Selecting vantage objects for similarity indexing

Reinier H. van Leuken, Remco C. Veltkamp, Rainer Typke
Institute of Information and Computing Sciences
Utrecht University, The Netherlands

Abstract

To make similarity searching in multimedia databases practical, indexing has become a necessity. Vantage indexing is an indexing technique which maps a dissimilarity space onto a vector space such that each object is represented by a vector of dissimilarities to a small set of m reference objects, the vantage objects. Querying takes place within this vector space, reducing the number of distance calculations to m. 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 and present experimental results based on data sets of different modality.

1. Introduction

Proximity searching in multimedia databases has gained more and more interest over the years. In particular searching in dissimilarity spaces (rather than extracting a feature vector for each database object) is an increasing area of research. With growing multimedia databases indexing has become a necessity. For instance, if a database contains 4 million objects and the evaluation of a complex distance measure that is defined on the objects takes half a second, querying by sequential search takes over 23 days per query.

Several indexing techniques have been proposed, and can be divided roughly in two categories. Traditional tree-based indexing techniques exploit the benefits of tree-like structures. An extensive survey of these methods is given by Chavez et al. [4]. Embedding techniques embed the dissimilarity space in a vector space, on which a distance measure can be defined. Examples of these strategies are BoostMap [1], SparseMap [9] and FastMap [5]. See the survey by Hjaltason and Samet [8].

The approach discussed here falls in the second category and uses so-called *vantage objects* to create the embedding [13]. Each database object is represented by a vector of distances to the vantage objects. See Section 2 for a more detailed discussion of the vantage indexing structure. The performance of a retrieval system that uses vantage indexing is influenced by the choice of vantage objects. This paper presents a method for selecting good vantage objects.

Recently, Pękalska et al. have investigated a related problem [11]. 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. [3] 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 [7]. 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.

1.1 Our contributions

Firstly, we propose a novel approach for the selection of vantage objects, based on criteria that are directly concerned with the retrieval performance, namely the minimization of the number of false positives in the returned sets. Secondly, each object in the database is a candidate vantage object, no random pre-selection is made. Thirdly, we have done experiments using two data sets of different modality: the MPEG-7 CE-Shape-1 part B set, consisting of 1400 shape images, and a subset of the RISM-A/II data set consisting

of 6300 fragments of music. We have compared our method to three other methods: random selection, the loss-based selection method and the MaxMin method. Our method improves over these methods.

2 Vantage indexing

Vantage indexing works as follows: given a multimedia database A and a distance measure $d : A \times A \rightarrow \mathbb{R}$, select from the database a set of m objects $A^* = \{A_1^*, ...A_m^*\}$, the so called vantage objects. Compute the distance from each database object A_i to each vantage object, 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.

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 used on the points in vantage space is L_∞.

Vleugels and Veltkamp show [13] that as long as the triangle inequality holds for the distance measure d defined on the database objects, recall (ratio of number of relevant retrieved objects to the total number of relevant objects in the whole data base) is 100%, meaning that there are no false negatives. However, false positives are not excluded from the querying results, so precision (ratio of number of relevant retrieved objects to the total number of retrieved objects) is not necessarily 100%. We claim that by choosing the right vantage objects, precision can increase significantly.

3 Selecting vantage objects

The retrieval performance of a vantage index can improve significantly with a proper choice of vantage objects. This improvement is measured in terms of false positives, as defined below. Let δ be the distance measure in vantage space.

Definition 1 Return set *Given $\epsilon > 0$ and query A_q, object A_i is included in the return set of A_q if and only if $\delta(A_q, A_i) \leq \epsilon$.*

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

We present a new technique for selecting vantage objects that is based on two criteria which address the number of false positives in the retrieval results directly. The first criterion (spacing) concerns the relevance of a single vantage object, the second criterion (correlation) deals with the redundancy of a vantage object with respect to the other vantage objects. We call this method Spacing-based Selection.

The main idea is to keep the number of objects that are returned for a query A_q and range ϵ low. Since false negatives are not possible under the condition that the triangle inequality holds for d, minimization of the number of false positives is achieved by spreading out the database along the vantage space as much as possible. False positives are, intuitively speaking, pushed out of the returned sets.

3.1 Spacing

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

A priori the query object A_q is unknown, so the distance $d(A_q, V_j)$ between a certain query A_q and vantage object V_j is unknown. The size of the range query (ϵ) is unknown beforehand as well. Optimal performance (achieved by small return sets given a query A_q and range ϵ) should therefore be scored over all possible queries and all possible ranges ϵ.

This is achieved by avoiding clusters on the vantage axis belonging to V_j. Our first criterion therefore concerns the *spacing* between objects on a single vantage axis, which is defined as follows:

Definition 3 *The spacing 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 is $\frac{1}{n-1}\sum_{i=1}^{n-1}((d(A_{i+1}, V_j) - d(A_i, V_j)) - \mu)^2$. 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.

3.2 Correlation

It is not sufficient to just select relevant vantage objects, they also should be non-redundant. A low variance of spacing does not guarantee that the database is well spread out in vantage space, since the vantage axes might be strongly correlated.

Therefore, we compute all linear correlation coefficients for all pairs of vantage objects and make sure these coefficients do not exceed a certain threshold. Experiments show that on the MPEG-7 shape images set pairwise correlation is sufficient and that higher order correlations are not an issue.

3.3 Algorithm

Spacing-based Selection selects a set of vantage objects according to the criteria defined above with a randomized

Figure 1. Examples of the MPEG-7 data set.

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, thus keeping the amount of work that has to be redone small. For details, see Algorithm 1. For the time complexity, see Section 5.

Algorithm 1 Spacing-based Selection
Input: Database A with objects $A_1, ..., A_n$, $d(A, A) \rightarrow \mathbb{R}$, thresholds ϵ_{corr} and ϵ_{spac}
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 do in random order
    for All objects V_j do
        compute d(A_i, V_j)
        add A_i to index
        if var(Spacing)(V_j) > ε_spac then
            remove V_j
            select new vantage object randomly
    if for any pair p(V_k, V_l), Corr(V_k, V_l) > ε_corr then
        remove p's worst spaced object
        select new vantage object randomly
```

4 Experimental results

We implemented our algorithm and tested it on two data sets of different modality: one data set of 1400 shape images and one set of 6300 fragments of music notation. By using different kinds of data we emphasize the domain independence of our method.

Shape retrieval. We used the MPEG-7 test set CE-Shape-1 part B, consisting of 1400 shape images, contained in 70 classes of 20 images per class. A few examples are given in Figure 1.

The distance measure used to calculate the distance between two of these shape images is the Curvature Scale Space (CSS) [10]. 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.

Figure 2. MPEG-7: false positive ratios.

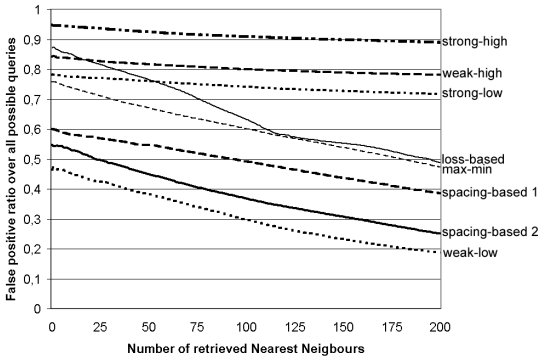

Table 1. MPEG-7: False positive ratios and average precision

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

Two shapes are now matched by comparing the peaks in their CSS-images.

To justify our 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 2. 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 2, 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 shows similar results for all 1400 queries, retrieving 100 nearest neighbors for each query. The left column in this table lists false positive ratios averaged over all 1400 queries, and averaged over a large number of selected sets of vantage objects. The right column shows average precision numbers. For some applications, 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 also by means of average precision: the mean of the precision scores obtained after each true positive is retrieved [2]. A maximum average precision score of 1.0 is obtained when all true positives are at the top of the retrieval ranking.

Music retrieval. We have compared Spacing-based Selection to random selection on a data set of 6300 chunks of five notes from a collection of music. These notes are viewed as points in a space in which the pitch and onset time are the axes [12]. The distance between two fragments is computed using the Proportional Transportation Distance [6], a modified version of the Earth Mover's Distance such that the triangle inequality holds.

Averaged over 200 queries, the ratio of false positives for random selection is 0.25. However, for Spacing-based Selection this ratio is 0.12. This performance was achieved by range-searching with a rather small search radius, returning on average 4 to 5 objects per query. With an increasing search radius, the number of false positives quickly dominates the false positives ratio for both methods. The difference in performance between random selection and Spacing-based Selection is then small. We conjecture that this is due to the fact that the size of the return set increases exponentially with an increasing search radius. Together with a possibly small number of true positives, this distorts results with larger search radii.

More results will be presented in a full version of this paper.

5 Concluding remarks

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 (8 in our experiments).

The main reason the loss-based method performs similarly to random selection is that it evaluates one nearest neighbor in vantage space. When retrieving more nearest neighbors, high performance is no longer guaranteed.

Future work will be to search automatically for an optimal number of vantage objects. In general, more vantage objects result in fewer false positives, but the index size increases resulting in longer querying times. During execution of the selection algorithm, the benefit of an extra vantage object might be evaluated to find a proper dimensionality given a data set and application.

Acknowledgements. This research was supported by the FP6 IST project 511572-2 PROFI.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: a method for effiecient approximate similarity rankings. In *IEEE Computer Vision and Pattern Recognition*, 2004.

[2] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Research and Development in Information Retrieval*, pages 33–40, 2000.

[3] B. Bustos, G. Navarro, and E. Chavez. Pivot selection techniques for proximity searching in metric spaces. *Pattern Recogn. Lett.*, pages 2357–2366, 2003.

[4] E.Chavez and G. Navarro. Searching in metric spaces. *ACM Computer Surveys*, pages 273–321, 2001.

[5] 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 '95*, pages 163–174.

[6] P. Giannopoulos and R. C. Veltkamp. A Pseudo-Metric for Weighted Point Sets. In *Proceedings of ECCV*, pages 715–730, 2002.

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

[8] G. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. In *PAMI*, pages 530–549, 2003.

[9] G. Hristescu and M. Farach-Colton. Cluster-preserving embedding of proteins. Technical Report 99-50, DIMACS, 8, 1999.

[10] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Proceedings of IDB-MMS'96*, pages 35–42.

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

[12] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In *Proceedings of ISMIR*, pages 107–114, 2003.

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