Selecting Vantage Objects for Similarity Indexing

REINIER H. VAN LEUKEN and REMCO C. VELTKAMP, Utrecht University

Indexing has become a key element in the pipeline of a 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 high computational costs involved with comparing the query to every object in the database. This is achieved by efficient pruning in order to select only 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 such that traditional access methods can be used for querying. 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.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Multimedia retrieval, indexing, embedding methods, vantage objects

ACM Reference Format:

Van Leuken, R. H. and Veltkamp, R. C. 2011. Selecting vantage objects for similarity indexing. ACM Trans. Multimedia Comput. Commun. Appl. 7, 3, Article 16 (August 2011), 18 pages.
DOI = 10.1145/2000486.2000490 http://doi.acm.org/10.1145/2000486.2000490

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

Authors' address: Information and Computing Sciences, Utrecht University, PO Box 80.089, 3508TB Utrecht, The Netherlands; email: {reinier, r.c.veltkamp}@cs.uu.nl.

DOI 10.1145/2000486.2000490 http://doi.acm.org/10.1145/2000486.2000490

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 metadata 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, that is, 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 with the final result.

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. Such incorrect dismissals and hits are called *false negatives* and *false positives*, respectively. The most well-known corresponding performance measures are called *recall* (number of retrieved relevant items with respect to total number of relevant items) and *precision* (number of retrieved relevant items with respect to total number of retrieved items).

In this article we focus on a specific indexing strategy, called vantage indexing [Vleugels and Veltkamp 2002]. With vantage indexing, objects are no longer compared directly, but investigated as to how similar their resemblance is to a set of reference objects: the vantage objects. In a sense, vantage indexing provides an embedding of a general metric space into a vector space. It therefore belongs to a well-studied and popular class of indexing strategies, extensively surveyed in a book by Samet [2006]. We focus particularly on the selection of vantage objects, since a good choice of vantage objects increases retrieval performance.

1.1 Our Contributions

First, 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. Second, 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 preselection 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. Third, we performed extensive experimentation using three data sets of different modality and order of magnitude: 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 five notes each. We have compared our method to five other methods: random selection, the loss-based selection method, the originally proposed MaxMin method, Sparse Spatial Selection, and Maximum Mean Distance, which are all outperformed by the proposed approach.

Parts of the work described here appeared previously in van Leuken et al. [2006].

2. VANTAGE INDEXING

Vantage indexing [Vleugels and Veltkamp 2002] 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 $A \subset \mathrm{U}$, where U is the universe of objects, and a distance measure $d : A \times A \to \mathbb{R}$, a set of m objects $A^* = \{A_1^*, \ldots, A_m^*\}$ is selected: the vantage objects. The distance from each database object A_i to each vantage object is computed, thus creating a point $p_i = (x_1, \ldots, x_m)$ 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). Distances in vantage space are bounded by each individual distance to a vantage object. Therefore, 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.

Each object is now represented as a point in m-dimensional vector space, and all those points together are stored in a balanced box tree, so that an approximate nearest-neighbor can be found in $O(\log n)$ search time [Arya et al. 1994]. Vantage indexing is an efficient way of indexing. Instead of computing a (possibly complex) distance measure in object space between a query and all database objects, only a fixed number m such distances are computed, followed by a cheap $O(\log n)$ time search in vector space.

The implications of using vantage indexing is that the data structure to store all distances in embedding space is kept in main memory. This is different from usual database management systems, although also for (relational) database management systems there is a trend to keep the complete index in main memory (for instance the Monet database system). Although the data structure (a balanced box tree) is potentially large, the approximate nearest-neighbor search that we use works best when the dimension is not more than about 20 a rule-of-thumb that helps to keep memory consumption reasonable.

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 into 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, which 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 [Linial et al. 1995], stress [Kruskal and M.Wish 1978] or the Cluster Preserving Ratio (CPR) [Histecru and Farach-Colton 1999] 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. With the following definitions, performance assessment is independent of human judgment (which is often instable and expensive to obtain for many queries). Moreover, in this way the quality of the matching algorithm or distance measure definition does not affect the performance evaluation of the indexing algorithms.

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 2.1. *False positive* A_p is a false positive for query A_q if $\delta(F(A_q), F(A_p)) \leq \epsilon$ 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.

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

Definition 2.2. *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, under metric conditions, 100 percent recall is guaranteed for a system using vantage indexing [Vleugels and Veltkamp 2002].

LEMMA 2.3. *No false negatives are possible for a query in a vantage index when the underlying distance measure d obeys metric properties.*

PROOF. Let object A_i be close to query in A_q in object space, that is, $d(A_i, A_q) \leq \epsilon$. A_i is returned for A_q from the vantage space if $d(A_i, A_v) - d(A_q, A_v) \leq \epsilon$ for all vantage objects A_v. As d obeys metric properties, by the triangle inequality, it is known that $d(A_i, A_v) \leq (A_i, A_v) + d(A_q, A_v)$. Since $d(A_i, A_q) \leq \epsilon$, it follows that $d(A_i, A_v) \leq d(A_q, A_v) + \epsilon$. Therefore, $d(A_i, A_v) - d(A_q, A_v) \leq \epsilon$, which is the condition for retrieval of A_i. □

This proof shows that, under metric conditions, an object that is within distance ϵ to the query in object space will always be retrieved from the vantage space with a range query of range ϵ. Therefore, a query will never yield false negatives as defined in Definition 2.2. When d violates the triangle inequality constraint, this guarantee is no longer given, and false negatives may occur. In practice, when the triangle inequality violation is within reasonable bounds, this can be overcome by searching with a larger ϵ.

Furthermore, the preceding proof illustrates why the same ϵ can be used for both δ and d: A_i is only retrieved from the vantage space if $d(A_i, A_v) - d(A_q, A_v) \leq \epsilon$ for all vantage objects A_v. On a conceptual level, this means that the vantage space distance between A_q and A_i is calculated with respect to individual vantage objects, and the largest of these distances determines the total vantage space distance. This is a direct consequence of using L_∞, sometimes referred to as $L_{\max}$ for δ. Since the maximum distance is taken, and not a combination, d and δ are in the same domain, so the same ϵ can be used.

In summary, the metric properties assure that vantage indexing is a *contractive embedding* of the object space, that is, $\forall A_1, A_2 \in \mathrm{U}, \delta(F(A_1), F(A_2)) \leq d(A_1, A_2)$. Contractive embeddings with respect to U always yield 100 percent recall in similarity searches [Hjaltason and Samet 2003]. 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, and the number or percentage of false positives must be kept small. By choosing the right vantage objects, the precision can increase significantly.

3. RELATED WORK

We distinguish between *partitioning* and *mapping* indexing 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 the 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, which are mostly stored in trees, are the kd-tree [Bentley 1975], the R-tree [Gutman 1984] or variants such as the $R+$-tree [Sellis et al. 1987] or $R*$-tree [Beckmann et al. 1990]. For a complete overview of these multidimensional access methods, see the surveys by Gaede and Günther [1998] and Böhm et al. [2001]. 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 (the 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, which don't allow easy space or data partitioning. Examples are weighted point sets (possibly matched with the Earth Mover's Distance [Rubner et al. 1998]) and polygonal curves (for instance, matched with turning angle functions [Arkin et al. 1991]). All that is given in these cases is a dataset A containing the object models and a distance measure d that outputs a distance given two models, A and d 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 [1993]. 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 [Bozkaya and Ozsoyoglu 1997], the M-Tree [Ciaccia et al. 1997], and the MVP-Tree [Bozkaya and Ozsoyoglu 1999]. These and other techniques for searching in metric spaces are surveyed by Chavez et al. [2001].

A powerful alternative for storing an object space in a tree is the mapping or embedding approach. Here, the database objects are again embedded 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 the distances it has to the pivots. Examples of these embedding techniques are Vantage indexing [Vleugels and Veltkamp 2002], Fastmap [Faloutsos and Lin 1995], SparseMap [Hristescu and Farach-Colton 1999], and MetricMap [Wang et al. 2000]; they are surveyed by Hjaltason and Samet [2003]. 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 [1998], can be used again.

In this article, we 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 [Kruskal and M.Wish 1978] reduce the dimensionality of a feature space, and actually make computations on this feature space. However, mapping approaches such as the ones mentioned before, do not 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 means that the distances to all objects in the database are used as feature descriptors, which is too computationally involved.

Pękalska et al. [2005] investigated a related problem. Their aim was 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.

A strategy similar to the one proposed in this article was proposed by Venkateswaran et al. [2006]. Not every database element is a candidate vantage object (or *reference*, as the authors call them), in contrast to our method, where each element is a candidate. Furthermore, their selection criteria are based on variance of distance (relevance) and distance between vantage objects (redundancy), whereas our selection criteria are based on variance of spacing (relevance) and correlation between vantage objects (redundancy). It is important to note that distance between vantage objects is not necessarily a good predictor for redundancy. In particular, when the dataset is not uniformly distributed, two vantage objects might be far apart but still have similar distances to the other objects, and may thus be redundant. The criterion used for assessing the relevance of a single object (variance of distance) might not be a good predictor either: it depends too much on the magnitude of the distances instead of their distribution. We will provide more details and two examples in Section 4. Finally, their criteria are evaluated over only a sample of the database.

BoostMap, as proposed by Athitsos et al. [2004], is a fundamentally different approach. Using a popular machine-learning technique, the AdaBoost framework, the combine many 1-dimensional classifiers into a high-dimensional classifier. In this case, a 1-dimensional classifier corresponds to a vantage object that, for a particular triplet (q, p_1, p_2), decides whether p_1 is closer to q than p_2 or not. The goal is to provide a combined high-dimensional classifier that outputs a similarity ranking for q that reflects the true ordering of the database with respect to q. The nature of the algorithms (machine-learning with intensive training rounds) and the fact that the embedding is designed to produce a ranking that is order-preserving rather than distance-preserving, make it fundamentally different from the proposed method.

The following methods have been implemented and compared to the proposed method; for experimental results see Section 5.

Bustos et al. [2003] propose to maximize the mean of distances d_μ between all objects in the embedded space or vantage space in order to disperse the objects evenly. They provide three algorithms implementing this criterion. As the best performing algorithm, they chose a greedy approach, which iteratively selects as the next vantage object the one that produces the largest d_μ, given the vantage objects that were already selected. In this article, we refer to this method as the Maximum Mean Distance, (MMD). A large drawback of MMD is the underlying assumption that the distribution of distances is uniform. When this distribution is not uniform (e.g., when there are strong clusters), maximizing d_μ does not necessarily produce an even spread of the objects in vantage space. Moreover, only a small set of objects is a candidate to be selected as a vantage object, and the selection criterion is evaluated only over a sample of distances.

Brisaboa et al. [2006] assume the distribution of distances to be uniform as well. They propose the following heuristic, called Sparse Spatial Selection (SSS): when a certain database object has a large enough distance to all the currently selected vantage objects, it is added to the set of vantage objects. A large advantage of this approach is that it does not require a predefined vantage space dimensionality. In their experiments, they show that the selected number of vantage objects reflects the *intrinsic dimensionality* of the dataset. However, a drawback of this method is that it is only concerned with the combined performance of vantage objects and not with their individual quality. Moreover,

database objects that are inspected first have a larger chance of becoming a vantage object. Finally, their selection criterion is based on the assumption that the distribution of distances is uniform.

Hennig and Latecki [2003] propose a loss-based strategy for selecting vantage objects. The loss of a database object is defined as the real (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 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.

When vantage indexing was introduced, a MaxMin approach was proposed for the selection of vantage objects [Vleugels and Veltkamp 2002]. 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.

4. SELECTING VANTAGE OBJECTS

In this section we present Spacing-Correlation Based Selection, a novel technique for selecting vantage objects that is based on two criteria that directly address the number of false positives (see Definition 2.1) in the retrieval results. 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.

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 should therefore be scored over all possible queries and over all possible query sizes. As good performance is achieved by aiming for small return sets, this notion gives rise to our definition of the vantage object quality criteria. To obtain small return sets for all queries and all query sizes, the database objects need to be dispersed, spread out over the vantage space as much as possible.

This dispersion can only be achieved to a certain extent, since for example, real object clusters cannot be taken apart, assuming d is metric (see Section 2.1). Given the 100 percent recall guarantee, only the number of false positives within a range around the query is reduced, since by spreading out the database over the vantage space as much as possible, 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).

4.1 Spacing

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:

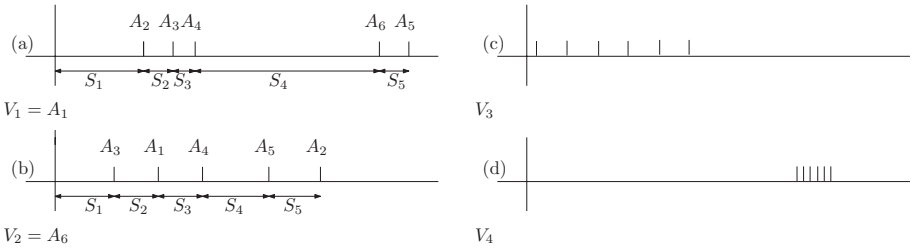


Fig. 1. On the left: schematic representation of a vantage axis with object clusters (a) and a vantage axis with dispersed objects (b). On the right: two vantage axes with a similar variance of spacing (equally uniform), but with different variance of distance.

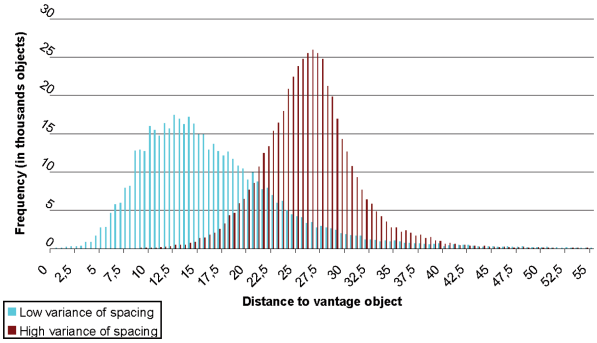


Fig. 2. Real-life example of distance distributions for vantage objects with a high and low variance of spacing (music dataset).

Definition 4.1. *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_{sp}^2 is

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

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 considered relevant.

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. Furthermore, Figure 1(c) and (d) illustrate the drawback of the variance of distance criterion for redundancy, used for instance in Venkateswaran et al. [2006]. The variance of distance in Figure 1(c) is larger than in Figure 1(d), so V_3 would be favored over V_4. However, the distribution of the objects along the axis is equally uniform in both cases, so they are equally suited to serve as a vantage object. This is reflected in the variance of spacing criterion; in both cases this value is close to zero.

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 in spacing. The dataset used here consists of 500,000 fragments of notated music; in this case, pairwise distance reflects musical similarity. It can be seen that the database objects have

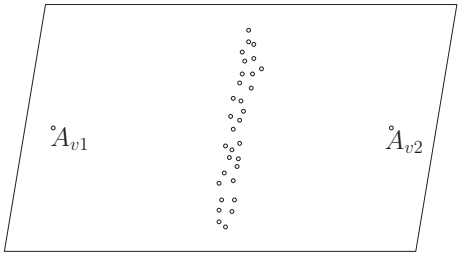


Fig. 3. Situation in which vantage objects A_{v1} and A_{v2} are far apart, but have similar distances to the other objects. These two vantage objects would not have been selected together based on the correlation criterion.

a wider variety of distances to the vantage object with a low variance in spacing than to the vantage object with a high variance in 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 in spacing values.

4.2 Correlation

It is not sufficient to just select relevant vantage objects, they should also be non-redundant. A low variance in 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 in the return set, and there is no point in using one in combination with the other. This redundancy of a vantage object V_1 with respect to another vantage object V_2 can be estimated by computing the linear correlation coefficient C between the distribution of database objects along their axes:

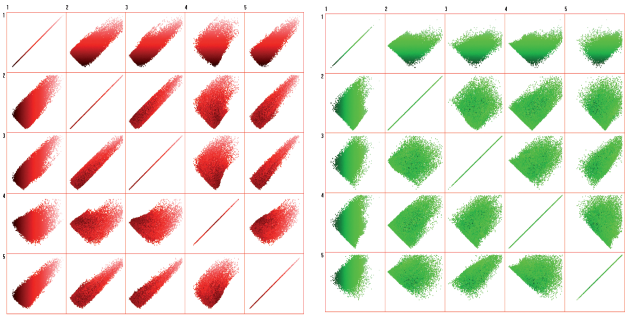
$$C(V_1, V_2) = \frac{\sum_i (d_{1i} - d_{2i}) - \sum_i d_{1i} \sum_i d_{2i}}{\sqrt{n \sum_i (d_{1i})^2 - (\sum_i d_{1i})^2} \sqrt{n \sum_i (d_{2i})^2 - (\sum_i d_{2i})^2}}$$

where d_{1i} and d_{2i} are short for, respectively, $d(V_1, A_i)$ and $d(V_2, A_i)$, that is, the distances between object A_i and vantage objects V_1 and V_2.

Note that two vantage objects that have a large distance to each other can still be redundant, since the distribution of distances that all objects have to these vantage objects may be similar. The distance criterion, used for instance in Venkateswaran et al. [2006], relies heavily on the assumption that the object space is uniform. For instance, see Figure 3: the two vantage objects lying on the left and the right of the mass of objects have a large distance to each other, but their discriminative power over the dataset is similar. The distribution of distances that all objects have to the vantage objects are almost equal.

On the other hand, we may argue that there may exist other correlations between the distribution of objects than a linear correlation. In practice, however, this is very unlikely, and we have never seen such correlations in our experiments.

To ensure that no redundant vantage objects are selected, we compute linear correlation coefficients for all pairs of vantage objects and make sure these coefficients do not exceed a certain threshold. Figure 4 illustrates the correlation criterion in a real-world example, using the dataset of 500,000



(a) (b)

Fig. 4. Scatterplot matrices for two sets of five selected vantage objects (music dataset). Vantage objects were selected randomly in (a), and by using the proposed method in (b). The latter shows weaker correlation, which is preferable.

fragments of music notation. 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 correlated with itself. The scatterplot matrix on the left displays pairwise correlations for five vantage objects that were selected randomly, whereas the five vantage objects in the scatterplot on the right were selected by the proposed method. 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.

4.3 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. A vantage object cannot degrade precision scores, at worst it cannot influence the precision at all, and thus is 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 evaluated 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 does not 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 nonredundant and the increase in performance 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, is to perform some pilot selection runs to estimate the optimal vantage space dimensionality under these constraints. Experiments with different numbers of vantage objects are presented in Section 5.

Another aspect influencing to the proper number of vantage objects is the *intrinsic dimensionality* ρ of the dataset. Given an object A_i from a dataset A consisting of n objects and a metric defined on these objects, we can say that n features are known for A_i, that is, its distances to all other objects. Hence this specific dataset spans an n-dimensional space. However, without loss of information, that is, while preserving the pairwise distances, this database can probably be embedded in a space of a lower dimensionality d. This value of d is closely related to the intrinsic dimensionality *rho* of the

dataset, which can be estimated with $\rho = \mu^2/2\sigma^2$ [Chavez and Navarro 2001], where μ denotes the mean and σ^2 the variance of the distances. It is a promising idea to translate a definition of intrinsic dimensionality to a function with which an appropriate number of vantage objects can be estimated. A possible drawback of the definition given above would be its dependency on a normal distribution: the definition is based on the behavior of uniformly distributed vector spaces under Minkowski metrics. In many practical cases, however, the vantage space is not uniformly distributed. It is interesting to see how much the estimation would suffer from this discrepancy, or how the approach could be relieved from the strong assumption of normality.

4.4 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. Typically these repair steps are, necessary only at the early stages in the execution of the algorithm. Since the database objects are added to the index in random order, intermediate spacing and the 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 the small discriminative power of a vantage object can therefore be detected early on, keeping the amount of work that has to be redone small (repositioning all the objects already added with respect to the new vantage object); see Algorithm 1.

ALGORITHM 1: Spacing-Correlation Based Selection

Input: Database A with objects $A_1, \ldots, A_n$, $d(A, A) \rightarrow$ 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 σ²_sp(V_j) > ε_sp then
            remove V_j
            select new vantage object V_new randomly
            reposition already added objects w.r.t V_new
    if ∃{V_k, V_l | Corr(V_k, V_l) > ε_corr} then
        if σ²_sp(V_k) > σ²_sp(V_l) then
            remove V_k
        else
            remove V_l
        select new vantage object randomly
```

4.5 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 distance calculation itself is considered a black box operation, and is not included in this complexity analysis; it depends totally on the application at hand (object type and corresponding distance function). Furthermore, note that index construction and vantage object selection are performed simultaneously. This means that at each iteration, a new

database object is added to the index and the criteria are evaluated over the extended index. When all objects have been added to the index, the final set of vantage objects satisfies the criteria and the complete index is obtained.

For a dataset containing n objects, the running time complexity is therefore $O(\sum_{i=0}^{n} P_i \times i + (1-P_i) \times k)$ 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 of ϵ_{spac} and ϵ_{corr}. There is a clear tradeoff 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_{sp} 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.

5. 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. The index structures reduce the query time from an order of hours for a linear scan in object space, to an order of a second when using vantage indexing. This section gives the results of the evaluation of the quality of the vantage object selection.

An advantage for defining a false positive, as in Definition 2.1, 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. In fact, evaluation with respect to a ground truth would assess the similarity measure. Here, however, we want to evaluate the quality of the vantage selection algorithm, regardless the similarity measure.

Please recall that with the vantage indexing scheme, or any *contractive embedding* in general, false negatives (see Definition 2 in Section 2.1) are avoided. Hence all query operations in these experiments yield a recall of 100%; the goal is to reduce the number of false positives in the returned sets.

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: that is, average precision. This measure is defined as the mean of the precision scores obtained after each true positive is retrieved [Buckley and Voorhees 2000]. 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 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. We 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 far exceeding ϵ. Therefore, instead of just calculating the precision scores using this definition of a false positive, we may obtain more information by addressing a weight to each false positive. This weight is defined as $d(A_{fp}, A_q) - \epsilon$, that is, the *extent* to which a false positive is actually a false positive. The Average Distance Error is the average of all these false positive weights, taken over a large set of queries.

5.1 Shape Retrieval

Our first dataset is the MPEG-7 test set CE-Shape-1 part B, consisting of 1,400 shape images (contours only), contained in 70 classes (e.g., apple, car, bat) of 20 images each [Latecki et al. 2000]. The number of vertices per contour is up to a few hundred. We have calculated the distances between two of these

Table I. False Positive Ratios and Average Precision for the MPEG-7 Set

Method (100 NN)	False Positive Ratio	Average Precision
SSS (m = 2)	0.81	0.27
Loss based	0.48	0.47
MMD	0.46	0.49
SSS (m = 8)	0.41	0.52
MaxMin	0.39	0.53
Spacing-Correlation based	0.21	0.65

contour images using Curvature Scale Space (CSS) [Mokhtarian et al. 1996]. The CSS is built by an iterative procedure to convolve the contour until all reflex vertices are eliminated.

In this experiment, we compare Spacing-Correlation-based selection to four existing methods which were all described in Section 3. These methods are called Sparse Spatial Selection (SSS) [Brisaboa et al. 2006], Maximum Mean Distance (MMD) [Bustos et al. 2003], Loss-based [Henning and Latecki 2003], and MaxMin [Vleugels and Veltkamp 2002]. All the methods were set to select eight vantage objects, except for SSS, which chooses its own number of vantage objects. Using the parameters reported in Brisaboa et al. [2006], the method selected for this dataset selects only two vantage objects. Because this leads to bad results, we reparametrized the method manually by trial and error such that it selects eight vantage objects as well.

The performance of these selection methods was evaluated by querying in their vantage spaces with all 1400 objects. The number of nearest neighbors that was retrieved for each query object ranges from 20 to 100. 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 2.1. For each vantage index, and all k-NN queries, $k = 20, \ldots, 100$, an average ratio of false positives was calculated over all 1400 queries. The results are displayed in Figure 5.

Although it may seem counterintuitive that the ratio of false positives declines when more nearest neighbors are retrieved, it is a natural consequence of the definition of a false positive. This definition is dependent on ϵ, so the definition of a false positive may change when more nearest neighbors are retrieved. Specifically, since the distance between the query and the furthest nearest neighbor that is retrieved defines ϵ, the definition will become more tolerant when more nearest neighbors are retrieved. As stated before, the advantage of this definition of a false positive precludes the need of a ground truth and makes performance comparison independent of the quality of the matching algorithm.

This experiment clearly shows that Spacing-Correlation-based selection outperforms the other selection techniques. This is mainly because the method achieves a better spread of the database objects in vantage space, since the selection criteria do not assume that the distances in object space are uniformly distributed. For example, MMD may select the vantage objects such that there exist clusters in vantage space. In this case, the spread is not even, but the mean distance between objects in vantage space may still be high. Furthermore, Spacing-Correlation-based selection selects both relevant and nonredundant vantage objects, whereas SSS is only concerned with nonredundancy.

Table I shows similar results, concentrated on 100-nearest-neighbor queries: on the left, false positive ratios averaged over 1400 queries; on the right, average precision.

5.2 Color-Based Photo Retrieval

The second dataset we used is an order of magnitude larger; it consists of 50,000 color photographs of 512×512 pixels. Color histograms of 64 bins were constructed for these photographs, and normalized

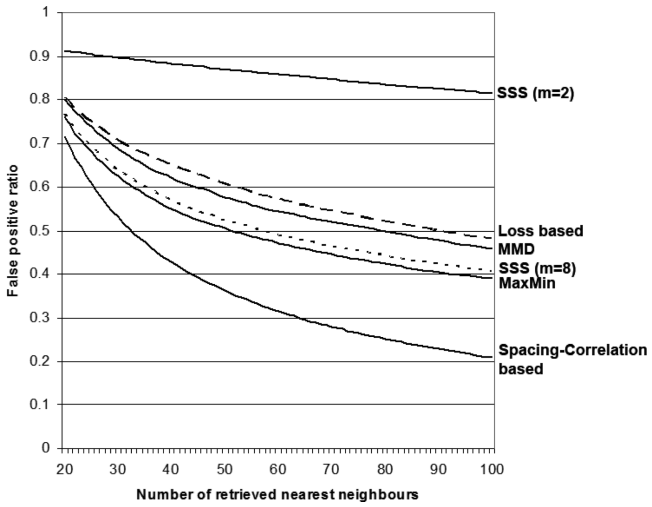


Fig. 5. Performance comparison of the proposed method with four existing methods: Sparse Spatial Selection (SSS), Loss-based, Maxmimum Mean Distance (MMD), and MaxMin. The figure shows false positive ratios, averaged over all 1400 queries from the MPEG-7 set (y-axis) with respect to the number of retrieved nearest neighbors (x-axis). In all cases the number of vantage objects was eight, except for one automatic run of SSS (m = 2). A lower false positive ratio indicates a better retrieval result.

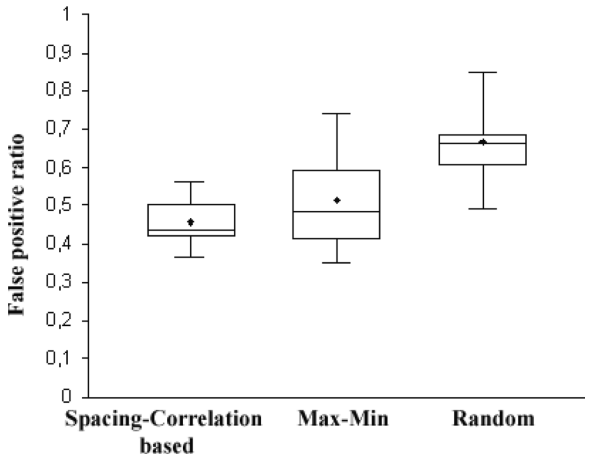


Fig. 6. Performance comparison of the proposed method with the MaxMin and random selection. The figure shows boxplots of false positive ratios on 1000 random queries from the set of photographs. A lower false positive ratio corresponds to a better retrieval result. Vantage space dimensionality: 16.

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. The methods in this experiment 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, see Figure 6 for results.

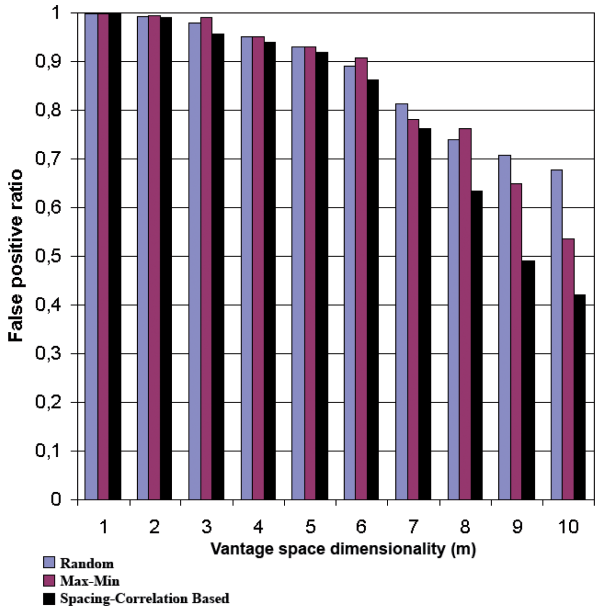


Fig. 7. Performance comparison of the proposed method with the MaxMin approach and random selection. The figure shows false positive ratios, averaged over 1000 random queries from the set of photographs (y-axis) with respect to vantage space dimensionality m (x-axis).

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 in 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. There is more reason to believe that a specific set of Spacing-Correlation-based selected vantage objects performs well than there is for the other selection methods, where random effects wildly influence the performance.

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; see Figure 7 for results. These results show once again that false positive ratios are smaller for Spacing-Correlation-based selection. 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, for example, randomly selected vantage objects. Furthermore, the higher the dimensionality of the vantage space, the larger the improvement in performance. This means that with Spacing-Correlation-based selection, more relevant and nonredundant objects can be found even though there are already a number of objects–selected, whereas at this point the other methods select more redundant vantage objects.

5.3 Music Retrieval

This third experiment is designed to illustrate intrinsic properties of our algorithm. First, we demonstrate scalability of our own algorithm on a dataset that is an order of magnitude larger than experiment 2, and two orders of magnitude larger than experiment 1. Second, we demonstrate that our algorithm not only reduces the number of false positives, but also the extent to which the false positives are false.

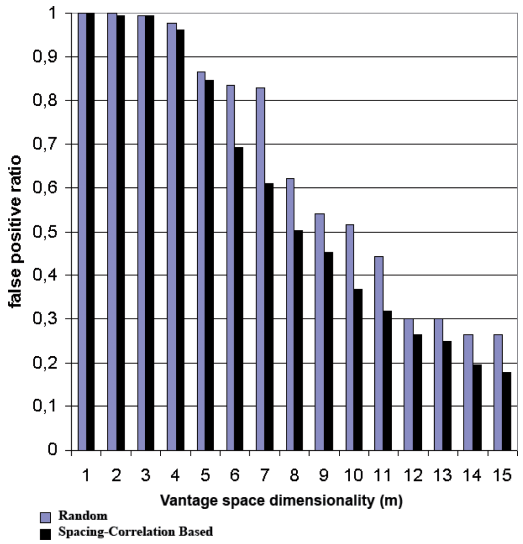


Fig. 8. Performance comparison of the proposed method with random selection. The figure shows false positive ratios, averaged over 1000 random queries from the set of musical segments (y-axis) with respect to vantage space dimensionality m (x-axis).

We have compared Spacing-Correlation-based selection to random selection on a data set of yet another order of magnitude; it consists of 500,000 segments of 5 notes each, from a collection of notated music. The notes in these segments are represented as weighted points in a space in which the pitch and onset time are the axes [Typke et al. 2003] and the duration denotes the weight. The distance between two fragments is computed using the Proportional Transportation Distance [Giannopoulos and Veltkamp 2002], which is a pseudo-metric version of the Earth Mover's Distance [Rubner et al. 1998], and which can be computed by a linear programming algorithm, for example, the simplex method between two sets of five nodes.

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

Figure 9 shows Average Distance Error (ADE) values for our music dataset. This experiment considers $m = 5$ 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; but the extent to which the false positives are false is also reduced. We may therefore say that pairwise distances are better preserved using Spacing-Correlation-based selection.

6. CONCLUDING REMARKS

Given a large set of object models and a corresponding distance measure, an indexing method is needed to perform efficient querying. Otherwise the query will have to be compared to every object in the dataset. Vantage indexing is a technique that belongs to the mapping approaches, where the features of the mapped object models correspond to distances they have to reference objects, called vantage objects. In this article we have presented Spacing-Correlation-based selection, which is a new approach for selecting good vantage objects. Two quality criteria were defined for vantage objects: variance of spacing (individual performance) and correlation (combined performance). Vantage objects that satisfy these criteria possess high discriminative power over the dataset and allow high-precision querying.

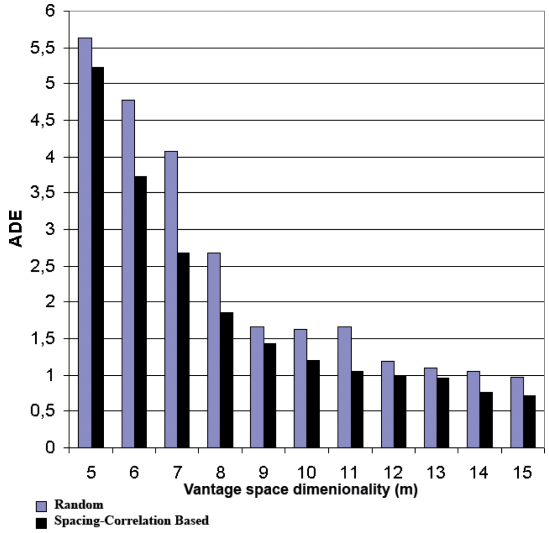


Fig. 9. Performance comparison of the proposed method with random selection. The figure shows the Average Distance Error (ADE), averaged over 1000 random queries from the set of musical segments (y-axis) with respect to the vantage space dimensionality m (y-axis).

The approach was 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 significantly fewer false positives than 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

ARKIN, E. M., CHEW, L., HUTTENLOCHER, D., KEDEM, K., AND MITCHELL, J. 1991. An efficiently computable metric for comparing polygonal shapes. *Patt. Anal. Mach. Intell.13*, 3, 209–216.

ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. 1994. An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of the 5th ACM SIAM Symposium on Discrete Algorithms*. 573–582.

ATHITSOS, V., ALON, J., SCLAROFF, S., AND KOLLIOS, G. 2004. Boostmap: A method for efficient approximate similarity rankings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR'04*). Vol. 2, IEEE, Los Alamitos, CA, 268–275.

BECKMANN, N., KRIEGEL, H., SCHNEIDER, R., AND SEEGER, B. 1990. The r*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (*SIGMOD'90*). ACM, New York, 322–331.

BENTLEY, J. 1975. Binary search trees used for associative searching. *Comm. ACM 18*, 9, 507–519.

BÓHM, C., BERCHTOLD, S., AND KEIM, D. A. 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv. 33*, 3, 322–373.

BOZKAYA, T. AND OZSOYOGLU, M. 1997. Distance-based indexing for high-dimensional metric spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (*SIGMOD 97*). ACM, New York.

BOZKAYA, T. AND OZSOYOGLU, M. 1999. Indexing large metric spaces for similarity search queries. *Trans. Datab. Syst. 24*, 3.

BRISABOA, N., FARINA, A., PEDREIRA, O., AND REYES, N. 2006. Similarity search using sparse pivots for efficient multimedia information retrieval. In *Proceedings of the 8th IEEE International Symposium on Multimedia* (*ISM'06*). IEEE, Los Alamitos, CA, 881–888.

BUCKLEY, C. AND VOORHEES, E. M. 2000. Evaluating evaluation measure stability. In *Research and Development in Information Retrieval*, 33–40.

BUSTOS, B., NAVARRO, G., AND CHAVEZ, E. 2003. Pivot selection techniques for proximity searching in metric spaces. *Patt. Recogn. Lett.* 2357–2366.

CHAVEZ, E. AND NAVARRO, G. 2001. Searching in metric spaces. *ACM Comput. Surv. 33*, 3, 273–321.

CIACCIA, P., PATELLA, M., AND ZEZULA, P. 1997. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd VLDB Conference*. 426–435.

FALOUTSOS, C. AND LIN, K.-I. 1995. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (*SIGMOD'95*). ACM, New York, 163–174.

GAEDE, V. AND GUNTHER, O. 1998. Multidimensional access methods. *ACM Comput. Surv. 30*, 2, 170–231.

GIANNOPOULOS, P. AND VELTKAMP, R. C. 2002. A pseudo-metric for weighted point sets. In *Proceedings of the European Conference on Computer Vision (ECCV'02)*. Lecture Notes in Computer Science, vol. 2352, Springer, Berlin, 715–730.

GUTMAN, A. 1984. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (*SIGMOD'84*). ACM, New York, 47–54.

HENNING, C. AND LATECKI, L. J. 2003. The choice of vantage objects for image retrieval. *Patt. Recogn. 36*, 9, 2187–219

HISTECRU, G. AND FARACH-COLTON, M. 1999. Cluster-preserving embeddings of proteins. Tech. rep., Rutgers University, Piscataway, NJ.

HJALTASON, G. AND SAMET, H. 2003. Properties of embedding methods for similarity searching in metric spaces. *Patt. Anal. Mach. Intell. 25*, 5, 530–549.

HRISTESCU, G. AND FARACH-COLTON, M. 1999. Cluster-preserving embedding of proteins. Tech. rep. 99-50, DIMACS 8.

KRUSKAL, J. AND WISH, M. 1978. *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA.

LATECKI, L. J., LAKAEMPER, R., AND ECKHARDT, U. 2000. Shape descriptors for non-rigid shapes with a single closed contour. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 424–429.

LINIAL, N., LONDON, E., AND RABINOVICH, Y. 1995. The geometry of graphs and some of its algorithmic applications. *Combinatorica 15*, 215–245.

MOKHTARIAN, F., ABBASI, S., AND KITTLER, J. 1996. Efficient and robust retrieval by shape content through curvature scale space. In *Proceedings of the British Machine and Vision Conference* (*BMVC'96*).

PEKALSKA, E., DUIN, R., AND PACLIK, P. 2005. Prototype selection for dissimilarity-based classifiers. In *Pattern Recognition*, Elsevier, Amsterdam, 189–208.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. 1998. A metric for distributions with applications to image databases. In *Proceedings of the IEEE 6th International Conference on Computer Vision* (*ICCV'98*). IEEE, Los Alamitos, CA, 59–88.

SAMET, H. 2006. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann.

SELLIS, T. K., ROUSSOPOULOS, N., AND FALOUTSOS, C. 1987. The r-tree: A dynamic index for multi-dimensional objects. In *Proceedings of the Conference on Very Large Databases (VLDB)*. 507–518.

TYPKE, R., GIANNOPOULOS, P., VELTKAMP, R. C., WIERING, F., AND VAN OOSTRUM, R. 2003. Using transportation distances for measuring melodic similarity. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 107–114.

VAN LEUKEN, R. H., VELTKAMP, R. C., AND TYPKE, R. 2006. Selecting vantage objects for similarity indexing. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*. 453–456.

VENKATESWARAN, J., LACHWANI, D., KAHVECI, T., AND JERMAINE, C. 2006. Reference-based indexing of sequence databases. In *Proceedings of the Conference on Very Large Databases (VLDB)*. 906–917.

VLEUGELS, J. AND VELTKAMP, R. C. 2002. Efficient image retrieval through vantage objects. In *Pattern Recognition*, 69–80.

WANG, X., WANG, J. T.-L., LIN, K.-I., SHASHA, D., SHAPIRO, B. A., AND ZHANG, K. 2000. An index structure for data mining and clustering. In *Knowledge and Information Systems*, 161–184.

YIANILOS, P. N. 1993. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York, 311–321.

Received July 2008; revised September 2009; accepted January 2010