Selecting vantage objects for similarity indexing

R.H. van Leuken

R. C. Veltkamp

institute of information and computing sciences, utrecht university

technical report UU-CS-2008-002

www.cs.uu.nl

Selecting vantage objects for similarity indexing

R.H. van Leuken and R.C. Veltkamp
Department of Information and Computing Sciences
Utrecht University
the Netherlands

February 4, 2008

Abstract

Indexing has become a key element in the pipeline of any multimedia retrieval system, due to continuous increases in database size, data complexity and complexity of similarity measures. The primary goal of any indexing algorithm is to overcome the high computational costs that are involved with comparing the query to every object in the database, a goal that is achieved by efficient pruning to select a small set of candidate matches. Vantage indexing is an indexing technique that belongs to the category of embedding or mapping approaches, because it maps a dissimilarity space onto a vector space in such a way that traditional access methods can be used for querying. As a result of this mapping, each object is represented by a vector of dissimilarities to a small set of m reference objects, called vantage objects. Querying takes place within this vector space. The retrieval performance of a system based on this technique can be improved significantly through a proper choice of vantage objects. We propose a new technique for selecting vantage objects that addresses the retrieval performance directly, and present extensive experimental results based on three data sets of different size and modality, including a comparison with other selection strategies. The results clearly demonstrate both the efficacy and scalability of the proposed approach.

1 Introduction

The demand for efficient systems that facilitate querying by example in multimedia databases is vastly increasing. This demand is raised within a large number of application domains, including criminology (face and fingerprint recognition), musicology (music information retrieval), trademark registration (automatic trademark retrieval), medicine (DNA fingerprinting) and content-based image or video retrieval on the web. The most common retrieval operations that should be supported by these systems are *range searching* (retrieve all objects that display similarity with the query up to a certain degree) and *k-nearest neighbor searching* (retrieve the k objects that are most similar to the query). Note that the traditional classification problem is of a somewhat different nature; there determination of the query's *class* (either chosen from a predefined set or not) is requested, and the answer given by the system is in most cases either correct or wrong.

The aforementioned application domains, where these searches are of crucial importance, are likely to deal with very large databases. Therefore, content based retrieval becomes a necessity, since tagging the objects with meta data is infeasible in most cases. Yet another consequence of increasing database sizes is that *indexing* is indispensable to the system's pipeline, further motivated by higher data complexity as well as more elaborate and thus more computationally expensive similarity measures. The primary goal of any indexing algorithm is to avoid sequential search, i.e. to overcome the high computational costs that are involved with having to compare the query with every object in the database. Typically, this is achieved by efficient pruning of the database to select a small collection of candidate matches. In turn, the actual matching process can be applied to this small set of retrieved items before presenting the user the final result of the query.

Besides the efficiency of this pruning mechanism, accuracy is an important design aspect. This issue is twofold: relevant items should not be excluded from the set of candidates, nor should there be too many irrelevant items retrieved. These dismissals and hits are called *false negatives* and *false positives* respectively, the most well known corresponding performance measures are called *recall* and *precision*.

1.1 Related work

Although there are many categorizations possible for indexing methods, ranging from memory type (cache, main memory, secondary memory) to data driven (search in index built on existing data) versus hypothesis driven (characterize query and return items with similar characterizations), we focus here on the difference between *partitioning* and *mapping* approaches. In the early years of content based multimedia retrieval, the main paradigm was based on feature extraction. Objects are characterized by vectors that are composed of numerical features, and similarity between two objects is usually calculated as a Minkowski metric between their corresponding vectors. In these cases, the general type of indexing that is applied is a partitioning, whether it is a space based partitioning or a data based partitioning. Examples of these partitioning strategies, that are mostly stored in trees, are the kd-tree [1], R-tree [2] or variants such as the $R+$-tree [3] or $R*$-tree [4]. For a complete overview of these multidimensional access methods see the survey by Gaede and Günther [5]. In general, these methods either partition the data space into disjoint cells of possibly varying size (kd-tree and related work), or associate a region with each object in the data space (R-tree family).

These multidimensional access methods are basically instances of a more generic paradigm, where a dataset of object models in whatever representation (feature vectors in the above mentioned case) are matched using a model matching algorithm. In many cases, objects are represented by other types of models than feature vectors, that don't allow a space or data partitioning so easily. Examples are weighted point sets (possibly matched with the Earth Mover's Distance [6]), polygonal curves (for instance matched with turning angle functions [7]). All that is given in these cases is a dataset containing the object models and a similarity measure that outputs a distance given two models, that together span the *object space*. Tree-based space partitioning techniques can still be used for indexing purposes, but they have to be built on different grounds, since there is no feature space anymore. When the object space is metric, exemplar or pivot objects can be stored in the tree nodes to guide the search. One of the first works in this field was by Yianilos [8]. All database objects are divided into concentric rings around one or multiple pivots and then stored in a tree. These example objects are often called vantage objects or vantage points. Other examples based on this strategy are the VP-tree [9], the M-Tree [10] and the MVP-Tree [11]. These and other techniques for searching in metric spaces are surveyed by Chaves et al. [12].

A powerful alternative for storing an object space in a tree, is the mapping or embedding approach. Here, the database objects are embedded again in an embedding space. Instead of dividing the database objects in concentric rings around pivots and storing them in a tree, the pivots now function as feature descriptors; each database object is characterized by distances it has to a set of pivots. Examples of these embedding techniques are Vantage indexing [13], Fastmap [14], SparseMap [15] and MetricMap [16]. These methods are surveyed by Hjaltason and Samet [17]. A big advantage of these methods over tree-based indexing methods is that the required number of online complex distance calculations is reduced to the dimensionality of the embedding. Once the query has been positioned in the embedding space, all that is needed is a geometric range query or a nearest neighbor query where no more complex distance calculations are involved. To facilitate these searches, access methods as surveyed by Gaede and Günther [5] can be used again.

In this paper, we will focus on the last type of indexing strategy, the mapping approach. The retrieval performance of these embedding techniques is influenced by the choice of the pivots, sometimes called reference objects, exemplars or *vantage objects*. Although these methods may resemble dimensionality reduction techniques such as Principle Component Analysis (PCA) or Multi Dimensional Scaling (MDS), they have different starting points. PCA and MDS [18] reduce the dimensionality of a feature space, and actually make computations on this feature space. However, mapping approaches such as the ones mentioned before, don't assume such a feature space. The only possible feature space in this context is an n-dimensional space, where n is the number of objects in the dataset. In practice, this would mean that the distances to all objects in the database are used as feature descriptors, which would be too computationally involved.

Pękalska et al. have investigated a related problem [19]. Their aim is to select a proper set of prototype objects given a set of objects represented by dissimilarities as well, however the set of prototypes is used for classifying new objects into predefined classes rather than retrieving similar objects from the data set.

Bustos et al. [20] have investigated the selection of pivots for tree-based indexing structures, hence using a different selection criterion, namely the maximization of the difference in distance two objects have to a pivot.

Hennig and Latecki propose a loss-based strategy for selecting vantage objects [21]. The loss of a database object is defined as the real (i.e. object-space) distance between this object and its nearest neighbor in vantage space. To compute the loss of a complete vantage space, this distance is averaged over all database objects. The loss measure is minimized in a greedy way during the selection of vantage objects, by choosing a new vantage object such that the loss combined with other vantage objects is minimal. Due to the computationally expensive nature of the algorithm, the loss measure is evaluated over random subsamples of the database.

Originally, a MaxMin approach was proposed for the selection of vantage objects [13]. The first vantage object is chosen at random, all further vantage objects are chosen such that the minimum distance to the other vantage objects is maximized. As we will see however, this property does not necessarily guarantee the best choice of vantage objects.

1.2 Our contributions

Firstly, we propose two criteria to assess the quality of vantage objects that are directly concerned with the retrieval performance, namely the reduction of the number of false positives in the returned sets. Secondly, we show how to select vantage objects according to these criteria in such a way that each object in the database is a candidate vantage object, no random pre-selection is made. Another attractive property of the approach is that the selection of the vantage objects and the actual construction of the index are handled at the same time. Thirdly, we have performed extensive experimentation using three data sets of different modality and size: the MPEG-7 CE-Shape-1 part B test set, consisting of 1400 shape images, a set of the 50,000 color photographs, and a dataset containing 500,000 fragments of notated music of 5 notes each. We have compared our method to three other methods: random selection, the loss-based selection method and the originally proposed MaxMin method, which are all outperformed by the proposed approach. These experiments therefore show both the scalability and efficacy of the method.

Parts of the work described in this paper appeared previously in [22].

2 Vantage indexing

Vantage indexing [13] is an embedding technique, that is used to map a dissimilarity space (preferably metric) to a feature space in which querying takes place. To be more specific, given a multimedia database $A \subset \mathbb{U}$, where $\mathbb{U}$ is the universe of objects, and a distance measure $d : A \times A \rightarrow \mathbb{R}$, a set of m objects $A^* = \{A^*_1, ...A^*_m\}$ is selected, the so called vantage objects. The distance from each database object A_i to each vantage object is computed, thus creating a point $p_i = (x_1, ...x_m)$, such that $x_j = d(A_i, A^*_j)$. Each database object corresponds to a point in the m-dimensional vantage space, let $F(A_i)$ denote this mapping of an object A_i to a point in vantage space.

A query on the database now translates to a range-search or a nearest-neighbor search in this m-dimensional vantage space: compute the distance from the query object q to each vantage object (i.e. position q in the vantage space) and retrieve all objects within a certain range around q (in the case of a range query), or retrieve the k nearest neighbors to q (in case of a nearest neighbor query). The distance measure δ used on the points in vantage space is L_∞, so

$$\delta(F(A_1), F(A_2)) = \max_{A_v \in V} |d(A_1, A_v) - d(A_2, A_v)|, \tag{1}$$

where V is the set of vantage objects.

2.1 Performance assessment

A large variety of performance measures based on false and true positives and false and true negatives can be used to assess the quality of a retrieval system that employs vantage indexing. However, a ground truth is generally required to classify candidate matches in false and true positives, and to detect whether there are still false negatives residing in the database. Establishing a ground truth for large databases is a time consuming and demanding task involving domain expertise, that can often only be performed for a small number of queries.

In the case of embedding or mapping methods, there are other ways to assess the index quality, such as distortion [23], stress [18] or the Cluster Preserving Ratio (CPR) [24] when a known clustering exists for the objects. The key idea behind these methods is the comparison between distances according to the original similarity measure (i.e., the distances in object space) and the distances in the embedding space (e.g. the vantage space). Distortion measures how much larger or smaller the distances are in the embedding space than in the object space, whereas stress measures the overall difference in distances. While these measures provide insight into how well (in terms of distance preserving properties) the original space has been embedded in a space more appropriate for querying, they are somewhat distant from the actual retrieval application. Therefore, we propose to stretch the definition of false and true positives beyond the borders of a ground truth toward the comparison of distances, in order to allow the use of performance measures designed for retrieval applications.

In the case of a range query, given $\epsilon > 0$ (the range) and query A_q, object A_i is included in the return set of A_q if and only if $\delta(A_q, A_i) \leq \epsilon$. A false positive can now be defined as follows:

Definition 1 False positive *A_p is a false positive for query A_q if $\delta(F(A_q), FA_(p)) \leq \epsilon$ and $d(A_q, A_p) > \epsilon$.*

Note that this definition is not limited to assessing the quality of range searching, it can be applied to nearest neighbor or k-nearest neighbor searching as well. Although there is no predefined, fixed range ϵ in these cases, the distance between the query and the furthest of the nearest neighbors can be used as ϵ. This distance was exactly the required range to retrieve the requested number of nearest neighbors, and in that sense a correct yet strict threshold for determining whether the objects are true or false positives. Furthermore, it may seem awkward to use the same ϵ for both δ and d. However, recall that δ is L_∞, see (1). Since the maximum difference of all distances is taken, and not a combination, d and δ are in the same domain, so the same ϵ can be used.

Along the same lines, we can define a false negative as follows:

Definition 2 **False negative** *A_n is a false negative for query A_q if $\delta(F(A_n), F(A_p)) > \epsilon$ and $d(A_q, A_p) \leq \epsilon$.*

However, if it is known that d is metric, 100 percent recall is guaranteed for a system using vantage indexing [13]. The metric properties assure that vantage indexing is a *contractive embedding* of the object space, i.e. $\forall A_1, A_2 \in \mathbb{U}, \delta(F(A_1), F(A_2)) \leq d(A_1, A_2)$. Contractive embeddings with respect to $\mathbb{U}$ always yield 100 percent recall in similarity searches [17]. As pointed out before however, the accuracy of a retrieval system is twofold; objects relevant to the query are to be included in the result, yet objects irrelevant to the query should be excluded from the result as much as possible. In other words, precision is important as well, the number or percentage of false positives must be kept small.

We claim that by choosing the right vantage objects, the precision can increase significantly. In the next Section, we present a strategy of selecting vantage objects that is concerned with this issue of retrieval performance directly.

3 Selecting vantage objects

In this Section, we present a novel technique for selecting vantage objects that is based on two criteria that address the number of false positives (see Definition 1) in the retrieval results directly. The first criterion, *spacing*, concerns the relevance of a single vantage object. The second criterion, *correlation*, concerns the redundancy of a vantage object with respect to the other vantage objects. We propose a randomized incremental construction algorithm that selects the vantage objects according to these criteria, and builds the corresponding vantage space at the same time. We call this method Spacing-Correlation Based Selection.

The main idea of the proposed approach is to keep the number of candidates that are returned for a query A_q and range ϵ as small as possible. Of course, a priori the query object A_q is unknown, so its location in vantage space is unknown as well. Furthermore, no prior knowledge is available on the size of the range query (ϵ), or the number of nearest neighbors that will be requested. Good performance (achieved by small return sets given a query A_q and range ϵ) should therefore be scored over all possible queries and over all possible ranges ϵ.

Small return sets are a result of dispersing the objects over the vantage space, in order to avoid dense object clusters. This dispersion can only be achieved to a certain extent, since real object clusters cannot be taken apart, assuming d is metric (see Section 2.1). Given the 100 percent recall guarantee, it is exactly the number of false positives within a range around the query that is reduced by spreading out the database over the vantage space as much as possible, since these are pushed outside the borders of the range ϵ.

Another way of looking at this dispersion of the database over the vantage space is through the discriminative power of a set of vantage objects. In a vantage space, similarity between database objects is interpreted as similarity in distance to the vantage objects. In case many database objects have similar distances to the vantage objects, the vantage space is limited in its discriminative power over the database. The discriminative power of the vantage space is maximized by spreading out the database as much as possible (i.e., within the boundaries as posed by the specific dataset that is to be embedded).

The following Sections describe the two criteria (spacing and correlation) in more detail, provide some insight into finding a proper vantage space dimensionality and present the selection algorithm and its computational complexity.

3.1 Spacing

In this Section we will define a criterion for the relevance of a single vantage object V_j.

Suppose for one given vantage object, the distances to all items are marked on a vantage axis. The discriminative power can then be measured by calculating how evenly spaced the marks on this axis are. Our first criterion therefore concerns the *spacing* between objects on a single vantage axis, which is defined as follows:

Definition 3 ***Spacing*** *S_i between two consecutive objects A_i and A_{i+1} on the vantage axis of V_j is $d(A_{i+1}, V_j) - d(A_i, V_j)$.*

Let μ be the average spacing. The variance of spacing σ^2_{sp} is

$$\sigma^2_{\text{sp}} = \frac{1}{n-1} \sum_{i=1}^{n-1} ((d(A_{i+1}, V_j) - d(A_i, V_j)) - \mu)^2$$

To ensure that the database objects are evenly spread in vantage space, the variance of spacing has to be as small as possible. A vantage object with a small variance of spacing has a high discriminative power over the database, and is said to be a relevant vantage object.

Figure 1: Schematic representation of a vantage axis with object clusters (a) and a vantage axis with dispersed objects (b).

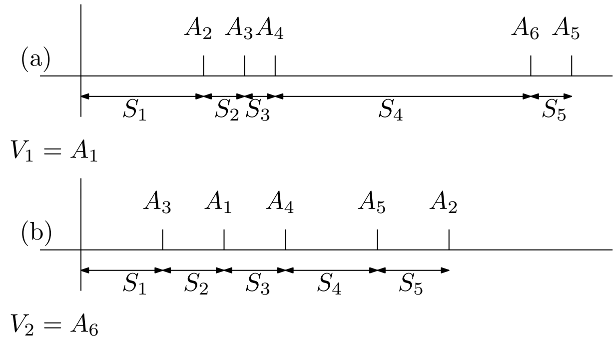
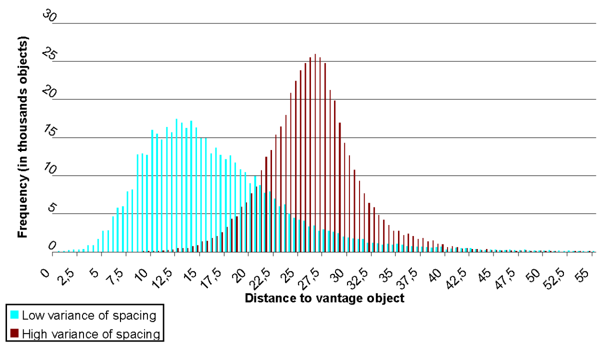


Figure 2: Distance distributions for vantage objects with a high and low variance of spacing



The spacing criterion is illustrated by Figure 1, where axes of two vantage objects are displayed schematically. Figure 1 (a) displays a vantage axis with clustered objects, resulting in a large variance of spacing compared to Figure 1 (b), where a vantage axis with dispersed objects is displayed. The spacing criterion is further illustrated by a real world example in Figure 2, where distance distributions for two vantage objects are visualized in a histogram, one with a low and one with a high variance of spacing. The dataset used here consists of 500,000 fragments of notated music; pairwise distance reflects in this case musical similarity. It can be seen that the database objects have a wider variety of distances to the vantage object with a low variance of spacing than to the vantage object with a high variance of spacing.

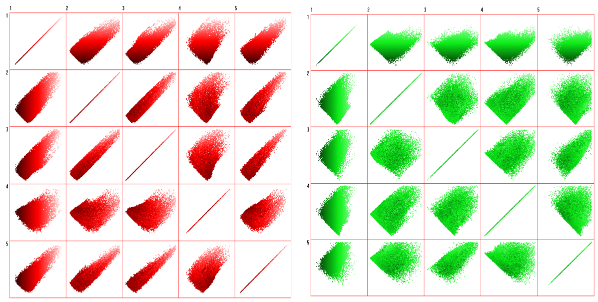
In these histograms, the distances are binned; in practice, most of the distances are unique. A large bin (e.g. around 27.5 histogram of high variance) therefore means that there are a lot of distances within a certain range around this value. As a consequence, the spacings of the distances within this bin must be small. Distances in a smaller bin are 'less packed', and thus have larger spacings in between. If the bin heights vary a lot, there exist a lot of different spacing values, resulting in a higher variance of spacing values.

3.2 Correlation

It is not sufficient to just select relevant vantage objects, they also should be non-redundant. A low variance of spacing for all vantage objects does not guarantee that the database is well spread out in vantage space, since all the vantage objects may provide a similar view on the database. Two redundant vantage objects produce the same reduction of the return set, and there is no point in using one in combination with the other. This redundancy of a vantage object with respect to another can be estimated by computing the linear correlation coefficient between the distribution of database objects along their axes. Note that two vantage objects that have a large distance to each other can still be redundant, since the distribution of distances all objects have to these vantage objects may be similar. On the other hand, one may argue that there may exist other correlations between the distribution of objects than a linear combination. In practice however this is very unlikely, and we have never seen such correlations in our experiments.

To ensure no redundant vantage objects are selected, we compute linear correlation coefficients for all pairs of vantage

Figure 3: Scatterplot matrices for 5 randomly selected vantage objects. Vantage objects were selected randomly in (a), and using the proposed method in (b). The letter shows less correlation, which is better.



(a) (b)

objects and make sure these coefficients do not exceed a certain threshold. Figure 3 illustrates the correlation criterion in a real world example, using the dataset of 500,000 fragments of music notation again. These scatterplots show how the distances of all objects in the database to two vantage objects are correlated. The maximum correlation results in a simple diagonal line, as can be seen on the diagonal of the matrices, where each vantage object is compared to itself. The scatterplot matrix on the left displays pairwise correlations for five vantage objects that where selected randomly, whereas the five vantage objects in the scatterplot on the right were selected by the proposed method, i.e. such that the correlation coefficient for every pair of vantage objects is low. Clearly, random selection of vantage objects results in stronger correlated vantage objects than the proposed method, since the objects in the right matrix are dispersed over the space much better.

3.3 The number of vantage objects

The dimensionality of the vantage space (defined by the number of vantage objects) and the retrieval performance are closely related. In general, the more vantage objects, the smaller the number of false positives will be. A vantage object cannot degrade precision scores, at worst it cannot influence the precision at all and thus be completely redundant. However, query times in a vantage space of higher dimensionality are longer. Therefore, the number of vantage objects should be set to an appropriate value given the needs of the application. In an interactive environment, the allowed dimensionality is limited, whereas in offline applications where precision is crucial, more vantage objects can be used. In our experimental work we evaluate the influence of the vantage space dimensionality on the retrieval results.

Although performance increases with a larger number of vantage objects, this increase is not necessarily gradual or unlimited; adding one vantage object doesn't make a great difference if the set is still too small to obtain good results. On the other hand, at some point it will be hard to find more vantage objects that are non redundant and performance increase with extra vantage objects will slow down. The best strategy for finding an appropriate number of vantage objects given a specific dataset and considering the needs of the application therefore, is to perform some pilot selection runs to estimate the optimal vantage space dimensionality under these constraints. Experiments with different number of vantage objects are presented in Section 4.

3.4 Algorithm

Spacing-Correlation Based Selection selects a set of vantage objects according to the criteria defined above with a randomized incremental algorithm. The key idea is to add the database objects one by one to the index while inspecting the variance of spacing and correlation properties of the vantage objects after each object has been added. As soon as either the variance of spacing of one object or the correlation of a pair of objects exceeds a certain threshold, a vantage object is replaced by a randomly chosen new vantage object. These repair steps are typically necessary only at early stages of execution of the algorithm. Since the database objects are added in random order to the index, intermediate spacing and correlation properties of vantage objects form a good estimator of the final properties once a sufficient number of database objects has been added to the index. Redundancy or small discriminative power of a vantage object can therefore be detected early on, keeping the amount of work that has to be redone (reposition all the objects that are already added with respect to the new vantage object) small. For details, see Algorithm 1.

Algorithm 1 Spacing-Correlation Based Selection
Input: Database A with objects $A_1, ..., A_n$, $d(A, A) \to \mathbb{R}$, thresholds ϵ_{corr} and ϵ_{sp}
Output: Vantage Index with vantage objects $V_1, V_2, ..., V_m$

select initial $V_1, V_2, ..., V_m$ randomly
for All objects A_i in random order **do**
  for All vantage objects V_j **do**
    compute $d(A_i, V_j)$
    add A_i to index
    if $\sigma^2_{sp}(V_j) > \epsilon_{sp}$ **then**
      remove V_j
      select new vantage object V_{new} randomly
      reposition already added objects w.r.t V_{new}
  if $\exists\{V_k, V_l|$ **Corr**$(V_k, V_l)> \epsilon_{corr}\}$ **then**
    if $\sigma^2_{sp}(V_k) > \sigma^2_{sp}(V_l)$ **then**
      remove V_k
    else
      remove V_l
    select new vantage object randomly

3.5 Complexity

The complexity of our algorithm is expressed in terms of distance calculations, since these are by far the most expensive part of the process. The running time complexity is then $O(\sum_{i=0}^{n} P_i \times i + (1 - P_i) \times k)$ where k is the (in our case constant) number of vantage objects and P_i is the chance that at iteration i a vantage object has to be replaced by a new one. This chance depends on the choice for ϵ_{spac} and ϵ_{corr}. There is a clear trade-off here: the stricter these threshold values are, the better the selected vantage objects will perform but also the higher the chance a vantage object has to be replaced, resulting in a longer running time. If we only look at spacing and set ϵ_{spac} such that for instance P_i is $(\log n)/i$, the running time would be $O(n\log n)$ since k is a small constant.

4 Experimental results

We implemented our algorithm and tested it on three data sets of different modality and size: one data set of 1,400 shape contour images, one collection of 50,000 color photographs and a set of 500,000 fragments of music notation.

An advantage of defining a false positive as in Definition 1, is that evaluating the performance on these datasets does not require a ground truth. To measure performance, the matching process is applied to the candidate matches as returned by the range query on the index. After the exact distances between the query and all candidate matches have been computed, the percentage of false positives within the returned set can be calculated. This is our first evaluation criterion.

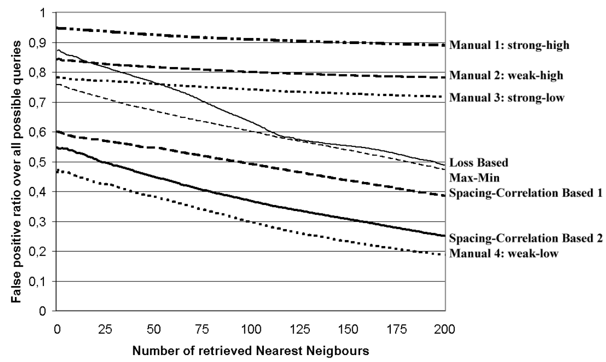
For some applications however, a shortcoming of just counting false positives is that it does not take into account the ranking of the true positives in the return sets. For this purpose, we have evaluated our results by means of a second performance measure as well: average precision. This measure is defined as the mean of the precision scores obtained after each true positive is retrieved [25]. A maximum average precision score of 1.0 is obtained when all true positives are at the top of the retrieval ranking.

During the music retrieval experiment we evaluated the results with another, third performance measure, which we call the Average Distance Error (*ADE*). Recall that a false positive A_{fp} is an object that lies within a range of ϵ to the query A_q in vantage space, but has a real distance $d(A_{fp}, A_q)$to the query that is larger than ϵ. One may argue that a false positive with a real distance to the query slightly larger than ϵ is not as bad as a false positive with a real distance exceeding ϵ by orders of magnitude. Therefore, instead of just calculating the precision scores using this definition of a false positive, one may obtain more information by addressing a weight to each false positive. This weight is defined as $d(A_{fp}, A_q) - \epsilon$, i.e. the *extent* to which a false positive is actually a false positive. The Average Distance Error is average of all these false positive weights, taken over a large set of queries.

In the next sections, we present experimental results on all three datasets, measuring the performance with measures described above.

Figure 4: Examples of the MPEG-7 data set.

Figure 5: False positive ratios on the MPEG-7 set. The lower the false positive ratio, the better the retrieval result is.



4.1 Shape retrieval

As a first dataset, we used the MPEG-7 test set CE-Shape-1 part B, consisting of 1,400 shape images, contained in 70 classes of 20 images per class [26]. A few examples are given in Figure 4.

The distance measure used to calculate the distance between two of these shape images is the Curvature Scale Space (CSS) [27]. This technique matches two shapes based on their CSS-image, which is constructed by iteratively convolving the contour with a Gaussian smoothing kernel, until the shape is completely convex. When at a certain iteration a curvature zero-crossing disappears due to the convolution process, a peak is created in the CSS-image. A CSS-image reflects in a way the effort it takes to smooth all the concavities in the curve until the image has become completely convex. Two shapes are now matched by comparing (the peaks in) their CSS-images. To justify our selection criteria, we manually selected four sets of eight vantage objects that either satisfy both criteria (weakest correlation and lowest variance of spacing: *weak-low*), none (strongest correlation and highest variance of spacing: *strong-high*) or a *strong-low* or *weak-high* combination.

The performance of these four sets of vantage objects was evaluated by querying with all 1400 objects. The number of nearest neighbors that was retrieved for each query object varied from 1 to 200. The distance of the furthest nearest neighbor functioned as ϵ, which was used to calculate the number of false positives among these nearest neighbors, see Definition 1. For each vantage index, and all k-NN queries, $k = 1, ..., 200$, an average ratio of false positives in result was calculated over all 1400 queries. The results are displayed in Figure 5, together with some typical runs of our algorithm, the MaxMin approach and the loss-based approach.

These results show that both criteria need to be satisfied in order to achieve good performance (only the set called *weak-low* scores less than 50% false positives for all sizes of nearest neighbor query). Furthermore, it shows that our algorithm can actually select a set of vantage objects in which these criteria are satisfied, since false positive ratios are low for these sets.

Table 1: False positive ratios and average precision values for the MPEG-7 set

Method (100 NN)	false positive ratio	average precision
Spacing-Correlation Based	0.51	0.42
MaxMin	0.57	0.35
Loss-based	0.71	0.22
Random	0.74	0.22

Figure 6: False positive ratios on the set of photographs. A lower false positive ratio corresponds to a better retrieval result.

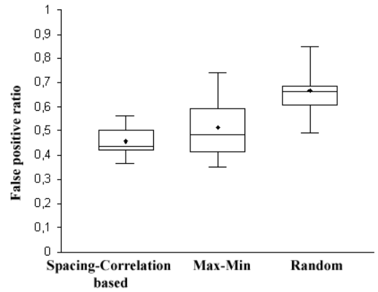


Table 1 shows similar results, concentrated on 100-nearest neighbor queries. The left column in this table lists false positive ratios averaged over 1400 queries, and averaged over a large number of selected sets of vantage objects. The right column shows average precision numbers.

4.2 Color based photo retrieval

The second dataset we used is significantly larger, it consists of 50,000 color photographs. Color histograms were constructed for these photographs, and normalized histogram matching was used as a distance measure. In this experiment, we compared our strategy for selecting vantage objects to randomly selecting the vantage objects and the MaxMin approach. Because the Loss-based method is computationally expensive to evaluate, this method has not been tested on a dataset of this size. All three strategies (Random, MaxMin and Spacing-Correlation Based) were applied multiple times (several runs), since randomness in the methods may influence the performance from one run to another. The performance for each run was measured over the same set of 1000 randomly chosen query objects, and is expressed in terms of the average false positive ratio given a fixed range and dimensionality of the vantage space. Boxplots showing the results are presented in Figure 6.

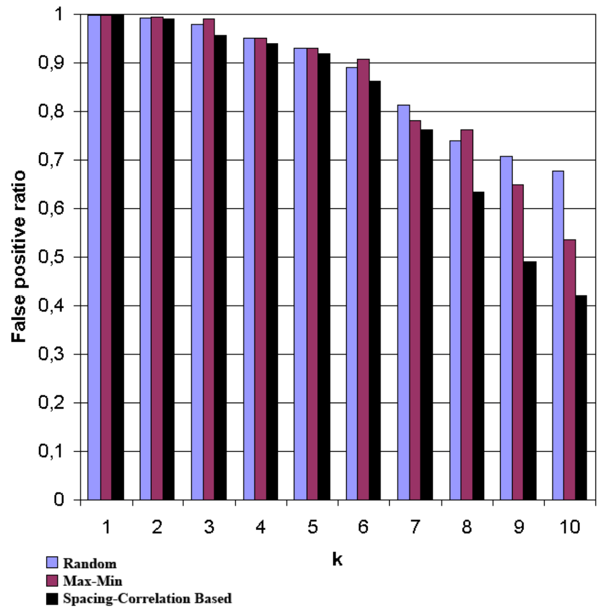
These results show that Spacing-Correlation Based selected vantage objects yield a lower false positive ratio (notice that both the median, represented by the line within the box, and the mean, represented by the diamond are lower). Furthermore, it shows that the variability over a large number of runs for Spacing-Correlation Based selection is lower. This means there is more reason to believe a specific set of Spacing-Correlation Based selected vantage objects performs well, than there is for the other selection methods, where random effects are influencing the performance wildly.

We also investigated the influence of the dimensionality of the vantage space on this dataset. Again, the range for all queries in this experiment was fixed, however the number of vantage objects that was used varied. For this experiment, we selected a typical run for each of the selection strategy and queried with 1000 random queries on each index. The results are displayed in Figure 7. These results show once more that false positive ratios are smaller for Spacing-Correlation Based selected vantage objects. In particular, they show that with well chosen objects, a vantage space of smaller dimensionality can yield the same performance as a vantage space of higher dimensionality with randomly selected vantage objects for instance. Furthermore, the higher the dimensionality of the vantage space, the larger the improvement in performance gets. This means that with Spacing-Correlation Based selection, more relevant and non-redundant objects can be found even though there are already a number of objects selected, whereas the other methods select at this point more redundant (and possibly less-relevant) vantage objects.

4.3 Music retrieval

We have compared Spacing-Correlation Based Selection to random selection on a data set of 500,000 segments of 5 notes each, from a collection of notated music. This collection is created by RISM, the Répertoire International des Sources Musicaux, and is called RISM A/II. The notes in these segments are represented as weighted points in a space in which the pitch and onset time are the axes [28] and the duration denotes the weight. The distance between two fragments is computed using a transportation distance. These transportation distances measure the minimum effort it takes to transform one weighted pointset into the other. In the case of the Earth Mover's Distance, a metaphor of transporting piles of dirt (one point set) to empty holes (the other point set) is used. The weight of the points corresponds to the mass of a pile, or the capacity of a hole. Both point sets can arbitrarily serve the role of receiver or supplier. A well known example of a transportation distance is the Earth Mover's Distance (EMD) [6]. However, for this EMD it is known that it does not obey

Figure 7: False positive ratios for different vantage space dimensionalities (k), using the set of photographs



the triangle inequality, so for this experiment the Proportional Transportation Distance [29] was used, which is a modified version of the EMD such that the triangle inequality holds.

The results for this experiment are shown in Figure 8. In vantage spaces of different dimensionality k (multiple selection runs per dimensionality), false positive ratios were computed over 1000 randomly chosen queries. Again, Spacing-Correlation Based selected vantage objects produce less false positives for all values of k than randomly selected vantage objects.

Figure 9 shows Average Distance Error (ADE) values for our music dataset. This experiment considers $k = 5$ and higher only, since smaller sets of vantage objects produce almost only false positives. These results show that Spacing-Correlation Based selection not only reduces the number of false positives; the extent to which the false positives are false is also reduced. One may therefore say that pairwise distances are better preserved using Spacing-Correlation Based selection.

5 Concluding remarks

An object space is spanned by a set of object models and a similarity measure that outputs a similarity or distance value for two given object models. Efficient querying of this object space requires indexing, since otherwise the query will have to be compared to every object in the dataset. When the object models are not feature vectors, a direct space or data partitioning of the object space is not trivial, and embedding or mapping methods are more appropriate. Every object model is mapped to a point in a new feature space, that can be partitioned using well known access methods that facilitate range or nearest neighbor querying.

Vantage indexing is such a mapping technique, where the features of the mapped object models correspond to distances they have to reference objects, called vantage objects. The selection of these vantage objects is crucial to the performance of these systems. In this paper, we have presented a new approach for selecting good vantage objects. The method, called Spacing-Correlation Based selection, uses two criteria that directly address the number of false positives. The first criterion is concerned with the relevance of the vantage objects. This individual performance of each vantage object is measured through the evenness of the distribution of distances all objects have to this vantage object. The second criterion is concerned with the redundancy of the vantage objects. The combined performance of a pair of two vantage objects is measured through the linear correlation coefficient of the distances all objects have to these two vantage objects. We have shown a selection strategy that chooses vantage objects according to these criteria, and at the same time constructs the actual index.

The approach has been tested on three real-life datasets of different size and modality: 1,400 silhouettes, 50,000 photographs and 500,000 musical segments. On all datasets, Spacing-Correlation Based selected vantage objects produce

Figure 8: False positive ratios for different vantage space dimensionalities (k) on the set of musical segments

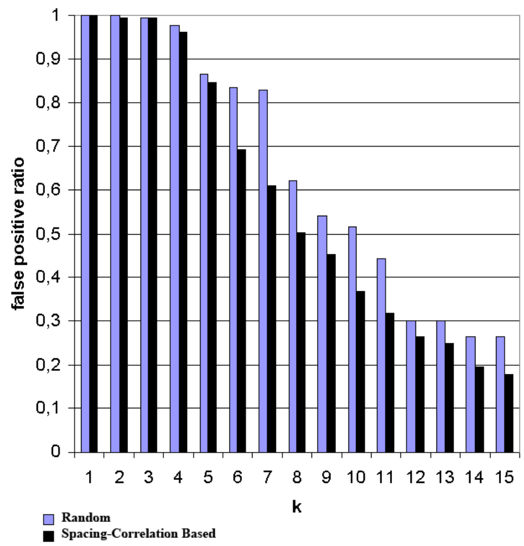
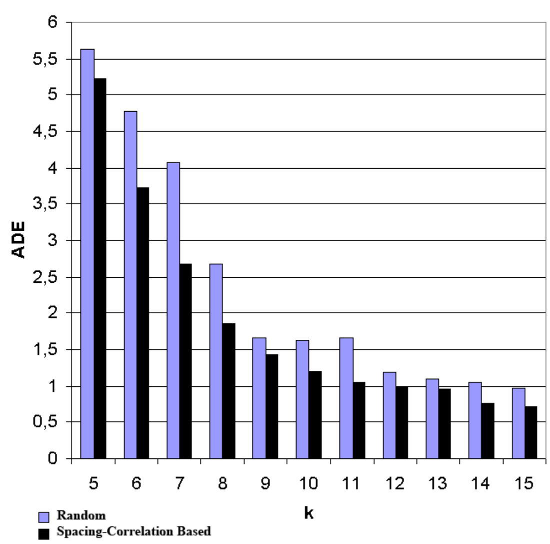


Figure 9: Average distance error for different vantage space dimensionalities (k) on the set of musical segments



significantly less false positives than random selection or other known selection techniques. In addition, we have shown that the variability in performance is smaller with Spacing-Correlation Based selection, and that the pairwise distances are better preserved.

References

[1] J. Bentley, "Binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 507–519, 1975.

[2] A. Gutman, "R-trees: A dynamic index structure for spatial searching," in *Proceedings of ACM-SIGMOD*, 1984, pp. 47–54.

[3] T. K. Sellis, N. Roussopoulos, and C. Faloutsos, "The r -tree: A dynamic index for multi-dimensional objects," in *The VLDB Journal*, 1987, pp. 507–518.

[4] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: An efficient and robust acces method for points and rectangles," in *Proceedings of ACM-SIGMOD*, 1990, pp. 322–331.

[5] V. Gaede and O. Gunther, "Multidimensional access methods," *ACM Computer Surveys*, vol. 30, no. 2, pp. 170–231, 1998.

[6] Y. Rubner, C. Tomasi, and L. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of Computer Vision*, 1998, pp. 59–66.

[7] E. M. Arkin, L. Chew, D. Huttenlocher, K. Kedem, and J. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on PAMI*, vol. 13, no. 3, pp. 209–216, 1991.

[8] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proceedings of SODA*, 1993, pp. 311–321.

[9] T. Bozkaya and M. Ozsoyoglu, "Distance-based indexing for high-dimensional metric spaces," in *Proceedings of SIGMOD*, 1997.

[10] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proceedings of VLDB*, 1997, pp. 426–435.

[11] T. Bozkaya and M. Ozsoyoglu, "Indexing large metric spaces for similarity search queries," *ACM Trans. Database Syst.*, vol. 24, no. 3, 1999.

[12] E. Chavez and G. Navarro, "Searching in metric spaces," *ACM Computer Surveys*, pp. 273–321, September 2001.

[13] J. Vleugels and R. C. Veltkamp, "Efficient image retrieval through vantage objects," *Pattern Recognition*, pp. 69–80, 2002.

[14] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of ACM SIGMOD*, 1995, pp. 163–174.

[15] G. Hristescu and M. Farach-Colton, "Cluster-preserving embedding of proteins," DIMACS, Tech. Rep. 99-50, 8, 1999.

[16] X. Wang, J. T.-L. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang, "An index structure for data mining and clustering," *Knowledge and Information Systems*, pp. 161–184, 2000.

[17] G. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," in *IEEE Transactions on PAMI*, 2003, pp. 530–549.

[18] J. Kruskal and M.Wish, "Multidimensional scaling," Beverly Hills, California," Sage Univ. Series, 1978.

[19] E. Pękalska, R. Duin, and P. Paclik, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, pp. 189–208, 2005.

[20] B. Bustos, G. Navarro, and E. Chavez, "Pivot selection techniques for proximity searching in metric spaces," *Pattern Recognition Letters*, pp. 2357–2366, 2003.

[21] C. Henning and L. J. Latecki, “The choice of vantage objects for image retrieval,” *Pattern Recognition*, pp. 2187–2196, 2003.

[22] R. H. van Leuken, R. C. Veltkamp, and R. Typke, “Selecting vantage objects for similarity indexing,” in *Proceedings of ICPR*, 2006, pp. 453–456.

[23] N. Linial, E. London, and Y. Rabinovich, “The geometry of graphs and some of its algorithmic applications,” *Combinatorica*, vol. 15, pp. 215–245, 1995.

[24] G. Histecru and M. Farach-Colton, “Cluster-preserving embeddings of proteins,” Rutgers University, Piscataway, Tech. Rep., 1999.

[25] C. Buckley and E. M. Voorhees, “Evaluating evaluation measure stability,” in *Research and Development in Information Retrieval*, 2000, pp. 33–40.

[26] L. J. Latecki, R. Lakaemper, and U. Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *Proceedings of CVPR*, 2000, pp. 424–429.

[27] F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space,” in *Proceedings of IDB-MMS*, 1996, pp. 35–42.

[28] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum, “Using transportation distances for measuring melodic similarity,” in *Proceedings of ISMIR*, 2003, pp. 107–114.

[29] P. Giannopoulos and R. C. Veltkamp, “A Pseudo-Metric for Weighted Point Sets,” in *Proceedings of ECCV*, 2002, pp. 715–730.