grid onto the data space in the first place. Afterwards, for each grid cell containing one or more data points, a set of at most k quantized approximating Minimum Area Rectangles (MARs) enclosing all the cell's data points is computed. The MAR computation is done by a recursive MAR computation algorithm heuristically determining up to k rectangles such that all points are enclosed and the sum of the surface areas is minimal. This algorithm has been described in [3] and is based on work from Becker et al. [5]. The idea of using quantized rectangles within a grid cell was already used in [3], utilizing it to encode *single* MBRs, so the work presented here is taking the quantization approach to its next step. Specifically, a distinction between a potential data region (being the cell of a partitioning in general) and an actual data region is made. The latter is a set of cell-interior MARs enclosing all data points located in the potential data region.

In order to reduce storage for encoding the interior MARs, the presence of the potential data region is exploited (this approach has been used in [7] and was originally described in [8]). For encoding a single rectangle in a two-dimensional space, four values need to be stored, specifying the lower left and the upper right corner. If we use b bits to encode one of these values, 2^b positions can be distinguished on an axis of a data cell. Using these positions, we can encode approximated MARs (also called *coded actual data regions*), which are a bit larger than the real MARs, but require fundamentally less storage compared to using float values.

In addition, storage space can further be reduced by minimizing the number of MARs for each cell. As can be seen in Fig. 5, especially quantized MARs can be neighbored and aligned in a way that it is possible to describe the area they cover with fewer rectangles. Here, we can apply algorithms designed to decompose rectilinear polygons into a set of axis-aligned rectangles (see [9] for an overview regarding this domain). The problem of dissecting a polygon with holes into a minimum set of (possibly) *overlapping* rectangles is asserted to be NP-hard [10], but there are algorithms existent to dissect polygons with holes into a minimum set of *non-overlapping* rectangles [11], which are applied in this work. To use these algorithms, we dissolve the borders of the MARs found by the aforementioned recursive MAR algorithm. Neighbored areas get condensed and are treated as rectilinear polygons, which then are decomposed into a set of non-overlapping, axis-aligned rectangles. If the number of resulting MARs is smaller than the original number, we use the decomposition result to describe the cell's data points. See Fig. 6 for a visualization of GridMAR$_r^{b,k}$ summaries.

A peer's summary is again represented by a bit vector. Grid cells not containing any data point are represented with 0, occupied cells are encoded with 1 followed by the MAR

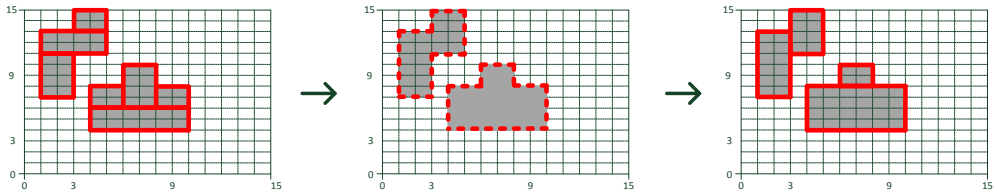

Figure 5: MAR minimization process: The area on the left can be described by using less than seven rectangles. To find the minimum number of non-overlapping rectangles, spatially neighbored rectangles get condensed into a rectilinear polygon (middle), which then gets decomposed into the desired result (right).

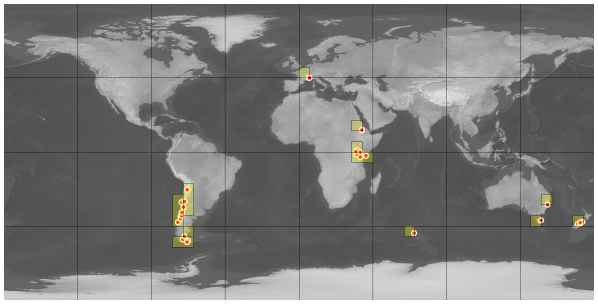

Figure 6: GridMAR$_r^{b,k}$ approach with 32 subspaces, 3 bits used for quantization and a maximum of 3 quantized MARs per cell ($r = 4$, $b = 3$, $k = 3$).

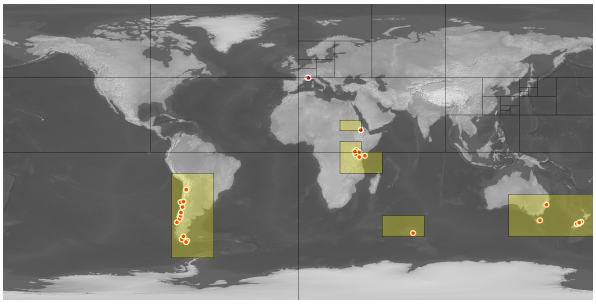

Figure 7: K-D-MAR$_n^{b,k}$ approach with 32 subspaces, 3 bits used for quantization and a maximum of 3 quantized MARs per cell ($n = 32$, $b = 3$, $k = 3$).

information for this cell. Since the number of MARs used for each cell can vary due to the MAR number minimization, at first the number of following MARs is encoded with $\lceil \log_2 k \rceil$ bits. Afterwards, each of the approximated MARs gets encoded with $4 \cdot b$ bits.

2.2.3 *K-D-MAR$_n^{b,k}$*

K-D-MAR$_n^{b,k}$ summaries take a k-d-tree like [12] space partitioning as a base, where the data space globally gets segmented into n rectangular cells of different size. The erratic space partitioning is learned from training data being inserted into the spatial access structure. At start, the data space is comprised of only one cell or bucket. The training data points sequentially are inserted into the bucket until a bucket overflow occurs, resulting in the original bucket being split into two. The process is reiterated until a desired amount of n buckets has been reached. As a split strategy, split dimension and split position need to be specified [7]. We use a simple approach, cyclically using longitude and latitude for the split dimension and splitting each cell in its middle for the split position. Similar to DFS$_n^b$, the space partitioning has to be adjusted to the data distribution of the data collection, thus the training data needs to be distributed proportionately (see section 3.1 for training data acquisition). Likewise DFS$_n^b$'s reference points, the global partitioning learned from training data needs to be distributed via rumor spreading. An exemplary visualization of the approach can be seen in Fig. 7.

Similar to GridMAR$_r^{b,k}$, for each occupied cell, the set of $\leq k$ MARs enclosing all the cell's data points is calculated, followed by the MAR minimization process. As a summary, bit vectors are utilized the same way as for GridMAR$_r^{b,k}$.

2.2.4 *MBR-MAR^{b,k}*

MBR-MAR^{b,k} differs from the aforementioned techniques as it does not take some sort of space partitioning as a base, but the MBR of the peer's data points. Based on this MBR, similarly to GridMAR$_r^{b,k}$ and K-D-MAR$_n^{b,k}$, a distinction between the potential data region (MBR) and the actual data region (set of MBR-interior, quantized MARs) is made.

In Kufer et al., [3], we examined the technique labelled MBRGrid$_r$ (also mentioned in subsubsection 2.1.3), which utilized an MBR as description base as well, but imposed an MBR-interior regular grid and simply binary encoded whether the grid cells were occupied. With the utilization of very fine-grained MBR-interior grids, long runs of zeros came up, since plenty of the cells are empty for fine-grained grids. Although we used *gzip*-compression for these representations (and tested alternatives as well), the storage requirements additional to the base MBR remained relatively high (see section 3.2). Hence, the idea behind MBR-MAR^{b,k} is to get rid of these zeros by just encoding areas that are occupied by data points. In some sense, this can be seen as a more informed approach to compression. See Fig. 8 for a visualization of MBR-MAR^{b,k} summaries.

Again, bit vectors are used as summaries. The first $4 \cdot 32$ bits encode the extents of the base MBR (lower left and upper right corner, float precision). Afterwards, the up to k MARs are sequentially encoded with $4 \cdot b$ bits each. Unlike for example GridMAR$_r^{b,k}$ (with possibly multiple cells containing data points), we do not need to encode the number of MARs, since there is no other superordinate structure containing quantized MARs besides the exterior MBR.

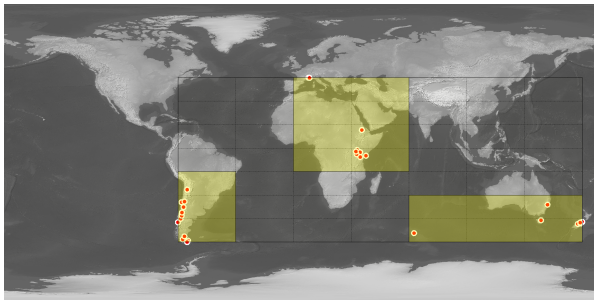

Figure 8: MBR-MAR^{b,k} approach with 3 bits used for quantization and a maximum of 3 quantized MARs inside the exterior MBR ($b = 3$, $k = 3$).

2.3 Peer Ranking and kNN Algorithm

A first search scenario based on the summaries described above would consider range queries. In this scenario, only peers for which the footprint described by the summary intersects the query region have to be contacted. A second scenario—and the one we consider in section 3—is the processing of kNN queries (each taking a single pair of lat/long coordinates as input). In this case a ranking of the peers is needed, which determines the sequence in which the peers have to be considered. In the following, we will describe how this ranking can be computed for the different summary types. The kNN algorithm using this ranking is described thereafter.

The peer ranking is conducted based on the information supplied by the summaries and with respect to the query location for the kNN query. For GridMAR$_r^{b,k}$, K-D-MAR$_n^{b,k}$ and MBR-MAR^{b,k} summaries, the ranking process is identical, since all these approaches at the very end encode several axis-aligned rectangular areas in a peer's summary.

To rank the peers for these approaches, for each peer, the ranking algorithm extracts the information from the peer's summary and constructs the rectangles containing the peer's data points, which can be done in an offline phase. At query time, the minimum distance between each rectangle and the query location is calculated (a query point located inside a rectangle has a distance of 0 for the corresponding rectangle). For each rectangle contained in the peer's summary, the distance information and the area covered by the rectangle are stored in an auxiliary data structure called R-Entry. The R-Entries of a peer get arranged in a queue to represent a peer during the ranking process. The R-Entries get sorted by 1) distance in ascending order and 2) in case of equal distances by covered area in ascending order (assuming a smaller area indicates a higher point density and therefore is more likely to include relevant data points than a larger area at the same distance).

To determine the ranking of two peers, the sorted R-Entries get compared one after another. If the first R-Entry of peer p_a is closer to the query location than the first R-Entry of peer p_b, p_a is ranked higher than p_b and vice versa. If both R-Entries yield the same distance, p_a is ranked higher than p_b, if p_a's R-Entry features a smaller covered area than p_b's R-Entry (and vice versa). If the area covered is the same for both R-Entries, the next R-Entries from the respective queues are compared, until a decision can be made. If the comparison does not result in a decision, a random ranking choice is made (a very rare case).

The DFS$_n^b$ ranking could be conducted basically the same way, just replacing the rectangular areas with the circular areas around the reference points of occupied cells. However, we refrain from this, for the following reason: On average, an occupied cell cannot be expected to feature a lot of data points, since space partitioning is adjusted to the underlying data collection (see section 3.1) and therefore data points (on average) should be distributed (relatively) evenly across the Voronoi cells, leaving only few data points in each cell. Knowing this, it is probable that a cell's data points are only located in few sectors of the maximum distance circle or "max ball" around a cell's reference point (Fig. 4 coincidently shows this). Furthermore, the "max balls" often tend to extend wide beyond the Voronoi cell borders, especially if the cells are elongated. Consequently, for DFS$_n^b$, the ranker originally used for UFS$_n$ summaries is utilized, which only uses binary information about cell occupancy:

The reference points c_j ($j \in \{0; n-1\}$) are sorted in ascending distance to the query location. The first element of the sorted list L corresponds to the reference point being closest to the query (implicitly forming the *query cluster*). If peer p_a administers documents in this query cluster while peer p_b does not, p_a is ranked higher than p_b and vice versa. If both peers feature the same value for the query cluster, the next element out of L is chosen and both peers are ranked according to their summary values for this very cluster. This procedure continues until a decision favoring one of the peers can be made or the end of L is reached, resulting in a random decision.

For DFS$_n^b$, the distance information encoded in the summaries is used for pruning purposes only (see kNN-algorithm below). Experiments evaluating both ranking schemes for DFS$_n^b$ show that the centroid based ranker leads to better results than a ranker employing distance information.

Using the described ranking techniques, the k nearest neighbors for a query point (we used $k = 50$ in our experiments) can be determined by a range query with decreasing query radius. First, the peers get ranked according to the ranking algorithm for the applied summaries. For each of the ten best ranked peers, the 50 data points closest to the query location get requested. From this set of up to 500 data points (some peers might maintain less than 50 data points), the 50 closest data points are determined to form the current intermediate top-50 result[2]. Subsequently, the ten peers which already have been considered get removed from the set of peers yet to be looked at. The distance of the currently fiftieth closest data point is taken for the query radius of the next round, in which ten further peers will be contacted if necessary. By taking the resource descriptions, the query point, and the query radius into account, some peers assuredly cannot have relevant data points and hence can be pruned (that is get removed from the set of peers to consider). After pruning, the ten top ranked peers of the remaining set are enquired for their 50 closest data points, possibly substituting some of the current top-50 data points. Afterwards, these ten peers get removed from the set of peers to consider and the next round commences. The procedure gets repeated until the set of peers to consider is empty,

[2]The consideration of ten peers at once is done to exploit the parallelism of our scenario. Obviously, this is a design parameter of the approach.

meaning the 50 nearest neighbors with respect to the query location have been determined.

3. EVALUATION

For our test data collection, in 2007, we crawled a large amount of publicly available, geo-referenced images which have been uploaded to Flickr (http://www.flickr.com). To assign images to peers, it is assumed that every Flickr user operates a peer of his own, resulting in an amount of 406,450 geo-referenced images being spread across 5,951 peers. There are few peers administering large proportions of the collection, which we will call big peers, and a lot of peers which maintain only one or slightly more images ("small peers"). Thus, the distribution of images to peers is heavily skewed which is a typical phenomenon for a lot of P2P settings [1]. The geographic distribution of the data points is uneven too, favoring North America, Europe, and Japan. Hence, it shows a usual distribution for geodata acquired from social networks, as for example depicted in [13].

3.1 Experimental Setting

For the queries, we use 200 image locations which are chosen in a two-step process. Firstly, a random peer is selected, and secondly, a randomly chosen geo-location of that peer is taken as query location. This simulates that all peers have the same probability for issuing a query.

Regarding the parameterization of the different approaches, it would be very difficult (and potentially unfair) to set parameters in a way that roughly the same summary sizes arise for all summary types, since the basic summary size is predetermined by intrinsic features of the different types. Thus, we varied the parameters such that summary computation and ranking can be performed in reasonable time (though we do not evaluate these aspects here) and parameterization seems suggestive in general.

k is set to 3, 6 and 9 for GridMAR$_r^{b,k}$ and K-D-MAR$_n^{b,k}$, and additionally to 12 and 15 for MBR-MAR^{b,k} (since there is only one superordinate space for which internal MARs get computed for this approach). For the hybrid techniques primarily using space partitioning, the number of subspaces is set to 512, 2,048 and 8,192 (meaning r is set to 16, 32 and 64 for GridMAR$_r^{b,k}$). For DFS$_n^b$, GridMAR$_r^{b,k}$ and K-D-MAR$_n^{b,k}$, parameter b (for the number of bits used to encode the quantized parts of the summaries) is varied from 2 to 4 to 6. The same applies for MBR-MAR^{b,k}, with the addition of varying b to 8, too (since on average, there are fewer values to be encoded in comparison to the other approaches). These parameters are generally suitable for our data or similarly distributed collections. If the data distribution significantly differs, parameters might have to be reconsidered.

For reasons of brevity, we will compare the different approaches with their respective "best" parameterizations only. For finding the "best" compromise between selectivity (fraction of peers which have to be contacted to determine the 50 closest points) and average summary size, we compared relative selectivity gains and relative average summary size growth while raising the respective parameters. As long as the relative selectivity gains in % are bigger than the relative summary size growth in %, we say it is beneficial to raise a parameter.

As mentioned in section 2.2, for DFS$_n^b$ and K-D-MAR$_n^{b,k}$, space partitioning needs to be adjusted to the underlying data collection. Two different strategies are applied. The first one is to randomly choose reference points (for DFS$_n^b$) respectively training data (for K-D-MAR$_n^{b,k}$) right out of the data collection. Simulating cases for which this might not be possible, the second strategy comes into play: There, data randomly is selected from the Geonames gazetteer[3] according to Gross Domestic Product per country statistics from Worldmapper[4]. That means, if a country would be responsible for 10% of the GDP worldwide, 10% of the reference points respectively training data points will be chosen from this country, using the Geonames gazetteer as a source for data points[5]. The GDP statistics are chosen since they better correlate with the data distribution of our data collection compared to other Worldmapper statistics such as income or internet usage. Applying these strategies, there should be a lot of small subspaces in areas with high global point density, whereas in areas with low global point density, there should be few big subspaces. Since the reference points as well as the training data are dependent on randomly chosen data, we always run ten different experiments with different seeds to minimize the effect of outliers.

The space efficiency of the different approaches is measured by analyzing average summary sizes. If it is beneficial, we apply some compression in order to reduce summary sizes (non-compressed summary sizes will be marked with * in the corresponding figures/tables), using Java's *gzip* implementation with default parameters. There are other, often costlier compression algorithms, which might result in improvements in the magnitude of about 10%. See [6] for a more detailed discussion regarding this topic.

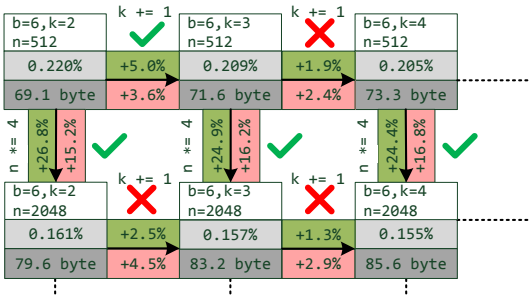

Figure 9: Snippet of the parameter optimization process for K-D-MAR$_n^{b,k}$ varying n and k.

3.2 Experimental Results

Fig. 9 exemplarily shows a snippet for the parameter optimization process for K-D-MAR$_n^{b,k}$, varying the parameters n and k. For this snippet, relative selectivity gain (green) always is bigger than relative summary size growth (red) when raising n, but mostly not for increasing k. In this snippet, the bottom left parameterization is the best one, since either directly or transitively it is superior compared to the others. In the following we describe some general observations for parameter variation:

For K-D-MAR$_n^{b,k}$ (both for using "internal" and "external" training data, that is right out of the data collection and

[3]http://www.geonames.org, checked on 14.07.2014.
[4]http://www.worldmapper.org, checked on 14.07.2014.
[5]For differentiation, we append "_e" to the respective technique acronym if data was chosen from the Geonames gazetteer as external source, for example DFS$_n^b$_e.

from the Geonames gazetteer, respectively), raising k (max. number of MARs) generally does not have great effects on selectivity. The average summary sizes only grow slowly—but mostly clearly overproportionate to selectivity. In contrast, raising both b (bits used for encoding quantized rectangles) and n (number of subspaces) results in notable selectivity improvements, with n being the more beneficial parameter for internal data and b being more beneficial for external data. For both cases, raising n (from 2,048 to 8,192) and b (from 4 to 6) from middle to high parameterization results in summary sizes growing so heavily (up to 20%) that their increase surpasses the selectivity gains. The small selectivity gains when increasing k are likely to be caused by the space partitioning being adapted to the underlying data collection. It appears that mostly one or two quantized MARs are sufficient to delimit the data points located in a certain cell adequately.

Looking at GridMAR$_r^{b,k}$, in general, a raise of all 3 parameters is effective (selectivity improves) and efficient (selectivity gain is higher than average summary size growth). Only increasing k from 4 to 6 is not efficient in most cases. Gains are biggest for boosting b, lowest for increasing k. Since the basic space partitioning is less accurate compared to K-D-MAR$_n^{b,k}$, more precise MARs are needed to overcome the lacking accuracy of the regular grid. Obviously, increasing the number of subspaces is always very efficient since it greatly boosts the base accuracy of the grid without increasing summary sizes too much due to *gzip*-compression.

Increasing both k and b is effective and efficient for MBR-MAR^{b,k}, raising b more so than raising k. As an exception, MBR-MAR^{b,k} summaries are on average smaller when not being compressed compared to being compressed (for the other approaches, it is always beneficial to compress).

For DFS$_n^b$ (both quantized and non-quantized), increasing the number of subspaces is always effective and efficient. When using quantization, raising b is always beneficial, too. Incrementing b is less effective compared to boosting the number of subspaces. This is because the distance information only gets used for pruning irrelevant peers. The quantized variant of DFS$_n^b$ offers almost the same selectivity as the non-quantized version and at the same time reduces summary sizes significantly.

The respective "best" parameterizations determined for the different approaches can be seen in Table 1 (indices in the left column along with their respective values for selectivity and space efficiency), complemented by some techniques evaluated in former work (old). The *baseline* shows the result when no summarization is conducted (that is the peers are represented directly by their data points) yielding optimum selectivity but low space efficiency.

If the *same parameterization* is used, generally summaries are more selective and bigger if training data (for K-D-MAR$_n^{b,k}$) or reference points (for DFS$_n^b$) originate directly out of the data collection. That is because in this case, more subspaces are occupied on a single peer's base, leading to bigger summaries featuring more information.

Comparing the new approaches among themselves for the *respective best parameterizations* (see Fig. 10), K-D-MAR$_n^{b,k}$ features best selectivity. Even though K-D-MAR$_n^{b,k}$ summary sizes are bigger on average in comparison with most remaining approaches, summary sizes are still moderate. Taking training data right out of the data collection for the respective best parameterizations results in more selective and smaller summaries. Interestingly, K-D-MAR$_n^{b,k}$_e (*external* data) offers significantly better performance selectivity compared to quantized DFS$_n^b$ (*internal* data), though at the price of (underproportionate) bigger summary sizes. Furthermore, GridMAR$_r^{b,k}$, though not conducting any adjustment to the global point distribution for space partitioning, closes in near to DFS$_n^b$ in terms of selectivity (especially when looking at the "balance of powers" between Grid$_r$ and UFS$_n$ in [2]) and clearly surpasses DFS$_n^b$_e (summary sizes are about the same compared to the quantized DFS$_n^b$ variants for GridMAR$_{64}^{6,4}$). Hence, K-D-MAR$_n^{b,k}$ and GridMAR$_r^{b,k}$ show that the computation of quantized MARs for rectangular data cells results in very selective summaries with low to moderate storage consumption. The good results of GridMAR$_r^{b,k}$ raise the question if the additional expenditure for adjusting the space partitioning to the global data distribution is justifiable, especially regarding DFS$_n^b$. K-D-MAR$_n^{b,k}$ should be the preferred choice when selectivity is very important and the adjustment is not costly, though. MBR-MAR^{b,k} (not using training data, as well) in some way substantiates this. Though selectivity is not quite as good as for GridMAR$_r^{b,k}$, it is achieved with very small average summary sizes. Furthermore, the DFS$_n^b$_e variant clearly gets surpassed concerning both selectivity and storage consumption.

Comparing the new approaches with techniques evaluated in [2] and [3], for K-D-MAR$_n^{b,k}$, it can be seen that encoding *several* quantized rectangles instead of just *one* is just about worthwhile (10% selectivity gain vs. 9.4% summary size growth when comparing K-D-MAR$_{2048}^{6,2}$ and K-D-MBR$_{2048}^6$). For the _e-variant, the improvement is bigger (23.2% selectivity gain vs. 13.4% summary size growth). Thus, it can be noted that, overall, K-D-MAR$_n^{b,k}$ is a slight improvement compared to the former state-of-the-art technique K-D-MBR$_n^b$. Though, the number of encoded rectangles per cell optimally remains small (2 or 3), since space partitioning by itself is already very decisive. For DFS$_n^b$, selectivity gains are tiny and not acceptable with regard to summary size growth in comparison to UFS$_n$. In contrast,

Table 1: Respective best parameterizations for the different evaluated novel approaches (including non-quantized DFS$_n^b$ for reasons of comparison; * stands for non compressed summaries)

Approach	**Selectivity**	**Space Efficiency** (**avg** min max) in byte		
DFS$_{8192}^6$ (quantized)	0.261%	**75.4**	50.9	691.8
DFS$_{8192}^6$_e (quantized)	0.576%	**68.0**	54.8	512.7
DFS$_{8192}^6$ (non-quant.)	0.256%	**93.4**	49.2	1,961.3
DFS$_{8192}^6$_e (non-quant.)	0.556%	**77.3**	53.8	1,388.9
GridMAR$_{64}^{6,4}$	0.355%	**72.9**	55.0	796.0
K-D-MAR$_{8192}^{6,3}$_e	0.179%	**84.4**	52.0	1,680.8
K-D-MAR$_{2048}^{6,2}$	0.161%	**79.6**	51.0	946.8
MBR-MAR^{8,15}	0.415%	**62.0***	31.0*	104.0*
—baseline—	0.138%	**265.8**	35.0	43,064
K-D-MBR$_{2048}^6$ (old)	0.179%	**72.7**	51.0	608.8
K-D-MBR$_{8192}^6$_e (old)	0.233%	**74.4**	52.0	797.6
UFS$_{8192}$ (old)	0.282%	**66.9**	48.0	467.4
UFS$_{8192}$_e (old)	0.682%	**60.4**	49.8	295.3
GridMBR$_{64}^6$ (old)	0.494%	**65.6**	55.0	329.0
MBRGrid$_{64}$ (old)	0.532%	**81.3**	53.0	414.0

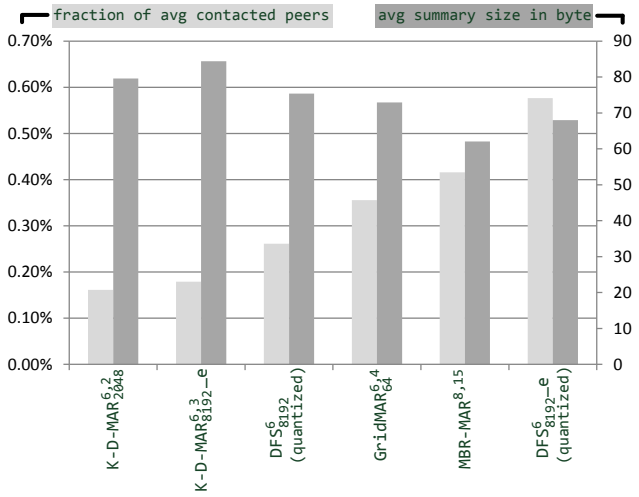

Figure 10: Novel hybrid techniques sorted by selectivity (respective left bar).

GridMAR$_r^{b,k}$ shows great selectivity improvement compared to GridMBR$_r^b$ at acceptable summary size growth, and MBR-MAR^{b,k} is an improvement in both target measures in relation to the comparable "old" techniques GridMBR$_r^b$ and MBRGrid$_r$.

4. RELATED WORK

The techniques for summarizing geospatial data discussed in this paper mostly stem from approaches or are combined from approaches known from spatial index structures (cf. [14] for an extensive overview over spatial index structures). Related index structures are the R-tree and its variants (like the R*-tree or the R$^+$-tree), the k-d-tree, Voronoi-diagram-based techniques or the LSD^h-tree. In general, the peer summaries are similar to the descriptions maintained in the inner nodes of tree-based index structures to represent subordinated trees. Hence, by utilization of appropriate strategies for example to determine subtrees for inserting data, the presented techniques could be suitable for an application in tree-based index structures, as well.

The DFS$_n^b$ and UFS$_n$ summaries utilize pruning techniques widely used in metric index structures to for example limit the number of distance computations. See [6] for the utilization of different pruning-criteria for metric index structures in a high-dimensional space or [14] for an overview on metric index structures in general.

Geographically focused crawling could also be a domain for utilizing compact resource descriptions [15]. If the geographic extent covered by a certain website or document archive can be indicated by a specific service, a crawler could estimate the potential usefulness of these resources with regard to its focused crawling task before actually harvesting the source.

5. CONCLUSION

This paper focuses on hybrid techniques which employ information quantization in order to reduce summary sizes. Four novel techniques are introduced. For approaches based on adapted space partitioning, the additional information encoded in the summaries in certain cases pays off in comparison to similar techniques without coded actual data regions. K-D-MAR$_n^{b,k}$ seems most promising in this category. The techniques not specifically adapting to the data distribution at hand (GridMAR$_r^{b,k}$, MBR-MAR^{b,k}) show remarkable improvements compared with similar previous approaches and even are opening the discussion, if the efforts for adapted space partitionings are justifiable.

6. REFERENCES

[1] F. Cuenca-Acuna, C. Peery, R. P. Martin, T. D. Nguyen. PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. *IEEE Intl. Symp. on High Performance Distributed Computing* 236–246, 2003

[2] S. Kufer, D. Blank, A. Henrich. Techniken der Ressourcenbeschreibung und -auswahl für das geographische Information Retrieval. *Proceedings of the IR Workshop at LWA 2012* 1–8, September 2012.

[3] S. Kufer, D. Blank, A. Henrich. Using Hybrid Techniques for Resource Description and Selection in the Context of Distributed Geographic Information Retrieval. *Advances in Spatial and Temporal Databases. Lecture Notes in Computer Science* 8098:330–347, August 2013.

[4] D. Blank, A. Henrich. Describing and Selecting Collections of Georeferenced Media Items in Peer-to-Peer Information Retrieval Systems. *Diaz, Laura ; Granell, Carlos ; Huerta, Joaquin (eds.): Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications* 1–20, 2012.

[5] B. Becker, P. G. Franciosa, S. Gschwind, T. Ohler, G. Thiemt, P. Widmayer. An Optimal Algorithm for Approximating a Set of Rectangles by Two Minimum Area Rectangles. *Intl. Workshop on Computational Geometry. LNCS* 553:22–29, 1991.

[6] D. Blank, A. Henrich. Resource Description and Selection for Range Query Processing in General Metric Spaces. *BTW 2013: 15. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web. GI-Fachbereich "Datenbanken und Informationssysteme"* 93–112, 2013.

[7] A. Henrich. The LSD^h-tree: An Access Structure for Feature Vectors. *Proc. of the 14th intl. conf. on Data Engineering* 262–369, 1998.

[8] B. Seeger, H.-P. Kriegel. The buddy-tree: an efficient and robust access method for spatial data base systems. *Proc. of th 13th intl. conf. on VLDB* 590–601, 1990.

[9] J. M. Keil Polygon Decomposition. *J.-R. Sack, J. Urrutia (Eds.), Handbook of Computational Geometry* 491–518, 1999.

[10] D. S. Franzblau, D. J. Kleitman. An algorithm for covering polygons with rectangles. *Inf. Control* 63(3):164–189, 1986

[11] V. Soltan, A. Gorpinevich. Minimum dissection of rectilinear polygon with arbitrary holes into rectangles. *SCG '92 Proceedings of the eighth annual symposium on Computational geometry* 296–302, 1992.

[12] J. L. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* 18(9):509–517, 1975.

[13] V. Murdock. Your Mileage May Vary: On the Limits of Social Media. *SIGSPATIAL Special* 3(2):62–66, July 2011.

[14] H. Samet. Foundations of Multidimensional and Metric Data Structures. *Morgan Kaufmann Publishers Inc., San Francisco, CA, USA* 2005

[15] D. Ahlers, S. Boll. Adaptive geospatially focused crawling. *CIKM '09 Proceedings of the 18th ACM conference on Information and knowledge management* 445–454, 2009.