Using Hybrid Techniques for Resource Description and Selection in the Context of Distributed Geographic Information Retrieval

Stefan Kufer, Daniel Blank, and Andreas Henrich

University of Bamberg, D-96047, Germany
{stefan.kufer,daniel.blank,andreas.henrich}@uni-bamberg.de
http://www.uni-bamberg.de/minf/

Abstract. The amount of media items on the web is increasing tremendously, especially regarding personal media items. To effectively collaborate over and share these massive amounts of media objects, there is a strong need for adequate indexing and search techniques. Trends like social networks, large-storage mobile devices and high-bandwidth networks make peer-to-peer (P2P) information retrieval systems of deep interest.

Hence, resource selection based on compact resource descriptions is used to efficiently determine promising peers w.r.t. a query. To design effective media search applications, multiple search criteria need to be addressed. Subsequently, besides text or visual media content, geospatial data is frequently used.

We propose techniques to summarize and select collections of georeferenced media items in P2P systems. Generally, these summarization techniques can be divided into geometric and space partitioning approaches. This paper presents and evaluates techniques of a third category, hybrid approaches that combine features of geometric and space partitioning techniques.

1 Introduction

During the last years, the amount of (especially personal) media data accessible via the World Wide Web has vastly increased. People write blogs, twitter about events or their lifes, use remote photo or video communities and share media content in social networks. Therefore, people not only store these personal media objects, but also interact with each other through them, for example by collaboratively tagging or commenting on various items. Consequently, online resources varying in size, media type and update frequency have to be administered [1].

Additionally, the availability—and with it the usefulness—of geospatial metadata has increased dramatically in recent times. Nowadays, digital cameras as well as mobile phones are often equipped with GPS sensors at reasonable costs. Hence, these devices are able to capture georeferenced information to enrich media data in many situations, like shooting videos or taking pictures. Supplementary, geo-tagging tools with rich user interfaces have emerged in several domains and there are large geo-tagging initiatives attempting to georeference

textual resources such as Wikipedia. Taken en masse, the increased importance of geospatial information in the context of searches can be recognized.

Obviously, geospatial information is not the only search criterion. Other criteria such as textual content, timestamps and (low-level) audio and visual content information can be used when searching for media items as well. An integrative combination of these criteria with spatial filter or ranking conditions can facilitate an effective retrieval of text, image, audio and video documents.

Our search scenario assumes a P2P system maintaining personal media archives. P2P systems are formed by computers (potentially) distributed all over the world, the peers, which can act as both clients and servers. By applying a scalable P2P IR protocol, a service of equals for the administration of media items can be established, without the requirement to maintain expensive infrastructure. In our scenario, the media items administered in a personal archive are stored locally on the peer (that is to the user's personal device), without the need to store media items on remote servers hosted by service providers such as Flickr or YouTube, reducing the dependency on service providers as informational gatekeepers. To facilitate retrieval, media items can be described by four criteria: 1) textual content, 2) low-level content features, 3) timestamps and 4) a geographic footprint. Hence, personal media archives can be represented by four corresponding resource descriptions (summaries). Each summary represents a feature aggregation in terms of the media items a resource (peer) maintains, for example an aggregation over all the geographic coordinates of all the media items a peer administers. As a scalable P2P protocol, Rumorama [2] is applied in our scenario, and establishes hierarchies of PlanetP-like [3] networks. In a PlanetP-like network, every peer knows all the resource descriptions of every other peer inside the network, enabling routing decisions while query processing (that is contacting the most promising peers, according to the summaries and with respect to a certain query, first). The distribution of resource descriptions in the network is assured by randomized rumor spreading [3].

The present paper studies novel techniques with respect to resource description and resource selection considering geographic metadata and thus continues work presented in [4]. There we examined techniques falling into either the category of geometrical approaches or the category of space partitioning approaches (cf. Sect. 2). The current paper introduces a third category, hybrid approaches combining features of the two previous approaches, and evaluates them with respect to [4].

2 Resource Description and Selection for Geographic Queries

Generally, in our scenario every peer maintains a chunk of images as media items, where every image is described by a single pair of lat/long-coordinates. These geocoordinates are basically treated as point data in a plane for resource description and selection. Consequently, distances are approximated using the Euclidean distance, since investigations in [5] showed that the usage of distance

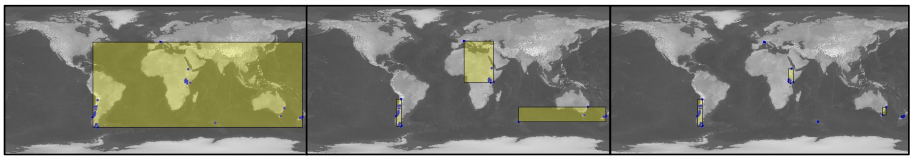

Fig. 1. Visualization of the simple MBR approach (on the left) and the RecMAR$_k$ technique with k=3 (on the middle) respectively k=6 (on the right) as examples for geometric approaches. The blue points denote the data points of the sample peer.

measures better suited for distance calculations between two points on earth (like Haversine distance [6] or Vincenty distance [7]) do not conduct noticeable changes in case of our data collection (cf. Sect. 3.1).

Resource selection is performed by ranking peers based on their resource descriptions, the query location and maybe some additional information such as reference points. The peer ranking defines the order in which peers get contacted during query processing. When searching for the k closest images with respect to a query location, the peer ranking should reflect that peers with a higher probability of administering a bigger fraction of the top-k images receive a higher rank than peers maintaining a smaller fraction of the top-k images. Our scenario requires a reasonable trade-off between the expressiveness of a resource description (allowing better selectivity) and its storage requirements for its representation. For the techniques presented in the following, differences in evaluation time are negligible compared to the time needed for accessing peers and will therefore not be considered.

The remainder of this section shortly presents the two classes of summary types evaluated in [4] and also describes the two most promising approaches found in these studies (cf. Sect. 2.1 and Sect. 2.2). Next, the class of hybrid approaches and the specific techniques examined in this paper (cf. Sect. 2.3) get introduced. Finally, the resource ranking algorithms will be described in Sect. 2.4.

2.1 Geometric Approaches

The first class of summary types computes a single or multiple geometrical shapes to enclose the set of point data a peer administers. Calculating approximated, concise representations of complex forms (a peer's point cloud in our case) is a standard problem in many computer science domains [8], therefore plenty of computational geometry algorithms exist and are applicable for this category of summaries. Figure 1 (on the left) shows one of the most basic techniques in this field, as it simply encloses all of a peer's data points into a minimum bounding rectangle (MBR) to describe its data, requiring two pairs of lat/long-coordinates to be stored.

In [4], we found the most promising technique of this class to be an approach where a peer's point cloud is described by several so-called minimum area

rectangles (MARs). The computation algorithm is based on work from Becker et. al. [8] to summarize a set of bounding boxes by two bounding boxes which minimize the area that is covered. This algorithm has been adjusted to point data and was transformed into a recursive version, continuing the disassembly until a predefined maximum of k MARs has been computed or a certain threshold *dist* has been undercut, which is taking the distance between the center of a MAR and the most distant of its associated data points into account. The threshold needs to be adjusted with respect to the underlying data collection to achieve appealing results. This technique, denoted RecMAR$_k$, shows excellent selectivity (that is the fraction of peers contacted to retrieve the relevant images) while keeping summary sizes at a reasonable level, since the algorithm allocates more MARs for peers with "complex" point data and less MARs for peers with less "complex", spatially narrowed point clouds. See Fig. 1 on the middle and on the right for a visualization of RecMAR$_k$.

2.2 Space Partitioning Approaches

The second category of summary types globally segments the data space into a certain number of subspaces. Thus, the segmentation is the same for all peers. This allows storing the information if a peer maintains data points for a certain subspace (or not) in the very peer's summary. Basically, there are three approaches to store information for a subspace: storing how many data points are contained in a cell (using integer values), storing whether at least one data point is contained in a subspace or not (using bits), or storing some kind of distance information concerning a peer's data points in a subspace (for example, the maximum distance between any data point being located in a certain subspace and the subspace's center, using float values). Figure 2 on the left shows the simplest space partitioning technique, mapping the lat/long-coordinates to a uniform grid (which results in non-uniform grid cell sizes on the sphere). The yellow highlighted cells contain data points, the related information can be stored in any of the aforementioned ways in a peer's summary.

Our experiments in [4] reveal the so-called Ultra Fine-grained Summaries (UFS$_n$) to be best when taking both description selectivity and summary sizes into account. The data space gets segmented based on n predetermined reference points invoking a Voronoi-like space partitioning (see Fig. 2 on the right). Therefore, a data point is assigned to the cell of the reference point being closest to it. To attain convincing results, we found data space segmentation has to be adjusted with respect to the underlying data collection (yielding smaller subspaces for areas with high global point density and bigger subspaces for low density areas). This can be achieved by randomly choosing reference points out of the underlying collection, or (simulating cases where this is not possible) from an external source where data points are similarly distributed as in the data collection. Using UFS$_n$, for each peer and each cell there is only binary information (1 or 0) stored depending on whether there is at least one data point inside a certain cell or not, allowing compression techniques to greatly reduce summary

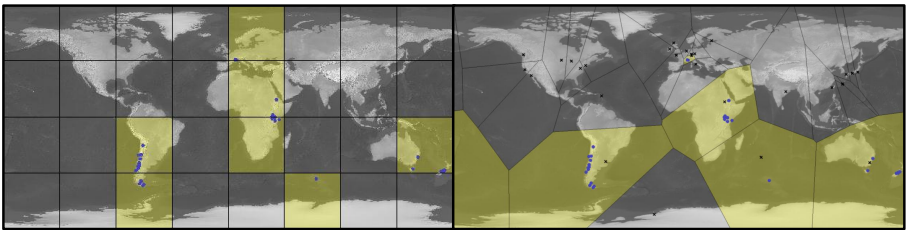

Fig. 2. Visualization of the simple grid (on the left) and the UFS_n technique (on the right), invoking a Voronoi-diagram-like space partitioning with 32 subspaces, as examples for space partitioning approaches. The black crosses on the right denote the reference points of the Voronoi diagram.

sizes. We compress the summaries using Java's standard gzip implementation[1]. To gain selectivity, the number of data cells can be increased. In [4], this approach results in very selective resource descriptions while keeping the average description size at a very low level.

2.3 Hybrid Approaches

This subsection introduces a third category of summary types, the hybrid approaches. These techniques combine characteristics of both geometric and space partitioning approaches, using geometric shapes as well as space segmenting to describe the location(s) of a peer's data. Basically, this category can be further distinguished into two subclasses, whether the techniques are using the geometric approach as first data description tool and affiliate some space partitioning method, or if they are doing it the other way round—some space segmentation followed by the usage of geometric shapes—. The first two techniques presented in the following are using geometric shapes as primary and space partitioning as secondary technique, while the subsequent two approaches are utilizing space partitioning primary and geometric shapes secondary.

$MBRGrid_r$. The first technique combines the basic geometric and space segmentation approaches presented earlier in the present paper. Initially, an MBR containing all of a peer's data points is computed. In a second step, the enclosed area gets segmented into subspaces by utilizing a uniform grid to partition the corresponding space. The number of subspaces can be adjusted by a parameter r, representing the number of rows on the grid. For this local segmentation, we did likewise as in [4] or [9] for global space segmentation, setting the number of

[1] The summaries of the hybrid summary types have been compressed the same way. For $RecMAR_k$, compression resulted in higher average summary sizes, therefore for this technique uncompressed summary sizes are taken into account when comparisons between the different techniques are drawn.

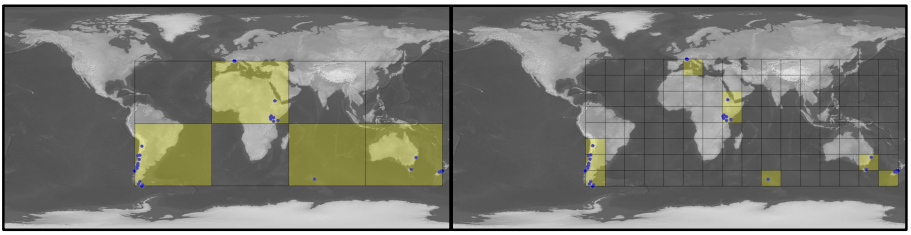

Fig. 3. Visualization of the MBRGrid$_r$ summaries with r=2 (on the left), respectively r=8 (on the right) for the interior grid

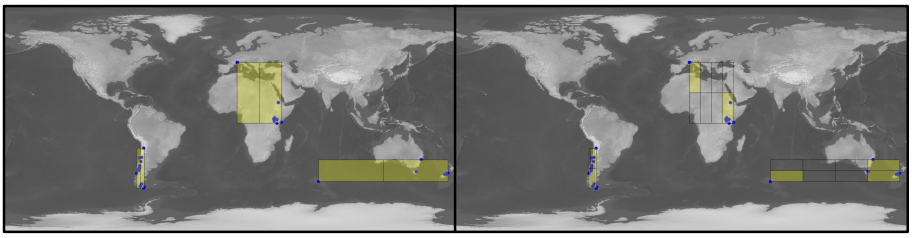

Fig. 4. Visualization of the KMARGrid$_k^r$ summaries with k=3 (computation of up to three enclosing rectangles) and r=1 (on the left), respectively r=2 (on the right) for the particular interior grids

columns twice as big as the number of rows. Adjusting the number of rows and columns according to, for example, the height/width-ratio of the enclosing MBR is conceivable and could be part of future work. See Fig. 3 for a visualization of an MBRGrid$_r$ summary.

A peer's summary is represented by a bit vector. First, the values encoding the two lat/long-coordinate pairs of the enclosing MBR are incorporated. Originally, the MBR bounds are captured as float values and get converted into binary information for summary inclusion ($4 \cdot 32$ bits). The summary's remainder consists of values 1 or 0, indicating whether the corresponding subspace contains at least one data point or not. If all of a peer's data points share (exactly) the same lat/long-coordinates, only the values specifying the MBR are encoded.

KMARGrid$_k^r$. Unsurprisingly, KMARGrid$_k^r$ takes the RecMAR$_k$ algorithm as a starting point to compute up to k MARs containing all of a peer's data points. The second step is similar to MBRGrid$_r$, except that in *each* MAR there is a Grid to be invoked. Again, grid granularity is determined by a parameter r, yielding r rows and $2 \cdot r$ columns for a MAR's grid. See Fig. 4 for a visualization of a KMARGrid$_k^r$ summary.

A bit vector represents a peer's summary as well. The grid-divided MARs are encoded one after another, with each using $4 \cdot 32$ bits for the rectangle extents and $r \cdot 2r$ bits to indicate cell occupancy for the respective interior grid.

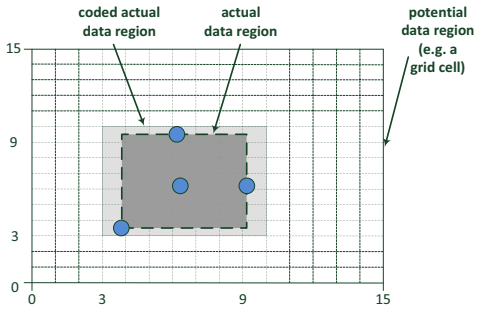

Fig. 5. Coding of an actual data region. The blue circles are the data points to be described (adapted from [14]).

GridMBR$_r^b$. GridMBR$_r^b$ is the first of two approaches doing it the alternative way, that is first segmenting the (whole) data space and afterwards using geometric shapes in a second step. GridMBR$_r^b$ initially segments the data space by imposing a regular grid onto the data space similar to the simple partitioning approach presented earlier. In the second step, for each occupied cell, an approximated MBR containing all the cell's data points is computed, pursuing an idea presented in [14].

Here a distinction between a potential data region (being the cell of a spatial access structure) and an actual data region is made, the latter being an MBR containing all the data points being located in the potential data region (that is a certain grid cell in our case). In [14], to reduce storage spent on encoding these interior MBRs, a technique originally introduced with the buddy-tree [15] is used, exploiting the presence of potential data regions. For encoding a MBR, generally four values need to be stored, specifying the lower left and upper right corner. Let's say b bits shall be used to encode one of these values. With this, we can distinguish 2^b different positions on an axis of a data cell. These positions can be used to encode an approximated MBR (also called the coded actual data region), being larger than the true MBR (the actual data region), but requiring significantly less storage than encoding the true MBR with float values. Figure 5 illustrates this for two-dimensional data with b=4.

For GridMBR$_r^b$, the parameter b is used to determine how much storage shall be used to encode one of the four required MBR bounding values, using b bits to encode a value (for example if b=3, eight different positions on each cell axis can be distinguished). Likewise as for example MBRGrid$_r$, a parameter r specifies the number of rows (r) and the number of columns ($2 \cdot r$) of the global grid.

As a peer's summary, a bit vector is used. Grid cells which do not contain any data point are encoded with 0, cells that contain at least one data point are encoded with 1. After an 1 representing an occupied cell, there follow $4 \cdot b$ bits encoding the four required values specifying lower left and upper right of the interior MBR. See Fig. 6 for an illustration of GridMBR$_r^b$ summaries for the sample peer.

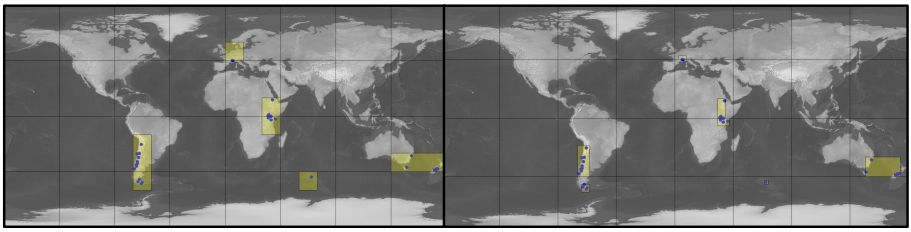

Fig. 6. Visualization of the GridMBR$_r^b$ summaries with r=4 (resulting in 32 grid cells) and b=2 (on the left) respectively b=4 (on the right), resulting in four respectively 16 possibilities on each axis to encode the interior MBRs

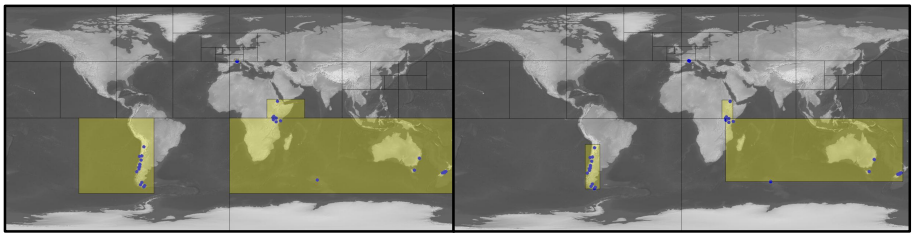

Fig. 7. Visualization of the non-uniform data space partitioning of K-D-MBR$_n^b$ with n=32 subspaces and b=2 (on the left) respectively b=4 (on the right) for encoding the cell-interior MBRs

K-D-MBR$_n^b$. The last hybrid approach takes the k-d-tree-like data space partitioning, called GFBu$_n$ in [4], as starting point. Using this technique, the data space gets segmented into rectangular cells of different sizes. Training data is used to learn a segmentation adjusted to the underlying data collection, while sources for training data are the same as for the reference points of the UFS$_n$ summaries, that is directly out of the data collection or from an external source with similar point distribution. At the beginning of the training process, the data space consists of only one cell or bucket. Training data points are sequentially inserted into this bucket, until a bucket-overflow occurs followed by splitting the bucket into two parts. Afterwards, further data points are inserted into the data structure. The whole process is repeated until the desired amount of n buckets has been reached.

As a second step after space partitioning, an approximated MBR is encoded for each cell containing at least one data point. The MBR computation is accomplished the same way as for GridMBR$_r^b$, that is there is a parameter b again, to adjust the amount of storage used to encode the MBR. Figure 7 illustrates this for the sample peer. For the summaries, bit vectors are used in the same way as for GridMBR$_r^b$.

2.4 Ranking

Ranking of peers is conducted based on the information supplied by the summaries. For the hybrid approaches, all the ranking algorithms operate on the same principle. Note that all these approaches at the very end encode information of rectangular areas in which a peer's data points are located.

To rank peers, for each peer the algorithm extracts the peer's summary information to construct all the rectangles containing the peer's data points. This is 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 results in a distance of 0 for this rectangle). For each rectangle, distance information and the area covered by the rectangle are stored in a so-called R-Entry. All R-Entries of a peer are arranged in a queue, sorted by distance in ascending order. If the distance of two R-Entries is the same, the one with the smaller area covered is favored.

To determine a ranking between two peers, the sorted R-Entries are 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 covers a smaller area than p_b's R-Entry and vice versa. If the area covered is the same for both R-Entries, the next entries from the queues are compared, etc. (until a decision can be made)[2]. If the R-Entry comparison does not lead to a decision, a random ranking choice is made.

For RecMAR$_k$ evaluated in [4], ranking works the exact same way. The UFS$_n$ ranking shows a little variation due to the computational complexity for calculating Voronoi cell boarders. There, the reference points c_j $(j \in \{0; n-1\})$ are sorted in ascending order with respect to the distance to the query location. The first element of the sorted list L corresponds to the cluster center being closest to the query (called query cluster). If peer p_a administers documents in this query cluster (that is 1 is set for the cluster in the peer's summary) while peer p_b does not (that is 0 is set), 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.

3 Evaluation

In this section, we will give an intital brief description of the data collection used for the evaluation (cf. Sect. 3.1). For comparability with previous evaluated approaches, we use the experimental setting conducted in [4], but only apply

[2] Obviously, not all the peers hold the same number of rectangular areas containing data points. In this case, for the peer represented by fewer rectangular areas, dummy entries are generated, whose values have the most unfavorable impact (that is infinite distance and infinite area) on the ranking for the respective peer.

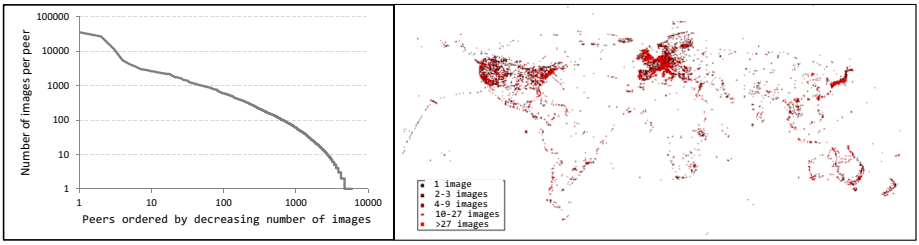

Fig. 8. Number of images per peer (left) and geographic distribution of images locations (right) for the evaluated data collection

one query mode (cf. Sect. 3.2), since results are very similar for different query point sources in [4]. Afterwards, the experimental results will be displayed and discussed in Sect. 3.3.

3.1 Data Collection

During 2007, a large amount of publicly available, georeferenced images uploaded to Flickr (http://www.flickr.com) was crawled. In our scenario, every Flickr user operates a peer of its own. Thus, we assign images to peers by means of the Flickr user ID. After some data cleansing, the collection consisted of 406,450 geo-tagged images from 5,951 different users/peers. The distribution of the number of images per peer (see Fig. 8) is very skewed which is typical for many P2P settings [3]. Few peers administer large amounts of the collection ("big peers"), while there are also many peers which store only few images ("small peers"). See [9] for a more detailed analysis. On the right, Fig. 8 illustrates the geographic distribution of the image locations, revealing an uneven spread with image hotspots in North America, Europe and Japan.

3.2 Experimental Setting

We use 500 image locations chosen in a two-step process as queries. First a random peer is selected, second we choose a random geo-location from this peer. This ensures the same likelihood that a query originates from an arbitrary peer regardless of its size, thus not favoring big peers in the ranking.

For parameterization, k for RecMAR$_k$ and KMARGrid^r_k is set to 3, 6 and 9. For space partitioning approaches and hybrid techniques relying primarily on space partitioning, we choose the number of subspaces to be 512, 2,048 and 8,192, resulting in r being set to 16, 32 and 64 for MBRGrid$_r$ and GridMBR^b_r. For interior grids (used by MBRGrid$_r$ and KMARGrid^r_k), r is set to 8, 16, 32 and 64. To encode approximated interior MBRs for GridMBR^b_r and K-D-MBR^b_n, we vary b from 2 to 4 to 6, corresponding to the bits used to encode one of the four required MBR values, respectively. These parameters are generally applicable to our or similarly distributed data collections. For significantly differing data sets, the parameters will have to be reconsidered.

To adjust space partitioning to our data collection for UFS_n and K-D-MBR_n^b, we use the same two strategies as in [4]. The first strategy is to choose reference points, respectively training data from the underlying data collection at random. The second strategy is to select the data from the Geonames gazetteer (http://www.geonames.org), employing Gross Domestic Product (GDP) per country statistics from Worldmapper (http://www.worldmapper.org) to approximate the data distribution of our collection (see [9] for details why we choose this approach)[3]. Since the space partitioning is affected by the randomly chosen training data, we run ten experiments with different seeds for the random number generators to minimize the effects of outliers.

For RecMAR_k and KMARGrid_k^r, we use the same *dist* value (cf. Sect. 2.1) deployed in [4] for RecMAR Q_k, meaning the 0.75-quantile of the top-50 data point distances determined for 500 queries.

Space efficiency of different resource descriptions is measured by analyzing summary sizes (cf. Sect. 3.3). Remember we apply Java's gzip implementation with default parameters if summary compression is beneficial (which is the case for all summary types except RecMAR_k). The measurements include 27 byte serialization overhead necessary in order to distribute the resource descriptions within the network. To assess the selectivity of different approaches, we calculate the fraction of peers that needs to be contacted on average in order to retrieve the 50 image locations closest with respect to a given lat/long-pair as query location. Summary sizes are strongly dependent on the techniques used, so it is not possible to report the selectivity for specific given summary sizes.

To determine the 50 nearest neighbors, a k-nearest neighbors (kNN) algorithm, implemented as a range query with decreasing query radius, is used. First, all peers are ranked according to the ranking algorithms (cf. Sect. 2.4). For each of the ten best ranked peers, the 50 image locations closest to the query location are requested. Out of this set, the 50 closest image locations are determined to form the current intermediate top 50 results[4]. Afterwards, the already considered peers are removed from the set of peers yet to look at. The distance of the fiftieth closest image location is the query radius for the next round, in which ten further peers will be contacted. Peers which can be pruned from search, due to their resource description, the query location and the query radius, are removed from the set of peers still to consider. A renewed ranking is not conducted, meaning the set of peers is only ranked once in the first query round. In each round, the ten best ranked peers of the remaining set are enquired for their 50 closest image locations according to the query location, possibly leading to a substitution of (some of) the current top 50's images. Afterwards, these peers

[3] For differentiation, we expand e to the respective technique acronym if data was chosen from the Geonames gazetteer as external source, for example UFS_n e.

[4] The consideration of ten peers at once is done to exploit the parallelism of our scenario. Nevertheless, the determination of the top 50 image location's retrieval time (meaning the fraction of peers which had to be visited in order to retrieve the top 50 image locations) is captured on a single peer basis, meaning the contacting position of the appropriate peer is captured on a single step base and not on a ten step base.

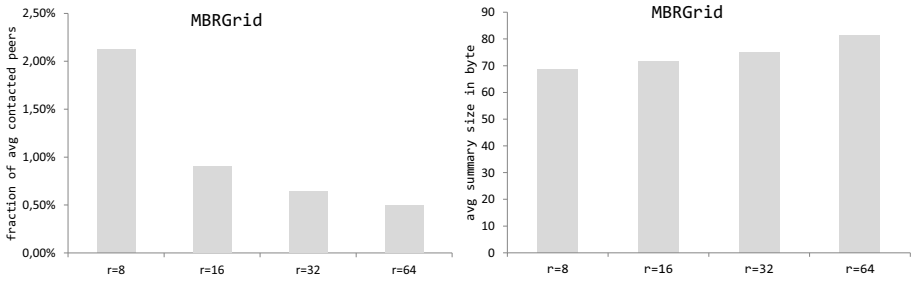

Fig. 9. Development of selectivity (on the left) and summary sizes (on the right) when varying parameter r for MBRGrid$_r$

are removed from the set of peers to consider. This procedure continues until the set of peers to consider is empty, meaning the 50 nearest neighbors with respect to the query location have been determined.

3.3 Experimental Results

In this section, we will evaluate the different techniques. For our 500 queries, an average of 8.234 peers maintain relevant image loactions, resulting in a fraction of 0.138% of the 5,951 peers. As a naive baseline it is interesting to note, that if all peers would directly transfer a (zipped) byte array containing all of their data points, average summary sizes would be 265.85 byte (cf. Table 3.3 at the end of this chapter).

In Fig. 9, retrieval performance and summary sizes are depicted for MBRGrid$_r$ with varying parameter r. There is a degressive improvement in selectivity with the increase of r, as the areas containing a peer's data points are described more precisely. At the same time, compression keeps summary size growth moderate. If r is altered from 8 to 64, selectivity is more than four times better while summaries are about 19% bigger on average. Also, selectivity growth is still disproportionate to summary size growth when increasing r from 32 to 64, making MBRGrid$_{64}$ the best choice when pondering between the two superordinate goals.

For KMARGrid$_k^r$, selectivity can obviously be enhanced by both increasing either one or both parameters k and r. Figure 10 on the left shows a degressive improvement as greater values for k and r are chosen. Combining geometric shapes with space partitioning seems very beneficial with respect to selectivity, since KMARGrid$_3^8$ is almost as good as RecMAR$_9$ (cf. Table 3.3), though it has to be admitted that the former requires about 13 byte extra storage on average. Generally, it can be seen that a decomposition into more MARs yields greater benefits for selectivity than a finer resolution of the interior grids. The utmost gains are attained by increasing k from 3 to 6. Afterwards, a further increase of k from 6 to 9 is not as beneficial, as our stopping criterion for further MAR decomposition comes into play, resulting in few peers to be described by the

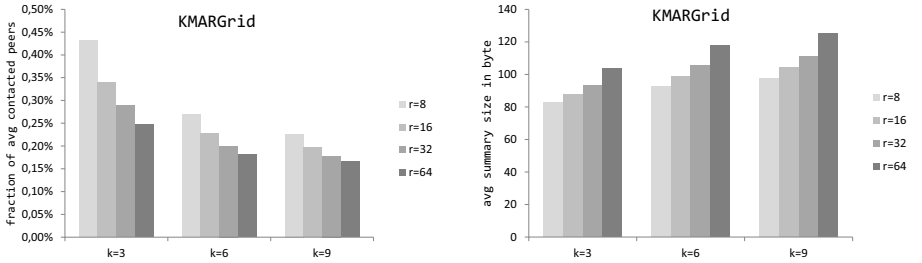

Fig. 10. Development of selectivity (on the left) and summary sizes (on the right) when varying parameters k and r for KMARGrid$_k^r$

maximum of nine or a matching amount of MARs. On the other hand, altering k from 6 to 9 leads to a rather moderate average summary size growth. The required average storage space is depicted in Fig. 10 on the right. Increasing k results in degressive summary size growth, while increasing r results in progressive summary size growth. When searching for the best compromise between selectivity and average summary sizes, we compare percentual selectivity gains and percentual average summary size growth when increasing our parameters k and r[5]. As long as percentual selectivity gains are higher than percentual summary size growth, we say it is beneficial to raise the parameters. Taking this into account, KMARGrid$_9^{32}$ results as best parameterization for this technique. Generally, at the beginning of the parameter rise (that is from k=3 and r=8 on), selectivity gains are vigorously disproportionate compared to average summary size growth, flattening during the further procedure until it is only slightly disproportionate or even slightly underproportionate.

The results for GridMBR$_r^b$ are shown in Fig. 11. A relatively precise encoding of interior MBRs in coarse grids results in high selectivity gains in comparison to increasing the number of subspaces. Taking for example GridMBR$_{32}^2$ as a basis and increasing parameters r, respectively b to the next level, GridMBR$_{32}^4$ shows a better selectivity compared to GridMBR$_{64}^2$, since GridMBR$_{32}^4$ only contacts a peer fraction of 0.94% while GridMBR$_{64}^2$ contacts 1.19% of the peers. At the same time, increasing b also results in more space efficient summary sizes (for example 58.2 byte for GridMBR$_{32}^4$ vs. 60.7 byte for GridMBR$_{64}^2$). Generally, it is beneficial to increase both r and b since the selectivity gain is still heavily disproportionate to summary size growth when comparing for example GridMBR$_{64}^6$ to both GridMBR$_{32}^6$ and GridMBR$_{64}^4$.

Results for both variants of K-D-MBR$_n^b$ are depicted in Fig. 12. Generally, selectivity increases most when raising b from 2 to 4, but also raising n from 512 to 2048 yields vast selectivity gains, both clearly disproportionate compared to summary size growth. A further parameter increment (b=6 and n=8,192) results in only slightly disproportionate or even underproportionate selectivity

[5] Obviously, the importance of selectivity and storage requirements has to be weighted for a concrete application scenario.

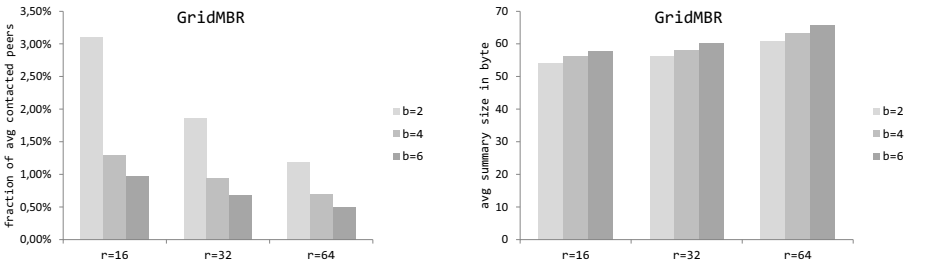

Fig. 11. Development of selectivity (on the left) and summary sizes (on the right) when varying parameters r and b for GridMBR$_r^b$

gains. Overall, it is very promising to encode interior MBRs for space partitioning with uneven cell sizes, as the larger subspaces often contain data points in only a narrow area. Thus, the data point occurrence areas can be described much more precisely. Likewise for MBRGrid$_r^b$, increasing b is more beneficial than increasing the number of subspaces (parameter n here). Generally, when comparing K-D-MBR$_n^b$ to K-D-MBR$_n^b$ e, the same behavior observed for UFS$_n$ and UFS$_n$ e in [4] can be noted. The e-variant shows worse selectivity but lower required storage space, since space partitioning is less fitted to the underlying data collection, resulting in less subspaces to be occupied with a peer's data points on average. This reduces the amount of interior MBRs which need to be encoded. Overall, selectivity gains area slightly better for the e-variant when increasing n and b, as the inferior base accuracy causes a higher gain potential. When comparing percentual selectivity gain with percentual summary size growth, different parameterizations are best for the respective variants. For training data right from the underlying data collection, K-D-MBR$_{2048}^6$ emerges as best compromise, while for training data from an external source, K-D-MBR$_{8192}^6$ e arises as most reasonable solution. It is worth mentioning that K-D-MBR$_{2048}^6$ outperforms K-D-MBR$_{8192}^6$ e both with respect to selectivity and summary size.

Comparing the different techniques on their respective best parameterization (cf. Fig. 12 and Table 3.3), both MBRGrid$_{64}$ and GridMBR$_{64}^6$ are significantly worse with respect to selectivity in comparison to the more complex hybrid approaches (and RecMAR$_9$ and UFS$_{8192}$ as well), even though for both the most precise examined parameterization has been chosen (cf. Fig. 13). Only UFS$_{8192}$ e with its just approximately fitted space partitioning is outperformed by MBRGrid$_{64}$ and GridMBR$_{64}^6$. However, GridMBR$_{64}^6$ requires significantly less storage than MBRGrid$_{64}$ and less storage than any other technique except UFS$_{8192}$ e, which is clearly outperformed by GridMBR$_{64}^6$ with respect to selectivity. Thus, GridMBR$_r^b$ could be worthwhile if somehow there is no possibility to adjust techniques with respect to the underlying data collection (which all of the other techniques except MBRGrid$_r$ to some extent do, may it be the use of stopping criteria or adapted space partitioning), as selectivity is still neat with very small average summary sizes.

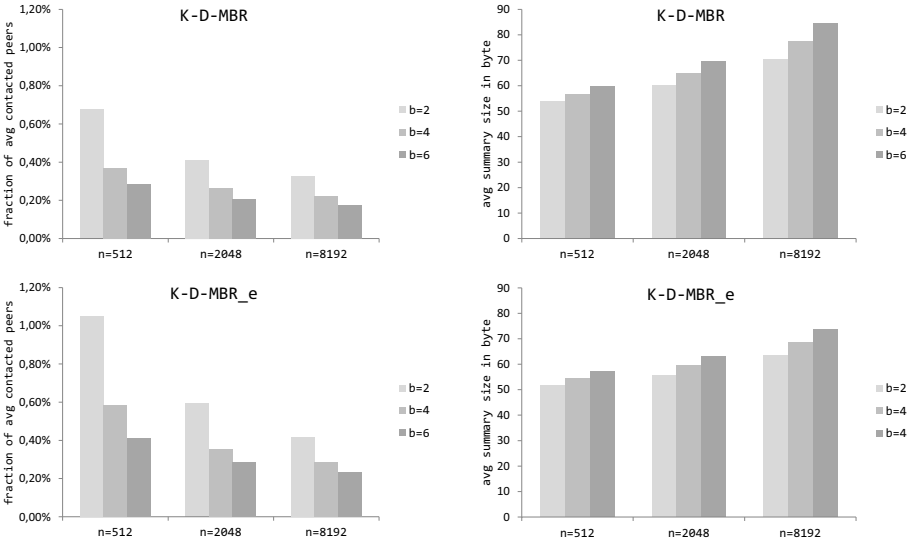

Fig. 12. Development of selectivity (on the left) and summary sizes (on the right) when varying parameters n and b for K-D-MBR^b_n(e)

The K-D-MBR$^6_{2048}$(e)-variants both significantly outperform their respective UFS$_{8192}$(e)-variants in terms of selectivity, despite a four times lower amount of subspaces, due to their usage of cell-interior MBRs. This is achieved by only slightly bigger average summary sizes. Even more surprisingly, K-D-MBR$^4_{2048}$ (not shown in Fig. 13) can outperform UFS$_{8192}$ both on selectivity *and* summary sizes, making K-D-MBR^b_n the overall superior technique compared to UFS$_n$. In terms of selectivity, K-D-MBR$^6_{2048}$ is yet slightly topped by KMARGrid$^{32}_9$, which is closing up to 0.04% with respect to the theoretical optimum (cf. Table 3.3). On the other hand, average summary sizes are almost 60% greater compared to K-D-MBR$^6_{2048}$. Generally, it can be seen that for the hybrid techniques, segmenting space first and computing geometric shapes second, results in far smaller summary sizes while selectivity is about equal compared to the alternative way (when matching GridMBR^b_r to MBRGrid$_r$ and K-D-MBR^b_n to KMARGrid^r_k).

Taking both selectivity and average summary sizes into account, K-D-MBR^b_n with its training data directly chosen from the underlying data collection seems best as it offers almost prime selectivity combined with still very small average summary sizes, clearly outperforming the former state-of-the-art approaches UFS$_n$ and RecMAR$_k$. Furthermore, K-D-MBR^b_n is much more insensitive in case training data origins from an external source than UFS$_n$. Considering pure selectivity, KMARGrid^b_k constitutes an advance as well, though at the cost of big summary sizes.

Interestingly, selectivity is not directly related to the data space area spanned on average by the different summarization techniques (cf. Table 3.3). This can be seen as RecMAR$_9$ and both K-D-MBR$^6_{2048}$ and K-D-MBR$^6_{8192}$ offer much better

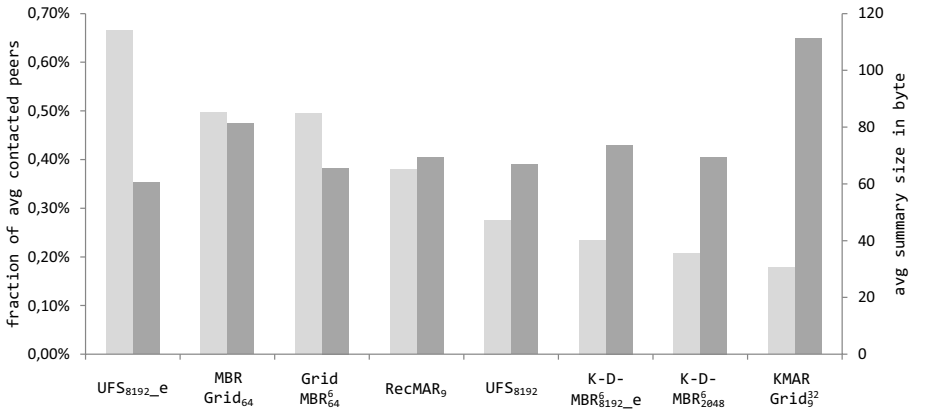

Fig. 13. Overview of evaluated techniques sorted by selectivity (light gray bars) and additionally showing average summary sizes (dark gray bars)

Table 1. Result table for our experiments (summary size values S_x in bytes), uncompressed summary size values are marked with *

Approach	Ø frac. of peers in %	$S_\emptyset$	S_{min}	S_{max}	Ø area covered
UFS_{8192} e	0.665	60.5	49.8	295.3	not calculated
$MBRGrid_{64}$	0.498	81.3	53	414	0.984
$GridMBR^6_{64}$	0.495	65.6	55	329	0.630
$RecMAR_9$	0.386	69.3*	43*	171*	2.933
UFS_{8192}	0.276	66.88	48	467.4	not calculated
K-D-MBR^6_{8192} e	0.234	73.7	48.1	947.5	0.900
K-D-MBR^6_{2048}	0.208	69.5	46.45	577.5	1.015
$KMARGrid^{32}_9$	0.178	111.2	53	994	0.024
Baselines	0.138	265.8	53	43064	–

selectivity, while the area on average overlaid by the respective descriptions is (in case of $RecMAR_9$ clearly) larger compared to $MBRGrid_{64}$ and $GridMBR^6_{64}$. It seems that in areas with low point density, taller delineated descriptions are acceptable if in high density areas the descriptions are subsequently more precise.

4 Related Work

This paper introduced and evaluated techniques to summarize geospatial data. The techniques presented predominantly orientate themselves towards approaches known from spatial index structures (cf. [10]), like for example the R-tree [11], the k-d tree [12], Voronoi-diagram-based techniques (for example [13]) or the LSD^h-tree [14].

At the very end our hybrid techniques are all based on describing rectangular areas containing data points. In the context of spatial index structures, other

hybrid techniques have been proposed, ultimately describing much more irregular areas, like [16] depicting how to combine MBRs with Voronoi-diagram-based space partitioning.

Our summary based P2P scenario constitutes the general frame to evaluate different geospatial summary techniques against one another. Alternatively, especially in the context of two-dimensional geo-data, it could be compelling to employ other P2P systems. Structured P2P systems might be suitable. There every peer is responsible for a certain subspace. New data to be administered within the P2P network is transferred in compliance with these responsibilities, reducing the autonomy of the peers [17]. At the same time, query processing can be achieved with logarithmic costs by using structured approaches [17]. An extensive overview of P2P technologies is given in [18].

Within the presented scenario, queries related to geographic data were processed independently from other summary types. It could be worthwhile to not only consider summary techniques in an isolated way, but also in cooperation. Effective searches of, for example, an image showing a sunset at the Grand Canyon, could be achieved this way. Currently, summaries for both—low level visual content and geographic information—would lead to two independent resource rankings. Both rankings would have to be combined into one with appropriate techniques (for example [19]). In [20], an approach aggregating several summary types into one summary to consider interdependency is presented. A further consideration of these combinations would be of value.

5 Conclusion and Outlook

This paper focuses on resource selection based on geographic information. It introduces a novel category of sumarization techniques besides the geometric approaches and space partitioning approaches in this context. Hybrid approaches combine features of the two former approaches. In terms of selectivity, the two more complex hybrid techniques (KMARGrid_k^r and K-D-MBR_n^b) presented in this paper can outperform the best techniques from the geometric and space partitioning approaches. While KMARGrid_k^r achieves this at the cost of significantly greater summary sizes, K-D-MBR_n^b even outperforms the former state-of-the-art approaches with respect to summary size. Future work will mainly adress the evaluation of all these techniques on collections which are not extracted from a social network with its typical long tail distribution and therefore offer a more uniform allocation with respect to the number of image locations per resource. Furthermore, we plan to investigate the adaption of the resource summarization techniques for other application fields, such as index structures.

References

1. Thomas, P., Hawking, D.: Server selection methods in personal metasearch: a comparative empirical study. Information Retrieval 12(5), 581–604 (2009)

2. Müller, W., Eisenhardt, M., Henrich, A.: Scalable summary based retrieval in P2P networks. In: Intl. Conf. on Information and Knowledge Management, pp. 586–593 (2005)
3. Cuenca-Acuna, F., Peery, C., Martin, R.P., Nguyen, T.D.: PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. In: IEEE Intl. Symp. on High Performance Distributed Computing, pp. 236–246 (2003)
4. Kufer, S., Blank, D., Henrich, A.: Techniken der Ressourcenbeschreibung und -auswahl für das geographische Information Retrieval. In: Proceedings of the IR Workshop at LWA 2012, pp. 1–8 (2012)
5. Blank, D., Henrich, A.: Describing and Selecting Collections of Georeferenced Media Items in Peer-to-Peer Information Retrieval Systems. In: Diaz, L., Granell, C., Huerta, J. (eds.) Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications, pp. 1–20 (2012)
6. Sinnott, R.: Virtues of the haversine. Sky and Telescope 68(2), 159 (1984)
7. Vincenty, T.: Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Suvery Review 22(176), 88–93 (1975)
8. Becker, B., Franciosa, P.G., Gschwind, S., Ohler, T., Thiemt, G., Widmayer, P.: An Optimal Algorithm for Approximating a Set of Rectangles by Two Minimum Area Rectangles. In: Bieri, H., Noltemeier, H. (eds.) CG-WS 1991. LNCS, vol. 553, pp. 22–29. Springer, Heidelberg (1991)
9. Blank, D., Henrich, A.: Description and Selection of Media Archives for Geographic Nearest Neighbor Queries in P2P Networks. In: Inf. Acc. for Pers. Media Archives at ECIR 2010, pp. 22–29 (2010)
10. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers Inc., San Francisco (2005)
11. Guttman, A.: R-trees: a dynamic index structure for spatial searching. SIGMOD Rec. 14(2), 47–57 (1984)
12. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Comm. ACM 18(9), 509–517 (1975)
13. Kolahdouzan, M., Shahabi, C.: Multidimensional binary search trees used for associative searching. In: Proc. of the 30th Intl. Conf. on Very Large Data Bases, pp. 840–851 (2004)
14. Henrich, A.: The LSD^h-tree: An Access Structure for Feature Vectors. In: Proc. of the 14th Intl. Conf. on Data Engineering, pp. 262–369 (1998)
15. Seeger, B., Kriegel, H.-P.: The buddy-tree: an efficient and robust access method for spatial data base systems. In: Proc. of th 13th Intl. Conf. on VLDB, pp. 590–601 (1990)
16. Sharifzadeh, M., Shahabi, C.: VoR-tree: R-trees with Voronoi diagrams for efficient processing of spatial nearest neighbor queries. Proc. VLDB Endow. 3(1-2), 1231–1242 (2010)
17. Doulkeridis, C., Vlachou, A., Nrvag, K., Vazirgiannis, M.: Part 4: Distributed Semantic Overlay Networks. Handbook of Peer-to-Peer Networking, 1st edn. Springer Science+Business Media (2009)
18. Shen, X., Yu, H., Buford, J., Akon, M.: Handbook of Peer-To-Peer Networking, 1st edn. Springer Publishing (2009)
19. Belkin, N.J., Kantor, P., Fox, E.A., Shaw, J.A.: Combining the evidence of multiple query representations for information retrieval. Inf. Processing and Management 31(3), 431–448 (1995)
20. Hariharan, R., Hore, B., Mehrotra, S.: Discovering gis sources on the web using summaries. In: Proc. of the 8th ACM/IEEE Joint Conf. on Digital Libraries, JCDL 2008, pp. 94–103 (2008)