

Dimensionality Reduction using Magnitude and Shape Approximations

Ümit Y. Ogras, Hakan Ferhatosmanoglu
The Ohio State University
Department of Computer and Information Science
{ogras,hakan}@cis.ohio-state.edu

ABSTRACT

High dimensional data sets are encountered in many modern database applications. The usual approach is to construct a summary of the data set through a lossy compression technique, and use this lower dimensional synopsis to provide fast, approximate answers to the queries. In this paper, we develop a novel dimensionality reduction technique based on partitioning the high dimensional vector space into orthogonal subspaces. First, we find a relation between the Euclidian distance of two n -dimensional vectors and the Euclidian distances of their projections on the orthogonal subspaces. Then, based on this relation we develop a method to approximate the Euclidian distance using novel inner product approximation. This process allows us to incorporate the shape information of the vectors to this approximation. While the inner product approximation is symmetric, i.e., captures only the magnitude information of the data, the proposed method takes both the magnitude and shape information of the original vectors into account through partitioning. In the experiments, we demonstrate the effectiveness of our technique by comparing it with commonly used methods.

Categories and Subject Descriptors:H.3.3[Information Search and Retrieval]: Indexing Methods

General Terms: Performance, Design, Experimentation, Theory, Algorithms.

Keywords: High Dimensional Data, Similarity Search, Shape Approximation.

1. INTRODUCTION

With the deployment of different types of information in large-scale applications, both the dimensionality and the amount of data that needs to be processed are increasing rapidly. Some examples of such applications are document databases [11], medical imaging [31], and multimedia information systems [17,37]. The general approach is to generate feature vectors that represent the original data objects and

*This work was partially supported by DOE Career Award DE-FG02-03ER25573

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.
Copyright 2003 ACM 1-58113-723-0/03/0011 ...\$5.00.

then to define a distance metric, e.g. Euclidian distance between these vectors that represents the (dis)similarity between the objects. A popular type of query is similarity query which is defined as finding the most similar data objects in the data set to a given query object. For example, in image databases a possible similarity query is to find the images most similar to a given image. The images are represented as d dimensional feature vectors, e.g., color, shape, texture, features, and the similarity between the images is defined by a distance function, e.g., Euclidean distance between the corresponding feature vectors. The k -nearest neighbor, k -NN, problem is defined as finding the k most similar feature vectors to a query point q . A closely related query is the ϵ -range query where all feature vectors that are within ϵ neighborhood of the query point q are retrieved.

For efficient query processing in multi-dimensional spaces, a number of index structures have been developed. Techniques for low-to-medium dimensional data sets are already incorporated into commercial database products [23,27], and image, scientific and medical applications [3,6,20,31,39,42]. However, neither the common multi-dimensional indexing techniques [21,36] nor their extensions [5,8,10,19,29,32] can be successfully scaled for very high dimensional data because of the effects of the infamous dimensionality problems [4,41]. It has been noted that as dimensionality increases, query performance in most of the current approaches degrades significantly [4]. This anomaly is referred as the dimensionality curse [18] and has attracted the attention of several researchers.

A popular solution to the problem of dimensionality curse is dimensionality reduction for scalable query performance. The most common approaches found in the literature for dimensionality reduction are linear-algebraic methods such as the Karhunen-Loeve Transformation (KLT) [24,28,33], or applications of mathematical transforms such as the Discrete Fourier Transform (DFT) [35], Discrete Cosine Transform (DCT) [25], or Wavelet Transform (DWT) [9]. As the transformations are known to be distance preserving, the general approach is to transform the high dimensional feature vectors and obtain lower dimensional vectors by taking the first few leading coefficients of the transformed vectors [2]. The general idea of these techniques depends on the observation that by using these transformations, a small subset of dimensions keeps a large portion of the information about the feature vectors. Several reduction techniques were proposed for time-series [2,20,30], image [17,26,34,43], and document data [11–13]. And recently a nonlinear dimensionality reduction was proposed in [40].

If the distance between the transformed vectors is a lower bound to the distance between the original feature vectors, then the lower-bound filtering property is said to hold [38]. When the lower-bound filtering property holds, a point in an ϵ neighborhood of the query point stays in an ϵ neighborhood in the transformed domain. On the other hand, since the query result is underestimated, the returned result set may contain extra data. These false hits can then be eliminated by checking the original distance between the feature vectors. Therefore, at the end the query result has only the data in the ϵ -neighborhood of the query point.

Approximation methods in similarity queries have also attracted attention [22,26]. It can be argued that approximate methods achieve efficiency at the expense of exact results. However exact results are difficult to obtain in several applications to begin with. One reason is that the generation of feature vectors from the original objects itself may be based on heuristics. Besides, the semantics expected from most application domains are not as strict as the exact queries used in relational databases. For example, the QBIC project at IBM provides the ability to run queries based on colors, shapes, and sketches [17,34]. As mentioned in the Asilomar Report on Database Research [7], imprecise information will not only appear as the output of queries, it already appears in data sources as well. For several applications, it is much more reasonable to define approximate queries; consider a user submitting a query such as ‘‘Are there any good Italian restaurants close to where I live?’’. There is no exact answer to this query since it is difficult to give a perfect definition of goodness and even closeness. In such instances it is useful to provide an approximate answer to the given query. Another reason for developing approximation techniques is the huge amount of data in a typical application for which exact answers would require a long period of time to execute. For example, the number of documents that can be reached by internet is increasing rapidly, the commercial data warehouses are doubling their sizes every 9-12 months, and satellite data repositories will soon add one to two terabytes of data every day [1]. If the current trends continue, large organizations will have petabytes of data that need to be processed [7]. However for approximation queries a careful analysis must be given to the approximation quality of the dimensionality reduction technique. It is essential to develop techniques which have accurate approximations to the original similarity distance so that the similar objects in the original domain remain similar in the transformed domain. The dimensionality reduction technique proposed in this paper considers both magnitude and shape of the original data and directly aims minimizing the error made in the approximations.

The remaining sections are organized as follows. In Section 2, we review the inner product approximation and describe how to use this approach to obtain fast approximation to the Euclidian distance. The drawback of the method and an effective enhancement to overcome this drawback is discussed in Section 3. In the same section, we also provide some illustrative examples to show the need for the shape approximation in addition to the magnitude approximation discussed in Section 2. In Section 4, we perform experiments to justify the results obtained in the preceding sections. The experiments clearly show that the incorporation of the shape approximation enhances the performance of the symmetric magnitude approximation. Furthermore, the

proposed method is compared with other methods and its superiority is demonstrated. Conclusions and future work appear in Section 5.

2. MAGNITUDE APPROXIMATION

Developing effective ways for dimensionality reduction is crucial for the query performance in modern databases. In applications, where the similarity is measured by the distances between multidimensional data points, it is required that the distance in the lower dimensional space closely approximates the original distance. Thus, it is desirable to develop dimensionality reduction techniques that achieve accurate approximations to the original similarity distance. In this section, we review our approach to generate small size synopsis of high dimensional data and then show how to utilize this method for similarity search.

Dimensionality reduction and similarity search based on inner product approximation is first introduced in [15]. In this technique, the p -th power symmetric function of a sequence $x = (x_1, x_2, \dots, x_n)$ is defined by

$$\psi_p(x) = x_1^p + x_2^p + \dots + x_n^p. \quad (1)$$

Note that, $\psi_p(x)$ is equivalently the p -th power of the p -norm $\|x\|_p$ which is defined as

$$\|x\|_p = \sqrt[p]{x_1^p + x_2^p + \dots + x_n^p}$$

Hence, we can write $\psi_p(x) = \|x\|_p^p$. Using this relation, we can express the ordinary Euclidean distance between x and y in terms of the power symmetric functions as

$$\|x - y\|_2 = \sqrt{\psi_2(x) + \psi_2(y) - 2 \langle x, y \rangle} \quad (2)$$

where $\langle x, y \rangle$ is the standard inner-product given by

$$\langle x, y \rangle = x_1y_1 + x_2y_2 + \dots + x_ny_n$$

We assume that x is a vector from a massive data set, say $V^{m \times n}$, consisting a huge number of n -dimensional vectors. The query vector, y , can be from the same data set or from a similar distribution. Note that, the ψ_2 values for each vector $x \in V$ can be pre-computed as the representative of the original vector x and stored. If we can find an approximation for the inner product in Equation 2 in terms of $\psi_2(x)$, we can use these ψ_2 values as a lower dimensional representative of the original data. The Cauchy-Schwarz inequality below provides an upper bound for the inner product

$$\langle x, y \rangle \leq \|x\|_2 \|y\|_2. \quad (3)$$

If we substitute this expression to Equation 2, we can obtain a lower bound to the Euclidian distance. The Cauchy-Schwarz inequality provides a first order approximation to the inner product. However, we want the answer to be accurate, i.e. we aim to minimize the error made in the distance computations. Hence, we look for an higher order approximation of inner product in terms of the quantities $\psi_p(x)$ for each data vector x in the database V . In [15], it is shown that the inner product of the sequences can be approximated as

$$\langle x, y \rangle^m \approx b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y) + \dots + b_m\psi_m(x)\psi_m(y) \quad (4)$$

for large n . Given that the components of x are independently drawn from a common (possibly unknown) distribution $F(t)$ with density $f(t)$, the optimal coefficients

m	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
2	$-\frac{1}{16}$	$\frac{45}{64}$						
3	$-\frac{5}{16}n$	$\frac{3}{2}n$	$-\frac{7}{6}n$					
4	$-\frac{59}{256}n^2$	$\frac{1575}{1024}n^2$	$-\frac{175}{64}n^2$	$\frac{1575}{1024}n^2$				
5	$-\frac{31}{256}n^3$	$\frac{9}{8}n^3$	$-\frac{27}{8}n^3$	$\frac{135}{32}n^3$	$-\frac{297}{160}n^3$			
6	$-\frac{4096}{221}n^4$	$\frac{11025}{16384}n^4$	$-\frac{6125}{2048}n^4$	$\frac{202125}{32768}n^4$	$-\frac{24255}{4096}n^4$	$\frac{35035}{16384}n^4$		
7	$-\frac{89}{4096}n^5$	$\frac{45}{128}n^5$	$-\frac{275}{128}n^5$	$\frac{825}{128}n^5$	$-\frac{1287}{128}n^5$	$\frac{1001}{128}n^5$	$-\frac{2145}{896}n^5$	
8	$-\frac{535}{65536}n^6$	$\frac{43659}{262144}n^6$	$-\frac{43659}{32768}n^6$	$\frac{2837835}{524288}n^6$	$-\frac{3972969}{327680}n^6$	$\frac{3972969}{262144}n^6$	$-\frac{81081}{8192}n^6$	$\frac{1378377}{524288}n^6$

Figure 1: $\langle x, y \rangle^m \approx b_1\psi_1(x)\psi_1(y) + \dots + b_m\psi_m(x)\psi_m(y)$: asymptotic expansion coefficients b_1, \dots, b_m for the uniform distribution.

b_1, b_2, \dots, b_m in the sense of least squares can be found independently of x and y . That is, we look for the constants that minimize

$$\int \left[\langle x, y \rangle^m - \sum_{j=1}^m b_j \psi_j(x) \psi_j(y) \right]^2 dx dy \quad (5)$$

where $dx = dx_1 dx_2 \dots dx_n$, $dy = dy_1 dy_2 \dots dy_n$, and the integral is over the $2n$ -dimensional unit cube I^{2n} . The normal equations that b_1, b_2, \dots, b_m must satisfy are found by differentiating (5) with respect to each b_i , and setting the resulting expressions to zero. Reader can refer to [15] for the details of the computations where the best set of coefficients for various distributions such as uniform, normal, exponential, binomial, poisson, beta and for non-parametric case are found. Figure 1 shows the asymptotic expansion coefficients b_1, b_2, \dots, b_m for the uniform distribution, and Figure 2 provides optimal coefficients b_1, b_2 for various parametric distributions.

Unknown Distribution If the coordinates of each vector x are drawn from a known parametric distribution family, then the parameters can be computed as mentioned before. However, the distribution of a real data set may be non-parametric or unknown to the user. Therefore, we need a method to estimate the best set of parameters for real data sets. Suppose we have a data set consisting of r n -dimensional vectors and we know the moments, μ_i of its density function. If x is added as the $(r+1)^{th}$ vector to the data set, then the new estimate of the moments can be obtained in terms of the previous moments as

$$\bar{\mu}_i[r+1] = \frac{r}{r+1} \bar{\mu}_i[r] + \frac{n - \psi_{i+1}(x)}{n(r+1)(i+1)} \quad (6)$$

Hence, this relation can be used to compute and maintain empirical moments of a density based on the sample points. Furthermore, we can utilize first four moments of the density function to compute the optimum coefficients using

$$\begin{aligned} b_1 &= \mu_1^2 \cdot \frac{2\mu_2^3 + \mu_1^2\mu_4 - 3\mu_1\mu_2\mu_3}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}, \\ b_2 &= \frac{\mu_1^4}{\mu_2} \cdot \frac{\mu_1\mu_3 - \mu_2^2}{\mu_2^3 - \mu_1^2\mu_4 - 2\mu_1\mu_2\mu_3}. \end{aligned} \quad (7)$$

Equations 6 and 7 provide the tools to compute the optimum coefficients for real data sets. The derivations of these results can be found in [16].

3. INCORPORATING SHAPE

The symmetry in the power functions is a drawback of p -norms approximation, which causes biased results in some cases. For example, consider the following two vectors $x =$

$[1, 2, 3, 4, 5]$, $z = [5, 4, 3, 2, 1]$ and any arbitrary query y . Note that,

$$\psi_1(x) = \psi_1(z), \psi_2(x) = \psi_2(z), \dots, \psi_m(x) = \psi_m(z)$$

Since we sum the p -th powers of the entries during the construction of the power symmetric functions, the indices of the entries do not effect the result. Hence, the information about the shape of the vectors is lost. The presence of this information in the original Euclidian distance is clear, since the differences of the entries corresponding to the same index are found and squared. Therefore, when the shape of one of the sequences is distorted (this could be flipping the sequence as in the example above, or interchanging some of the entries of the sequence), this is directly reflected to the Euclidian distance. While the power functions ψ_1 and ψ_2 in Equation 2 are symmetric, independent of the location of the entries, the inner product is dependent on the shape of the sequences. As a result, by approximating the inner product by a symmetric expression shape information is lost. Since the p -norms method already produces very close approximations to the actual Euclidian distance, it is reasonable to think that combining it with shape approximation will result in even higher performance. In this section, we introduce a method for obtaining combined magnitude and shape approximations. For real data sets, the entries of a vector closer to each other are expected to be more correlated than those that are farther from each other. For example, in a time series data originating from a sensor reporting temperature, in a stock market data or in a series reporting the traffic load of a particular network the values corresponding to a certain time period are more correlated than the values for arbitrary time instances. Hence, instead of using the exact location of each entry, as in exact Euclidian approximation, we can partition the vectors into non-overlapping portions and capture part of the shape information. The following mathematical derivation is inspired by this observation.

Let us divide the N -dimensional space into k orthogonal subspaces denoted by $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$. Define the B_l and Z_l as l -dimensional vectors consisting of all 1's and 0's, respectively, i.e.

$$B_l = [1, 1, \dots, 1]_{1 \times l} \text{ and } Z_l = [0, 0, \dots, 0]_{1 \times l}$$

Based on these definitions, each subspace can be expressed as

$$\begin{aligned} \mathcal{S}_1 &= [B_l \ Z_{N-l}] \\ \mathcal{S}_2 &= [Z_l \ B_l \ Z_{N-2l}] \\ \mathcal{S}_3 &= [Z_{2l} \ B_l \ Z_{N-3l}] \\ &\vdots \\ \mathcal{S}_k &= [Z_{N-l} \ B_l] \end{aligned}$$

Distribution	Density $f(x)$	Range	b_1	b_2
Uniform	1	$0 \leq x \leq 1$	$-\frac{1}{16}$	$\frac{45}{64}$
Power	cx^{c-1}	$0 \leq x \leq 1$	$-\frac{2c^3}{(c+1)^2(c^2+3c+4)}$	$\frac{c^2(c+2)^2(c+4)}{(c+1)^2(c^2+3c+4)}$
Exponential	$(1/b)\exp(-x/b)$	$0 \leq x \leq \infty$	$\frac{b^2}{2}$	$\frac{1}{8}$
Binomial	$\binom{N}{x}p^xq^{N-x}$	$0 \leq x \leq N$	$\frac{N^2p^2(1-2p)}{np-3p+2}$	$\frac{N^2p^2}{(np-p+1)(np-3p+2)}$
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$	$-\infty \leq x \leq \infty$	$\frac{2\mu^2\sigma^4}{\mu^4+\sigma^4}$	$\frac{\mu^4(\mu^2-\sigma^2)}{(\mu^2+\sigma^2)(\mu^4+\sigma^4)}$
Poisson	$\lambda^x \exp(-\lambda)/x!$	$0 \leq x \leq \infty$	$\frac{\lambda^2}{\lambda+2}$	$\frac{\lambda}{(\lambda+2)(\lambda+1)}$
Beta	$\frac{(v+w-1)!x^{v-1}(1-x)^{w-1}}{(v-1)!(w-1)!}$	$0 \leq x \leq 1$	$\frac{2v^2(w-v-1)}{(v+w)^2((v+1)^2+(v+3)w)}$	$\frac{v^2(w+v+1)^2(w+v+3)}{(v+w)^2((v+1)^3+(v+1)(v+3)w)}$

Figure 2: $\langle x, y \rangle^2 \approx b_1\psi_1(x)\psi_1(y) + b_2\psi_2(x)\psi_2(y)$: optimal asymptotic expansion coefficients b_1, b_2 for various parametric distributions.

where $l = \frac{N}{k}$. Furthermore, let us take the projections of the vectors x and y on to each of these subspaces as

$$\begin{aligned} XP_i &= \langle x, \mathbb{S}_i \rangle \\ YP_i &= \langle y, \mathbb{S}_i \rangle \quad i = 1, 2, \dots, k \end{aligned}$$

Using the projections defined above, the Euclidian distance between x and y can be written as

$$\begin{aligned} \|x - y\|^2 &= \|XP_1 + XP_2 + \dots + XP_k - YP_1 - YP_2 - \dots - YP_k\|^2 \\ \|x - y\|^2 &= \|(XP_1 - YP_1) + (XP_2 - YP_2) + \dots + (XP_k - YP_k)\|^2 \end{aligned} \quad (8)$$

Note that the projections

$$\{XP_1, XP_2, \dots, XP_k\} \text{ and } \{YP_1, YP_2, \dots, YP_k\}$$

constitute orthogonal sets. Consequently, the differences $(XP_i - YP_i)$ $i = 1, 2, \dots, k$, are orthogonal. As a result, we can rewrite Equation 8 as

$$\|x - y\|^2 = \|XP_1 - YP_1\|^2 + \|XP_2 - YP_2\|^2 + \dots + \|XP_k - YP_k\|^2 \quad (9)$$

Equation 9 is exact computation of Euclidian distance. At this stage, we compute each of the Euclidian distances in this Equation using the p -norms approximation given in Equation 4. As stated before, this process causes the shape information to be lost. However, since we are using it to approximate the Euclidian distance between localized sub-vectors of the original vectors, we still capture global shape information. This phenomena will be further illustrated in Section 3.1. Let us call these approximate results as DP_i , where

$$DP_i \approx \|XP_i - YP_i\|^2 \quad i = 1, 2, \dots, k$$

Then, the approximate Euclidian distance between x and y that combines both magnitude and shape approximations is given by

$$\|x - y\|^2 \approx DP_1^2 + DP_2^2 + \dots + DP_k^2 \quad (10)$$

In the following section, we provide the intuition behind the proposed method by giving examples.

3.1 Illustration of Incorporating Shape

In Section 2, we described symmetric Euclidian distance approximation. Later, in Section 3 we proposed a method to incorporate the shape information into this technique. In this part, we illustrate these ideas with an example and demonstrate how the approximation is enhanced.

The intuition behind the Euclidian distance is studied in [30]. Here, we will consider two simple sequences and illustrate the computation of the Euclidian distance between them using exact expression (Equation 2), using symmetric approximation of inner product (Equation 4), and adding shape information to the symmetric approximation (Equation 10). The process is illustrated in figures 3 and 4. For

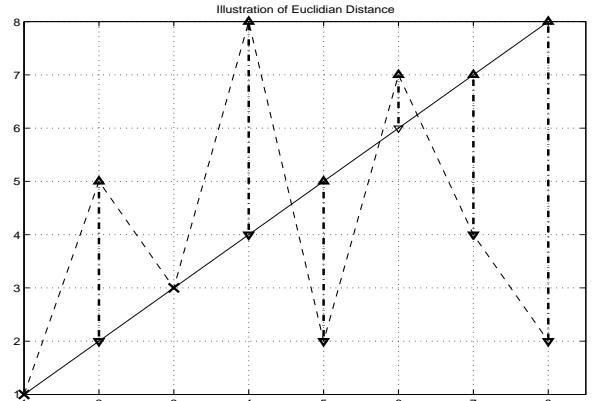
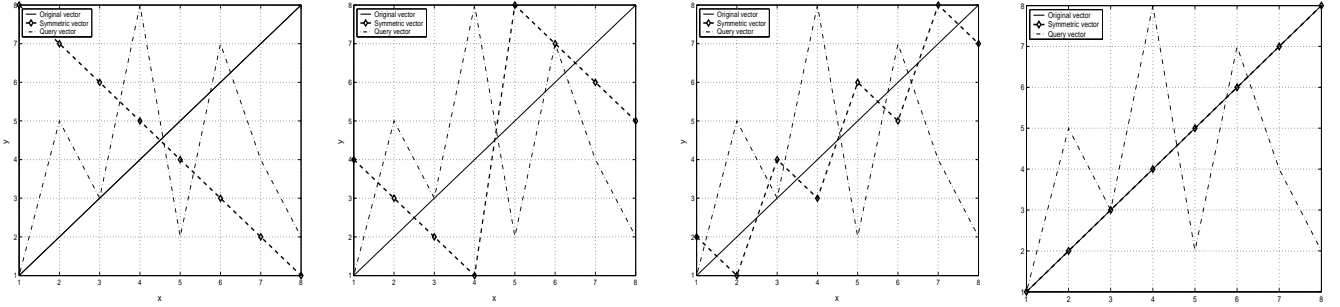


Figure 3: - : x , -- : y . The Euclidian Distance is illustrated by the arrows.

illustration purposes, the first sequence is selected simply as $x = [1, 2, 3, 4, 5, 6, 7, 8]$ and the second sequence is selected arbitrarily as $y = [1, 5, 3, 8, 2, 7, 4, 2]$. x and y are plotted in Figure 3. The differences between the entries of x and y corresponding to the same indices are illustrated by the arrows. The Euclidian distance is found by summing the squared norms of these arrows. A symmetrical vector to x in the sense of power functions (Equation 1) is any vector having the same entries with an arbitrary order, because in the calculation of p -th power functions, the location of the entries is not important. For example, $x_1 = [3, 5, 2, 8, 6, 4, 5, 7]$ has the same $\psi_1, \psi_2, \dots, \psi_m$ values with x , although the shapes are quite different. Egecioglu [14] observes this fact and decomposes an n -dimensional vector x into two pairs as $(s(x), \sigma(x))$ to account for this fact. $s(x)$ is the sorted version of x into weakly increasing coordinates, and $\sigma(x)$ is the permutation of indices. In this decomposition, $s(x)$ captures the magnitude information, while $\sigma(x)$ preserves the shape information. Then, the Euclidian distance is approximated by convex combination of two measures. The performance of this method is compared with our results in the final section.



(a) Due to the symmetry in the approximation the original vector and its flipped version cannot be distinguished

(b) The vectors are divided into 2 orthogonal components. The resulting symmetric vector is closer to the original one

(c) The vectors are divided into 4 orthogonal components. The shape of the symmetric vector approaches further to the original one

(d) The vectors are divided into 8 orthogonal components. The symmetric vector coincides with the original vector

Figure 4: The effect of incorporating shape information is illustrated for various partitions.

We analyze fully symmetric approximation of Euclidian distance in Figure 4(a). In addition to x and y , we also plotted the flipped version of x , call x_f , in this figure. Note that, x and x_f are symmetric, hence they have same power functions. If we use only the magnitude information, the approximate Euclidian distance of y to x and x_f would be found the equal to each other. To see the effect of adding shape information, let us divide the subspace into two orthogonal subspace as explained in Section 3. The projections of x onto these subspaces are given as $x_{p1} = [1, 2, 3, 4, 0, 0, 0, 0]$ and $x_{p2} = [0, 0, 0, 0, 5, 6, 7, 8]$. We also partition y in the same way and then, compute the symmetric approximation of the vectors in each subspace to find the overall approximation using Equation 10. This process is illustrated in Figure 4(b). It can be observed that the symmetrical vectors in the subspaces are closer to the original vector for this case. Hence, part of the shape information is captured. The effectiveness of the method can be seen better, if we increase the number of partitions. In Figure 4(c), we divided the n dimensional space into four subspaces. It is clearly seen that the ambiguity in the shape decreases considerably, hence the quality of the approximation increases. In the extreme case, dividing the space into 8 partitions results in exact computation of the Euclidian distance. In this case, there is no ambiguity as seen in Figure 4(d). The incorporation of shape information increases the quality of the distance approximation at the expense of storage space. If we employ only magnitude approximation, we need to store m components, $\psi_1(x), \dots, \psi_m(x)$, for each high dimensional vector x in the data set. On the other hand, including shape information by partitioning the n -dimensional space into k orthogonal subspaces requires storing $(k \times m)$ ψ values, i.e. m components for each of the k subspaces. Note that p -norm approximation given in Equation 4 is a special case of the proposed approximation when $k = 1$. By increasing k , we localize part of the information and thus, refine the approximation. What is more important, we enhance the selectivity of the technique. That is, inclusion of the shape information enables us to distinguish vectors that cannot be distinguished by the p -norm approximation alone as illustrated in Figure 4. Let us assume that we have a query of length n , which is not in the database. It is possible to find $n! - 1$ different

vectors having the same power functions. Since the difference between them is in their shape, this information is lost during the approximation. This means symmetric approximation cannot distinguish between these vectors. If at least one of these $n! - 1$ vectors is in the database, this method will give wrong result for the point query. Similarly, the fact that $n!$ different vectors result in the same power functions causes an ambiguity, and degrades the performance for $k - nm$ and range queries. If we partition the space into 2 orthogonal spaces, and apply the p -norm approximation the number of vectors with the same power functions drops to $(n/2)! \times (n/2)!$. So, there is a huge reduction in the ambiguity. In general, if we use k partitions, this number drops to $((n/k)!)^k$. Hence, the selectivity property enhances considerably.

Number of partitions It is clear that there is a trade-off between the accuracy and the number of subspaces, i.e. summary size. We can enhance the accuracy of the results by increasing the number of partitions at the expense of space, in the limit we would reach exact euclidian distance by using n partitions. We modelled this trade-off as

$$J = bk + \frac{\delta n}{k} \quad (11)$$

The first term is the summary size, if we use k partitions and represent each of them with b coefficients. The second term determines the number of symmetrical vectors, if we use k partitions. δ relates the number of symmetrical vectors to the refinement obtained in the accuracy. The minimum of J can be found as

$$k_{opt} = \sqrt{\frac{\delta n}{b}}$$

In our experiments, it is found that $\delta = 0.1$ gives satisfactory results.

4. EXPERIMENTS

p -norms approach provides a fast and accurate approximation to the Euclidian distance. However, the symmetry in the computation of the power functions $\psi_1, \psi_2, \dots, \psi_m$ may degrade the performance of the method in query processing. This deficiency can be solved if we incorporate information

about the shape of the vectors into the approximation as discussed in Section 3. In this section, we validate the performance of the proposed technique following a systematic approach. First, we validate the observation mentioned in the preceding paragraph using real data sets. Specifically, we explore the effect of incorporating shape information to the symmetric p -norms approximation by comparing number of false hits obtained for $k - nn$ search. Then, we show that the proposed technique produces accurate approximation to Euclidian distance by analyzing the approximation quality. The remaining two sets of experiments are devoted for the comparison of our technique with the existing methods. We evaluate the query performance of our technique for $k - nn$ search using two different data sets and provide comparison with the existing methods. Next, we compare our technique with the approximation developed in [14], which was also discussed in Section 3.

Comparison with symmetric p -norms approximation

In this part, we analyze the gain obtained by incorporating shape information to the symmetric p -norms approximation. In addition to the stock market data used for the previous experiment, we use wireless telephony data of size 64×1000 . For the real data sets, the optimum coefficients are found by experimentally estimating the moments of the data set as discussed in Section 2.

In this experiment, we first computed the actual pairwise distances and found k nearest neighbors of each vector. Then, the dimensionality is reduced using symmetric p -norms approximation and proposed approach. For each of these cases, we compute the approximate Euclidian distances and find the k nearest neighbors. Finally, we compare the number of false hit obtained for each of these cases. The results obtained using stock market data are shown in Figures 5. For the proposed approximation, the data set is partitioned into 4 orthogonal subspaces, and the distances in each subspace is approximated using $m = 2$, and same space is devoted to the symmetric approximation. It is observed that the performance of the magnitude approximation is enhanced by incorporating shape approximation. The number of false hits caused by proposed method is always smaller than the p -norms approximation alone, as expected. Furthermore, the difference increases as the number of nearest neighbors we are looking increases. Experiment with the other real data set gave similar results.

Error performance We begin our experimental analysis by justifying that the proposed method results in accurate Euclidian distance approximation. We also demonstrate the superiority of this method in approximating Euclidian distance over widely used methods such as KLT, DFT and DCT.

In this experiment we used two different data sets, one from a uniform distribution in the interval $(0, 1)$, and the other from a exponential distribution with parameter 1. For each case, the data set consists of 500 vectors with dimension, n , varying from 16 to 1024. We first calculate the exact distances between each pair of vectors in the original data set. Then, we obtain the reduced dimensional vectors employing SVD, DFT, DCT, and proposed technique. That is, for SVD, DFT and DCT, we transformed the data and retained the leading coefficients as the lower dimensional representative. And for proposed technique, we partitioned the space into orthogonal subspaces, calculated $\psi_1(x), \dots, \psi_m(x)$, for each vector in the data set and stored

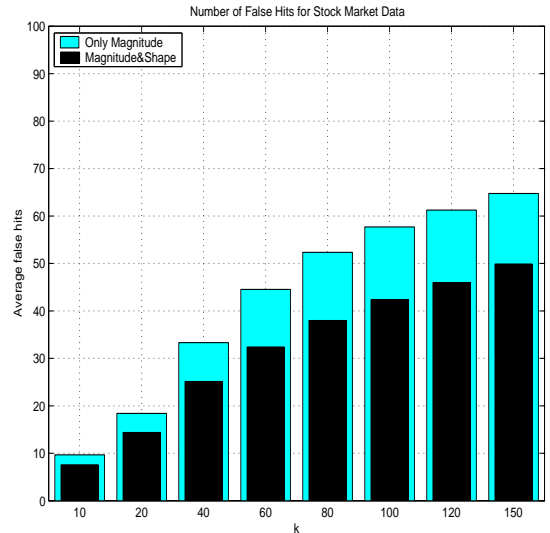


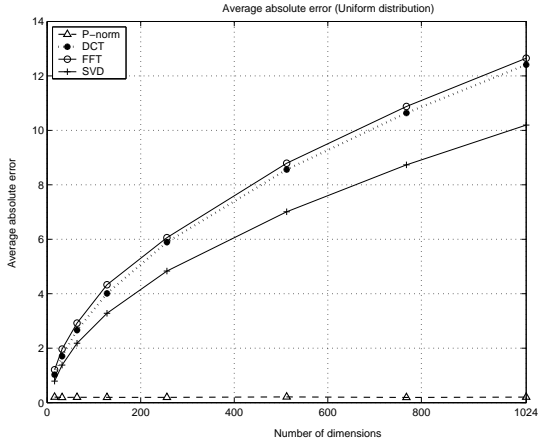
Figure 5: The comparison of number of false hits obtained by symmetric p -norms approximation and proposed technique using real data. Number of false hits are shown for varying k -nearest neighbors.

these values as the summary. Specifically, we used 2 partitions and used $m = 2$ in the p -norms approximation (summary size for each vector is 4) in our method. Consequently, for SVD and DCT 4 coefficients, for DFT 5 coefficients are stored in the summary. Finally, the performance of these summary techniques are evaluated by computing the pairwise distances using the lower dimensional representatives and comparing the average absolute errors defined by

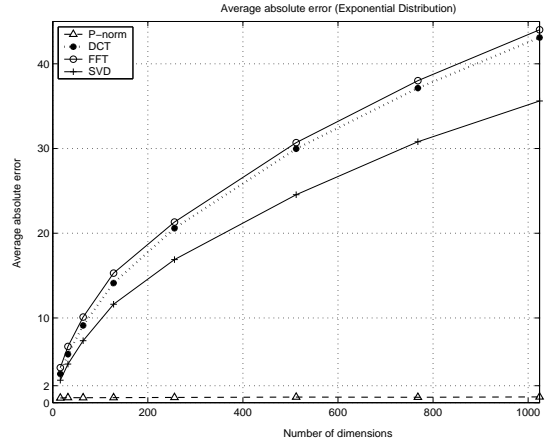
$$\frac{\sum_{\forall \text{pairs}(x,y)} \|d(x,y) - \hat{d}(x,y)\|}{\text{Number of pairs}}$$

where $d(x,y)$ is the actual distance and $\hat{d}(x,y)$ is the distance found using the summary. This comparison is repeated for dimensions ranging from 16 to 1024 and the change of average absolute errors with dimension is shown in Figure 6. For The results clearly show that approximations based on p -norms performs considerably better than other methods. Experiments repeated for various distributions and different sizes of summaries yielded similar results.

Query performance In this part, we perform experiments to evaluate the performance of the combined shape and magnitude approximations for $k - nn$ queries using real data sets. The first data set is stock market data containing 360 days price movements of 2000 companies. In this experiment, the actual pairwise distances between each pair in the database is computed. Then, the k nearest neighbors of each vector is found and stored. After that, we reduced the dimension of the data using p -norms plus shape approximation, KLT, DCT and FFT as in the previous case. For p norms, we used $m = 2$ and partitioned the data set into $k = 4$ subspaces. For the DFT, we retained 9 coefficients and for other methods we used 8 coefficients. Then, the pairwise distances are computed using the reduced dimensional vectors and the k nearest neighbors are found for each case. Finally, we took the average of the number of false hits over the data set. We repeated the experiments for $k = 20$ to $k = 200$. The results obtained using $m = 2$ and 4 partitions is given in Figure 7(a)-(c) 7(b) for $k = 80$, $k = 100$

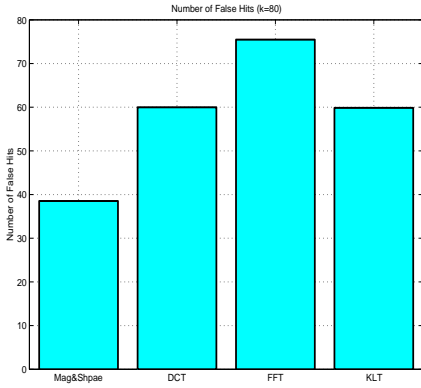


(a) Uniform distribution on the interval (0,1).

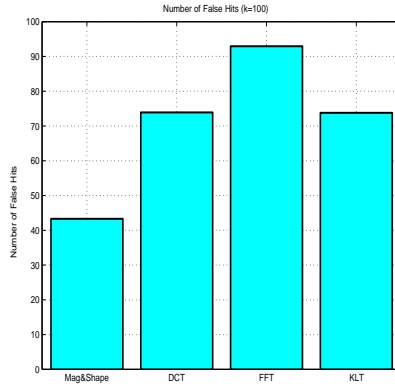


(b) Exponential distribution with parameter 1.

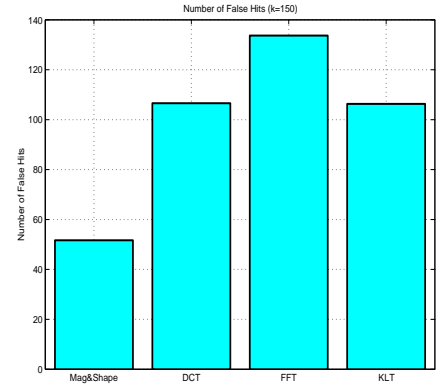
Figure 6: Comparison of average absolute errors obtained using each technique. For p -norms 2 partitions are used with $m = 2$, for other techniques, 5 coefficients are used.



(a) Number of false hits obtained for 80 nearest neighbors.



(b) Number of false hits obtained for 100 nearest neighbors.



(c) Number of false hits obtained for 150 nearest neighbors.

Figure 7: Comparison of the performances of the techniques for $k - nn$ query.

and $k = 150$, respectively. We observe that the p -norms combined with shape approximation has better performance than KLT, DCT and FFT. This result shows that the approach adapted in Section 3 indeed enhances the quality of the approximation achieved by magnitude approximation. To justify this observation we repeated the experiments with different data sets and obtained similar results as it will be shown in the next experiments.

Comparison with the technique in [14] In the last set of experiments, we further analyze the effect of incorporating shape information both comparing the results with symmetric case and with the approach discussed in [14]. In [14], an n -dimensional vector x is decomposed into two pairs as $(s(x), \sigma(x))$, as mentioned in Section 3.1. $s(x)$ is the sorted version of x into weakly increasing coordinates, thus, it results in the same magnitude approximation as x . On the other hand, $\sigma(x)$ is the permutation of indices, hence it maintains the information about the shape of the vector. Given two vectors x and y , the author defines a measure

of distance between the permutations $\sigma(x)$ and $\sigma(y)$, in addition to the magnitude approximation based on $s(x)$ and $s(y)$. Then, the Euclidian distance is approximated as convex combination of these two measures

$$D(x, y) \approx \lambda s(x, y) + (1 - \lambda)\pi(x, y)$$

where $s(x, y)$ is the magnitude approximation, $\pi(x, y)$ is the shape approximation and λ is the bias factor. The experiments are performed using the same real data sets. This time, the query is selected from a similar data set and k nearest neighbors are found using original vectors and reduced dimensions. In addition to proposed method and symmetric p -norms approximations, we also implemented the algorithm described in [14] to find the shape approximation using the permutations $\sigma(x)$ and $\sigma(y)$. After finding magnitude and shape approximations separately, we varied the bias factor λ from 0 to 1 and computed the Euclidian distance approximation corresponding for each λ . The variation of number of false hits for different λ values are shown in Figure 8.

Metric	p-norms	Shape Approximation (from [14])	Proposed Method
Number of false hits	60	97	40

Table 1: Comparison of the dimensionality reduction methods with respect to number of false hits using real data.

This plot shows that, the best results are achieved, when the bias is towards the magnitude approximation. Adding the shape approximation degrades the quality of the Euclidian distance approximation, although it provides a means to incorporate the similarity in the shape of the vectors with adjustable bias towards magnitude or shape. We summarize the results of this experiment in Table 1. We give the number of false hits obtained using the our method, magnitude approximation ($\lambda = 1$) and shape approximation ($\lambda = 0$) for 100 nearest neighbor search. It is clear that, the performance of the shape approximation alone worse than the magnitude approximation. Combining those two does not increase the performance for k -nn queries. The combined magnitude and shape approximation proposed in this paper, on the other hand, enhances the performance of symmetric magnitude approximation. The experiments are also performed for synthetic and other real data sets, and similar results are found.

5. CONCLUSION

In this paper, we presented dimensionality reduction techniques for efficient and accurate approximation of Euclidian distance between high dimensional vectors. We first presented a method based on the approximation of the standard inner-product. The inner product is approximated as a certain function of the p -NORMS (Equation 1) of the vectors. Assuming that the components of the vectors in the data set are identically distributed, we find optimal universal constants b_1, b_2, \dots, b_m so that the approximation

$$\langle x, y \rangle^m \approx b_1 \psi_1(x) \psi_1(y) + \dots + b_m \psi_m(x) \psi_m(y)$$

is the best possible for large n in the least-squares sense. Then, this approximate result is used along with the norms of x and y to approximate the Euclidian distance. Although this approach achieves superior results compared to widely used methods such as KLT and DFT, the symmetry in the p -NORMS is a drawback. Due to this symmetry a vector x and another vector obtained from x by exchanging the position of its entries arbitrarily will result in the same p -NORMS. Hence, the approximation carries information only about the magnitude of the original vector, while the shape information is lost. For this reason, we propose to incorporate shape approximation to the existing magnitude approximation so as to enhance the performance of the method. We achieve this result by partitioning the n dimensional space into orthogonal subspaces and taking the projections of the original vectors on to each of these subspaces. Then, we find a relation between the Euclidian distance of x and y and the Euclidian distance of the projected vectors. That is, our approach is to approximate the distances between the projections using p -NORMS and find the overall approximation using this relation. The experiments performed with real date sets demonstrates the effectiveness of our technique.

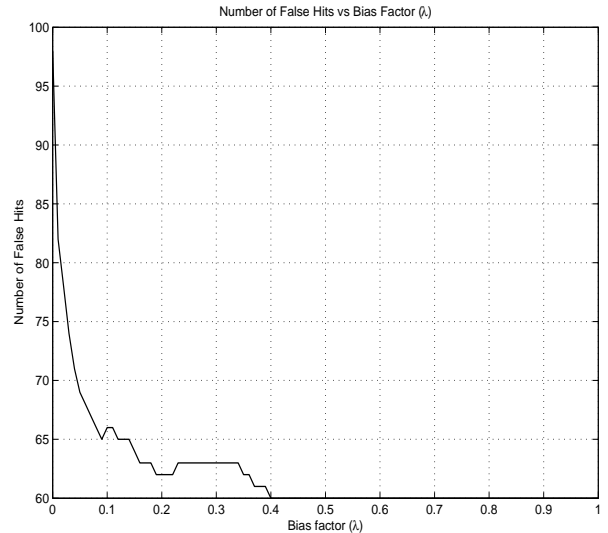


Figure 8: Number of false hits obtained for convex combination of magnitude and shape approximations for varying bias factor λ

6. REFERENCES

- [1] A. Acharya, M. Uysal, and J. Saltz. Active disks: Programming model, algorithms and evaluation. In *ASPLOS-VIII*, pages 81–91, September 1998.
- [2] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *4th Int. Conference on Foundations of Data Organization and Algorithms*, pages 69–84, 1993.
- [3] A. Baruffolo. R-trees for astronomical data indexing. *ASP Conf. Ser., Astronomical Data Analysis Software and Systems VIII*, 172:375, 1999.
- [4] S. Berchtold, C. Bohm, D. Keim, and H. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In *Proc. ACM Symp. on Principles of Database Systems*, pages 78–86, Tuscon, Arizona, June 1997.
- [5] S. Berchtold, D. Keim, and H. Kriegel. The X-tree: An index structure for high-dimensional data. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 28–39, Bombay, India, 1996.
- [6] S. Berchtold and H.-P. Kriegel. S3: Similarity search in CAD database systems. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 564–567, Tuscon, Arizona, 1997.
- [7] P. Bernstein, M. Brodie, S. Ceri, D. DeWitt, M. Franklin, H. Garcia-Molina, J. Gray, J. Held, J. Hellerstein, H. Jagadish, M. Lesk, D. Maier, J. Naughton, H. Pirahesh, M. Stonebraker, and J. Ullman. The Asilomar report on database research. *ACM Sigmod Record*, 27(4), December 1998.
- [8] D. A. Keim C. Bohm, S. Berchtold. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33:322–373, 2001.
- [9] K. R. Castleman. *Digital Image Processing*. Prentice-Hall, Inc., 1996.

- [10] K. Chakrabarti and S. Mehrotra. The hybrid tree: An index structure for high dimensional feature spaces. In *Proc. Int. Conf. Data Engineering*, pages 440–447, Sydney, Australia, 1999.
- [11] S. Deerwester, S.T. Dumais, G.W.Furnas, T.K. Launder, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
- [12] D.Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proc. of the 17th ACM-SIGIR Conference*, pages 282–291, 1994.
- [13] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23:229–236, 1991.
- [14] O. Egecioglu. Parametric approximation algorithms for high-dimensional euclidean similarity. In *Proc. of the 5-th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001)*, pages 79–90, September 2001.
- [15] O. Egecioglu and H. Ferhatosmanoglu. Dimensionality reduction and similarity distance computation by inner product approximations. In *Proceedings of the 9th ACM Int. Conf. on Information and Knowledge Management*, pages 219–226, McLean, Virginia, November 2000.
- [16] Ö. Egecioglu. How to approximate the inner-product:fast dynamic algorithms for euclidean similarity. Technical Report TRCS98-37, Comp. Sci. Dept., UC, Santa Barbara, December 1998.
- [17] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
- [18] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 419–429, Minneapolis, May 1994.
- [19] H. Ferhatosmanoglu, D. Agrawal, and A. El Abbadi. Optimal partitioning for efficient I/O in spatial databases. In *Proc. of the European Conference on Parallel Computing (Euro-Par)*, Manchester, UK, August 2001.
- [20] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi. Approximate nearest neighbor searching in multimedia databases. In *Proc of 17th IEEE Int. Conf. on Data Engineering (ICDE)*, pages 503–511, Heidelberg, Germany, April 2001.
- [21] V. Gaede and O. Gunther. Multidimensional access methods. *ACM Computing Surveys*, 30:170–231, 1998.
- [22] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 518–529, Edinburgh, Scotland, UK, September 1999.
- [23] Informix. <http://www.ibm.com/software/data/informix/blades/spatial/rtree.html>, 2002.
- [24] N. S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Inc., 1984.
- [25] T. Kailath. *Modern Signal Processing*. Springer Verlag, 1985.
- [26] K. V. R. Kanth, D. Agrawal, and A. Singh. Dimensionality reduction for similarity searching in dynamic databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 166–176, Seattle, Washington, June 1998.
- [27] K. V. R. Kanth, S. Ravada, and D. Abugov. Quadtree and r-tree indexes in oracle spatial: A comparison using gis data. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Madison, Wisconsin, June 2002.
- [28] H. Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Science Fenn*, 1947.
- [29] N. Katayama and S. Satoh. The sr-tree: an index structure for high-dimensional nearest neighbor queries. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 369–380, Tucson, Arizona, May 1997.
- [30] E. J. Keogh, K. Chakrabarti, S. Mehrotra, and M. J. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 151–162, Santa Barbara, CA, 2001. ACM.
- [31] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast nearest neighbor search in medical image databases. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 215–226, Mumbai, India, 1996.
- [32] K. Lin, H. V. Jagadish, and C. Faloutsos. The TV-tree: An index structure for high-dimensional data. *VLDB Journal*, 3:517–542, 1995.
- [33] M. Loeve. *Fonctions aleatoires de seconde ordre. Processus Stochastiques et Mouvement Brownien*, 1948.
- [34] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *Proc. of the SPIE Conf. 1908 on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, February 1993.
- [35] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 1989.
- [36] H. Samet. *The Design and Analysis of Spatial Structures*. Addison Wesley Publishing Company, Inc., Massachusetts, 1989.
- [37] T. Seidl and Kriegel H.-P. Efficient user-adaptable similarity search in large multimedia databases. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 506–515, Athens, Greece, 1997.
- [38] T. Seidl and H.P. Kriegel. Optimal multi-step k-nearest neighbor search. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Chicago, Illinois, U.S.A., June 1998. ACM.
- [39] V.S. Subrahmanian. *Principles of Multimedia Database Systems*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.
- [40] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, N. Koudas, and H. D. Schwetman. Non-linear dimensionality reduction techniques for classification and visualization. In *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.
- [41] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the Int. Conf. on Very Large Data Bases*, pages 194–205, New York City, New York, August 1998.
- [42] A. J. Wicenc and M. Albrecht. Methods for structuring and searching very large catalogs. *ASP Conf. Ser., Astronomical Data Analysis Software and Systems VII*, 145:512, 1998.
- [43] D. Wu, D. Agrawal, A. El Abbadi, and T. R. Smith. Efficient retrieval for browsing large image databases. In *Proc. Conf. on Information and Knowledge Management*, pages 11–18, November 1996.