On Fast Non-Metric Similarity Search by Metric Access Methods

Tomáš Skopal

Charles University in Prague, FMP, Department of Software Engineering,
Malostranské nám. 25, 118 00 Prague 1, Czech Republic
tomas@skopal.net

Abstract. The retrieval of objects from a multimedia database employs a measure which defines a similarity score for every pair of objects. The measure should *effectively* follow the nature of similarity, hence, it should not be limited by the triangular inequality, regarded as a restriction in similarity modeling. On the other hand, the retrieval should be as *efficient* (or fast) as possible. The measure is thus often restricted to a *metric*, because then the search can be handled by *metric access methods* (MAMs). In this paper we propose a general method of non-metric search by MAMs. We show the triangular inequality can be enforced for any *semimetric* (reflexive, non-negative and symmetric measure), resulting in a metric that preserves the original similarity orderings (retrieval effectiveness). We propose the *TriGen* algorithm for turning any black-box semimetric into (approximated) metric, just by use of distance distribution in a fraction of the database. The algorithm finds such a metric for which the retrieval efficiency is maximized, considering any MAM.

1 Introduction

In multimedia databases the semantics of data objects is defined loosely, while for querying such objects we usually need a similarity measure standing for a judging mechanism of how much are two objects similar. We can observe two particular research directions in the area of content-based multimedia retrieval, however, both are essential. The first one follows the subject of retrieval *effectiveness*, where the goal is to achieve query results complying with the user's expectations (measured by the *precision* and *recall* scores). As the effectiveness is obviously dependent on the semantics of similarity measure, we require the possibilities of similarity measuring as rich as possible, thus, the measure should not be limited by properties regarded as restrictive for similarity modeling.

Following the second direction, the retrieval should be as *efficient* (or fast) as possible, because the number of objects in a database can be large and the similarity scores are often expensive to compute. Therefore, the similarity measure is often restricted by metric properties, so that retrieval can be realized by metric access methods. Here we have reached the point. The "effectiveness researchers" claim the metric properties, especially the triangular inequality, are too restrictive. However, the "efficiency researchers" reply the triangular inequality is the most powerful tool to keep the search in a database efficient.

In this paper we show the triangular inequality is not restrictive for similarity search, since every semimetric can be modified into a suitable metric and used for the search instead. Such a metric can be constructed even automatically, just with a partial information about distance distribution in the database.

1.1 Preliminaries

Let a multimedia object $\mathcal{O}$ be modeled by a *model object* $O \in \mathbb{U}$, where $\mathbb{U}$ is a model universe. A multimedia database is then represented by a dataset $\mathbb{S} \subset \mathbb{U}$.

Definition 1 (similarity & dissimilarity measure)
Let $s : \mathbb{U} \times \mathbb{U} \mapsto \mathbb{R}$ be a *similarity measure*, where $s(O_i, O_j)$ is considered as a similarity score of objects $\mathcal{O}_i$ and $\mathcal{O}_j$. In many cases it is more suitable to use a *dissimilarity measure* $d : \mathbb{U} \times \mathbb{U} \mapsto \mathbb{R}$ equivalent to a similarity measure s as $s(Q, O_i) > s(Q, O_j) \Leftrightarrow d(Q, O_i) < d(Q, O_j)$. A dissimilarity measure assigns a higher score (or *distance*) to less similar objects, and vice versa.

The measures often satisfy some of the metric properties. The *reflexivity* ($d(O_i, O_j) = 0 \Leftrightarrow O_i = O_j$) permits the zero distance just for identical objects. Both reflexivity and *non-negativity* ($d(O_i, O_j) \geq 0$) guarantee every two distinct objects are somehow positively dissimilar. If d satisfies reflexivity, non-negativity and *symmetry* ($d(O_i, O_j) = d(O_j, O_i)$), we call d a *semimetric*. Finally, if a semimetric d satisfies also the *triangular inequality* ($d(O_i, O_j) + d(O_j, O_k) \geq d(O_i, O_k)$), we call d a *metric* (or metric distance). This inequality is a kind of transitivity property; it says if O_i, O_j and O_j, O_k are similar, then also O_i, O_k are similar. If there is an upper bound d^+ such that $d : \mathbb{U} \times \mathbb{U} \mapsto \langle 0, d^+ \rangle$, we call d a *bounded metric*. The pair $\mathcal{M} = (\mathbb{U}, d)$ is called a (bounded) *metric space*. □

Definition 2 (triangular triplet)
A triplet (a, b, c), $a, b, c \geq 0$, $a + b \geq c$, $b + c \geq a$, $a + c \geq b$, is called a *triangular triplet*. Let (a, b, c) be ordered as $a \leq b \leq c$, then (a, b, c) is an *ordered triplet*. If $a \leq b \leq c$ and $a + b \geq c$, then (a, b, c) is called an *ordered triangular triplet*. □

A metric d generates just the (ordered) triangular triplets, i.e. $\forall O_i, O_j, O_k \in \mathbb{U}$, ($d(O_i, O_j)$, $d(O_j, O_k)$, $d(O_i, O_k)$) is triangular triplet. Conversely, if a measure generates just the triangular triplets, then it satisfies the triangular inequality.

1.2 Similarity Queries

In the following we consider the *query-by-example* concept; we look for objects similar to a query object $Q \in \mathbb{U}$ (Q is derived from an example object). Necessary to the query-by-example retrieval is a notion of *similarity ordering*, where the objects $O_i \in \mathbb{S}$ are ordered according to the distances to Q. For a particular query there is specified a portion of the ordering returned as the query result. The *range query* and the *k nearest neighbors (k-NN) query* are the most popular ones. A range query (Q, r_Q) selects objects from the similarity ordering for which $d(Q, O_i) \leq r_Q$, where $r_Q \geq 0$ is a distance threshold (or query radius). A k-NN query (Q, k) selects the k most similar objects (first k objects in the ordering).

1.3 Metric Access Methods

Once we have to search according to a metric d, we can use the *metric access methods* (MAMs) [5], which organize (or index) a given dataset $\mathbb{S}$ in a way that similarity queries can be processed efficiently by use of a *metric index*, hence, without the need of searching the entire dataset $\mathbb{S}$. The main principle behind all MAMs is a utilization of the triangular inequality (satisfied by any metric), due to which MAMs can organize the objects of $\mathbb{S}$ in distinct classes. When a query is processed, only the candidate classes are searched (such classes which overlap the query), so the searching becomes more efficient (see Figure 1a).

In addition to the number of distance computations $d(\cdot, \cdot)$ needed (the *computation costs*), the retrieval efficiency is affected also by the *I/O costs*. To minimize the search costs, i.e. to increase the retrieval efficiency, there were developed many MAMs for different scenarios (e.g. designed to secondary storage or main memory management). Besides others we name *M-tree*, *vp-tree*, *LAESA* (we refer to a survey [5]), or more recent ones, *D-index* [9] and *PM-tree* [27].

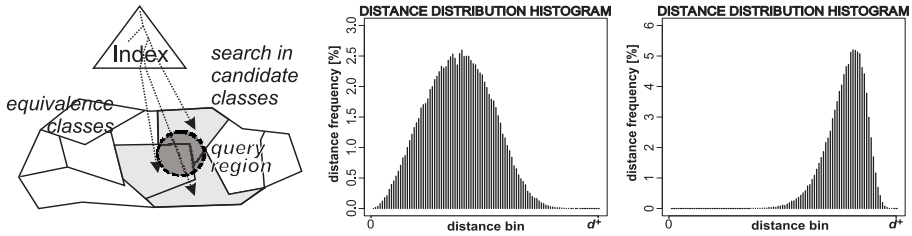

Fig. 1. Search by MAMs (a), DDHs indicating low (b) and high (c) intrinsic dim.

1.4 Intrinsic Dimensionality

The metric access methods are not successful for all datasets and all metrics; the retrieval efficiency is heavily affected by *distance distribution* in the dataset. Given a dataset $\mathbb{S}$ and a metric d, the efficiency limits of any MAM are indicated by the *intrinsic dimensionality*, defined as $\rho(\mathbb{S}, d) = \frac{\mu^2}{2\sigma^2}$, where μ and σ^2 are the mean and the variance of the distance distribution in $\mathbb{S}$ (proposed in [4]). In Figures 1b,c see an example of distance distribution histograms (DDHs) indicating low ($\rho = 3.61$) and high ($\rho = 42.35$) intrinsic dimensionalities.

The intrinsic dimensionality is low if there exist tight clusters of objects. Conversely, if all the indexed objects are almost equally distant, then intrinsic dimensionality is high, which means the dataset is poorly intrinsically structured. A high ρ value says that many (even all) of MAM's classes created on $\mathbb{S}$ are overlapped by every possible query, so that processing deteriorates to sequential search in all the classes. The problem of high intrinsic dimensionality is, in fact, a generalization of the *curse of dimensionality* [31, 4] into metric spaces.

1.5 Theories of Similarity Modeling

The metric properties have been argued against as restrictive in similarity modeling [25, 28]. In particular, the reflexivity and non-negativity have been refuted

[21, 28] by claiming that different objects could be differently self-similar. Nevertheless, these are the less problematic properties. The symmetry was questioned by showing that a prototypical object can be less similar to an indistinct one than vice versa [23, 24]. The triangular inequality is the most attacked property [2, 29]. Some theories point out the similarity has not to be transitive. Demonstrated by the well-known example, a man is similar to a centaur, the centaur is similar to a horse, but the man is completely dissimilar to the horse.

1.6 Examples of Non-Metric Measures

In the following we name several dissimilarity measures of two kinds, proved to be effective in similarity search, but which violate the triangular inequality.

Robust Measures. A *robust* measure is resistant to outliers – anomalous or "noisy" objects. For example, various *k-median distances* measure the kth most similar portion of the compared objects. Generally, a k-median distance d is of form $d(O_1, O_2) = k\text{–}med(\delta_1(O_1, O_2), \delta_2(O_1, O_2), \ldots, \delta_n(O_1, O_2))$, where $\delta_i(O_1, O_2)$ is a distance between O_1 and O_2, considering the ith portion of the objects. Among the partial distances δ_i the $k\text{–}med$ operator returns the kth smallest value. As a special k-median distance derived from the Hausdorff metric, the *partial Hausdorff distance* (pHD) has been proposed for shape-based image retrieval [17]. Given two sets $\mathcal{S}_1, \mathcal{S}_2$ of points (e.g. two polygons), the partial Hausdorff distance uses $\delta_i(\mathcal{S}_1, \mathcal{S}_2) = dNP(\mathcal{S}_1^i, \mathcal{S}_2)$, where dNP is the Euclidean (L_2) distance of the ith point in $\mathcal{S}_1$ to the nearest point in $\mathcal{S}_2$. To keep the distance symmetric, pHD is the maximum, i.e. $pHD(\mathcal{S}_1, \mathcal{S}_2) = max(d(\mathcal{S}_1, \mathcal{S}_2), d(\mathcal{S}_2, \mathcal{S}_1))$. Similar to pHD is another modification of Hausdorff metric, used for face detection [20], where the average of dNP distances is considered, instead of k-median.

The *time warping distance* for sequence aligning has been used in time series retrieval [33], and even in shape retrieval [3]. The *fractional L_p distances* [1] have been suggested for robust image matching [10] and retrieval [16]. Unlike classic L_p metrics ($\mathrm{L}_p(u, v) = (\sum_{i=1}^{n} |u_i - v_i|^p)^{\frac{1}{p}}$, $p \geq 1$), the fractional L_p distances use $0 < p < 1$, which allows us to inhibit extreme differences in coordinate values.

Complex Measures. In the real world, the algorithms for similarity measuring are often complex, even adaptive or learning. Moreover, they are often implemented by heuristic algorithms which combine several measuring strategies. Obviously, an analytic enforcement of triangular inequality for such measures can be simply too difficult. The COSIMIR method [22] uses a back-propagation neural network for supervised similarity modeling and retrieval. Given two vectors $u, v \in \mathbb{S}$, the distance between u and v is computed by activation of three-layer network. This approach allows to train the similarity measure by means of user-assessed pairs of objects. Another example of complex measure can be the *matching by deformable templates* [19], utilized in handwritten digits recognition. Two digits are compared by deforming the contour of one to fit the edges of the other. The distance is derived from the amount of deformation needed, the goodness of edges fit, and the interior overlap between the deformed shapes.

1.7 Paper Contributions

In this paper we present a general approach to efficient and effective non-metric search by metric access methods. First, we show that every semimetric can be non-trivially turned into metric and used for similarity search by MAMs. To achieve this goal, we modify the semimetric by a suitable *triangle-generating modifier*. In consequence, we also claim the triangular inequality is completely unrestrictive with respect to the effectiveness of similarity search. Second, we propose the *TriGen* algorithm for automatic conversion of any "black-box" semimetric (i.e. semimetric given in a non-analytic form) into (approximated) metric, such that intrinsic dimensionality of the indexed dataset is kept as low as possible. The optimal triangle-generating modifier is found by use of predefined base modifiers and by use of distance distribution in a (small) portion of the dataset.

2 Related Work

The simplest approach to non-metric similarity search is the *sequential search* of the entire dataset. The query object is compared against every object in the dataset, resulting in a similarity ordering which is used for the query evaluation. The sequential search often provides a baseline for other retrieval methods.

2.1 Mapping Methods

The non-metric search can be indirectly carried out by various *mapping methods* [11, 15] (e.g. MDS, FastMap, MetricMap, SparseMap). The dataset $\mathbb{S}$ is embedded into a vector space $(\mathbb{R}^k, \delta)$ by a mapping $F : \mathbb{S} \mapsto \mathbb{R}^k$, where the distances $d(\cdot, \cdot)$ are (approximately) preserved by a cheap vector metric δ (often the L_2 distance). Sometimes the mapping F is required to be *contractive*, i.e. $\delta(F(O_i), F(O_j)) \leq d(O_i, O_j)$, which allows to filter out some irrelevant objects using δ, but some other irrelevant objects, called *false hits*, must be re-filtered by d (see e.g. [12]). The mapped vectors can be indexed/retrieved by any MAM.

To say the drawbacks, the mapping methods are expensive, while the distances are preserved only approximately, which leads to *false dismissals* (i.e. to relevant objects being not retrieved). The contractive methods eliminate the false dismissals but suffer from a great number of false hits (especially when k is low), which leads to lower retrieval efficiency. In most cases the methods need to process the dataset in a batch, so they are suitable for static MAMs only.

2.2 Lower-Bounding Metrics

To support similarity search by a non-metric distance d_Q, the QIC-M-tree [6] has been proposed as an extension of the M-tree (the key idea is applicable also to other MAMs). The M-tree index is built by use of an index distance d_I, which is a metric *lower-bounding* the query distance d_Q (up to a scaling constant $S_{I \to Q}$), i.e. $d_I(O_i, O_j) \leq S_{I \to Q}\ d_Q(O_i, O_j), \forall O_i, O_j \in \mathbb{U}$. As d_I lower-bounds d_Q, a query

can be partially processed by d_I (which, moreover, could be much cheaper than d_Q), such that many irrelevant classes of objects (subtrees in M-tree) are filtered out. All objects in the non-filtered classes are compared against Q using d_Q. Actually, this approach is similar to the usage of contractive mapping methods (d_I is an analogy to δ), but here the objects generally need not to be mapped into a vector space. However, this approach has two major limitations. First, for a given non-metric distance d_Q there was not proposed a general way how to find the metric d_I. Although d_I could be found "manually" for a particular d_Q (as in [3]), this is not easy for d_Q given as a black box (an algorithmically described one). Second, the lower-bounding metric should be as tight approximation of d_Q as possible, because this "tightness" heavily affects the intrinsic dimensionality, the number of MAMs' filtered classes, and so the retrieval efficiency.

2.3 Classification

Quite many attempts to non-metric nearest neighbor (NN) search have been tried out in the classification area. Let us recall the basic three steps of classification. First, the dataset is organized in classes of similar objects (by user annotation or clustering). Then, for each class a description consisting of the most representative object(s) is created; this is achieved by *condensing* [14] or *editing* [32] algorithms. Third, the NN search is accomplished as a classification of the query object. Such a class is searched, to which the query object is "nearest", since there is an assumption the nearest neighbor is located in the "nearest class". For non-metric classification there have been proposed methods enhancing the description of classes (step 2). In particular, condensing algorithms producing *atypical points* [13] or *correlated points* [18] have been successfully applied.

The drawbacks of classification-based methods reside in static indexing and limited scalability, while the querying is restricted just to approximate (k-)NN.

3 Turning Semimetric into Metric

In our approach, a given dissimilarity measure is turned into a metric, so that MAMs can be directly used for the search. This idea could seem to disclaim the results of similarity theories (mentioned in Section 1.5), however, we must realize the task of **similarity search** employs only a limited modality of **similarity modeling**. In fact, in similarity search we just need to order the dataset objects according to a single query object and pick the most similar ones. Clearly, if we find a metric for which such similarity orderings are the same as for the original dissimilarity measure, we can safely use the metric instead of the measure.

3.1 Assumptions

We assume d satisfies reflexivity and non-negativity but, as we have mentioned in Section 1.5, these are the less restrictive properties and can be handled easily; e.g. the *non-negativity* is satisfied by a shift of the distances, while for the *reflexivity*

property we require every two non-identical objects are at least d^--distant (d^- is some positive distance lower bound). Furthermore, searching by an *asymmetric measure* δ could be partially provided by a symmetric measure d, e.g. $d(O_i, O_j) = min\{\delta(O_i, O_j), \delta(O_j, O_i)\}$. Using the symmetric measure some irrelevant objects can be filtered out, while the original asymmetric measure δ is then used to rank the remaining non-filtered objects. In the following we assume the measure d is a bounded semimetric, nevertheless, this assumption is introduced just for clarity of the following presentation. Finally, as d is bounded by d^+, we can further simplify the semimetric such that it assigns distances from $\langle 0, 1\rangle$. This can be achieved simply by scaling the original value $d(O_i, O_j)$ to $d(O_i, O_j)/d^+$. The same way a range query radius r_Q must be scaled to r_Q/d^+, when searching.

3.2 Similarity-Preserving Modifications

Based on the assumptions, the only property we have to solve is the triangular inequality. To do so, we apply some special modifying function on the semimetric, such that the original similarity orderings are preserved.

Definition 3 (similarity-preserving modification)
Given a measure d, we call $d^f(O_i, O_j) = f(d(O_i, O_j))$ a *similarity-preserving modification of* d (or *SP-modification*), where f, called the *similarity-preserving modifier* (or *SP-modifier*), is a strictly increasing function for which $f(0) = 0$. Again, for clarity reasons we assume f is bounded, i.e. $f : \langle 0, 1\rangle \mapsto \langle 0, 1\rangle$. □

Definition 4 (similarity ordering)
We define $SimOrder_d : \mathbb{U} \mapsto 2^{\mathbb{U}\times\mathbb{U}}, \forall O_i, O_j, Q \in \mathbb{U}$ as $\langle O_i, O_j\rangle \in SimOrder_d(Q) \Leftrightarrow d(Q, O_i) < d(Q, O_j)$, i.e. $SimOrder_d$ orders objects by their distances to Q. □

Lemma 1
Given a metric d and any d^f, then $SimOrder_d(Q) = SimOrder_{d^f}(Q), \forall Q \in \mathbb{U}$.
Proof: As f is increasing, then $\forall Q, O_i, O_j \in \mathbb{U}$ it follows that $d(Q, O_i) > d(Q, O_j) \Leftrightarrow f(d(Q, O_i)) > f(d(Q, O_j))$. ■

In other words, every SP-modification d^f preserves the similarity orderings generated by d. Consequently, if a query is processed sequentially (by comparing all objects in $\mathbb{S}$ to the query object Q), then it does not matter if we use either d or any d^f, because both ways induce the same similarity orderings. Naturally, the radius r_Q of a range query must be modified to $f(r_Q)$, when searching by d^f.

3.3 Triangle-Generating Modifiers

To obtain a modification forcing a semimetric to satisfy the triangular inequality, we have to use some special SP-modifiers based on metric-preserving functions.

Definition 5 (metric-preserving SP-modifier)

A SP-modifier f is *metric-preserving* if for every metric d the SP-modification d^f preserves the triangular inequality, i.e. d^f is also metric. Such a SP-modifier must be additionally *subadditive* $(f(x) + f(y) \geq f(x + y), \forall x, y)$. □

Lemma 2

(a) Every concave SP-modifier f is metric-preserving.
(b) Let (a, b, c) be a triangular triplet and f be metric-preserving, then $(f(a), f(b), f(c))$ is a triangular triplet as well.

Proof: For the proof and for more about metric-preserving functions see [8]. ■

To modify a semimetric into metric, we have utilized a class of metric-preserving SP-modifiers, denoted as the triangle-generating modifiers.

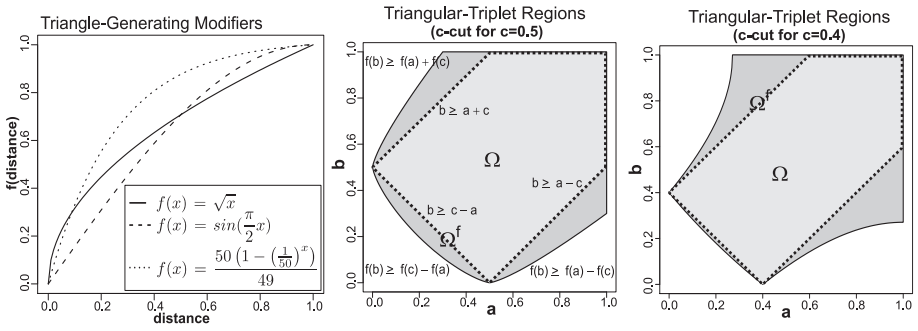

Fig. 2. (a) Several TG-modifiers. Regions Ω, Ω^f; (b) $f(x) = x^{\frac{3}{4}}$ (c) $f(x) = sin(\frac{\pi}{2}x)$

Definition 6 (triangle-generating modifier)

Let a strictly concave SP-modifier f be called a *triangle-generating modifier* (or *TG-modifier*). Having a TG-modifier f, let a d^f be called a *TG-modification*. □

The TG-modifiers (see examples in Figure 2a) not only preserve the triangular inequality, they can even enforce it, as follows.

Theorem 1

Given a semimetric d, then there always exists a TG-modifier f, such that the SP-modification d^f is a metric.

Proof: We show that every ordered triplet (a, b, c) generated by d can be turned by a single TG-modifier f into an ordered triangular triplet.

1. As every semimetric is reflexive and non-negative, it generates ordered triplets just of forms $(0, 0, 0)$, $(0, c, c)$, and (a, b, c), where $a, b, c > 0$. Among these, just the triplets (a, b, c), $0 < a \leq b < c$, can be non-triangular. Hence, it is sufficient to show how to turn such triplets by a TG-modifier into triangular ones.

2. Suppose an arbitrary TG-modifier f_1. From TG-modifiers' properties it follows that $\frac{f_1(a)}{f_1(c)} > \frac{a}{c}$, $\frac{f_1(b)}{f_1(c)} > \frac{b}{c}$, hence $\frac{f_1(a)+f_1(b)}{f_1(c)} > \frac{a+b}{c}$ (theory of concave functions). If $(f_1(a) + f_1(b))/f_1(c) \geq 1$, the triplet $(f_1(a), f_1(b), f_1(c))$ becomes triangular

(i.e. $f_1(a) + f_1(b) \geq f_1(c)$ is true). In case there still exist triplets which have not become triangular after application of f_1, we take another TG-modifier f_2 and compose f_1 and f_2 into $f^*(x) = f_2(f_1(x))$. The compositions (or nestings) $f^*(x) = f_i(\dots f_2(f_1(x))\dots)$ are repeated until f^* turns all triplets generated by d into triangular ones – then f^* is the single TG-modifier f we are looking for. ■

The proof shows the more concave TG-modifier we apply, the more triplets become triangular. This effect can be visualized by 3D regions in the space $\langle 0,1\rangle^3$ of all possible distance triplets, where the three dimensions represent the distance values a,b,c, respectively. In Figures 2b,c see examples of region[1] Ω of all triangular triplets as the dotted-line area. The super-region Ω^f (the solid-line area) represents all the triplets which become (or remain) triangular after the application of TG-modifier $f(x) = x^{\frac{3}{4}}$ and $f(x) = sin(\frac{\pi}{2}x)$, respectively.

3.4 TG-Modifiers Suitable for Metric Search

Although there exist infinitely many TG-modifiers which turn a semimetric d into a metric d^f, their properties can be quite different with respect to the efficiency of search by MAMs. For example, $f(x) = \begin{cases} 0 & (\text{for } x = 0) \\ \frac{x+d^+}{2} & (\text{otherwise}) \end{cases}$ turns every d^+-bounded semimetric d into a metric d^f. However, such a metric is useless for searching, since all classes of objects maintained by a MAM are overlapped by every query, so the retrieval deteriorates to sequential search. This behavior is also reflected in high intrinsic dimensionality of $\mathbb{S}$ with respect to d^f.

In fact, we look for an *optimal* TG-modifier, i.e. a TG-modifier which turns only such non-triangular triplets into triangular ones, which are generated by d. The non-triangular triplets which are not generated by d should remain non-triangular (the white areas in Figures 2b,c), since such triplets represent the "decisions" used by MAMs for filtering of irrelevant objects or classes. The more often such decisions occur, the more efficient the search is (and the lower the intrinsic dimensionality of $\mathbb{S}$ is). As an example, given two vectors u, v of dimensionality n, the optimal TG-modifier for semimetric $d(u,v) = \sum_{i=1}^{n} |u_i - v_i|^2$ is $f(x) = \sqrt{x}$, turning d into the Euclidean (L_2) distance.

From another point of view, the concavity of f determines how much the object clusters (MAMs' classes respectively) become indistinct (overlapped by other clusters/classes). This can be observed indirectly in Figure 2a, where the concave modifiers make the small distances greater, while the great distances remain great; i.e. the mean of distances increases, whereas the variance decreases. To illustrate this fact, we can reuse the example back in Figures 1b,c, where the first DDH was sampled for $d_1 = L_2$, while the second one was sampled for a modification $d_2 = L_2^f$, $f(x) = x^{\frac{1}{4}}$.

In summary, given a dataset $\mathbb{S}$, a semimetric d, and a TG-modifier f, the intrinsic dimensionality is always higher for the modification d^f than for d, i.e. $\rho(\mathbb{S}, d^f) > \rho(\mathbb{S}, d)$. Therefore, an optimal TG-modifier should minimize the increase of intrinsic dimensionality, yet generate the necessary triangular triplets.

[1] The 2D representations of Ω and Ω^f regions are c-cuts of the real 3D regions.

4 The TriGen Algorithm

The question is how to find the optimal TG-modifier f. Had we known an analytical form of d, we could find the TG-modifier "manually". However, if d is implemented by an algorithm, or if the analytical form of d is too complex (e.g. the neural network representation used by COSIMIR), it could be very hard to determine f analytically. Instead, our intention is to find f automatically, regardless of analytical form of d. In other words, we consider a given semimetric d generally as a black box that returns a distance value from a two-object input.

The idea of automatic determination of f makes use of the distance distribution in a sample $\mathbb{S}^*$ of the dataset $\mathbb{S}$. We take m ordered triplets, where each triplet (a, b, c) stores distances between some objects $O_i, O_j, O_k \in \mathbb{S}^* \subseteq \mathbb{S}$, i.e. $(a{=}d(O_i, O_j), b{=}d(O_j, O_k), c{=}d(O_i, O_k))$. Some predefined *base TG-modifiers* f_i (or *TG-bases*) are then applied on the triplets; for each triplet (a, b, c) a modified triplet $(f_i(a), f_i(b), f_i(c))$ is obtained. The *triangle-generating error* ε_Δ (or *TG-error*) is computed as the fraction of triplets remaining non-triangular, $\varepsilon_\Delta = \frac{m_{nt}}{m}$, where m_{nt} is the number of modified triplets remaining non-triangular. Finally, such f_i are selected as *candidates* for the optimal TG-modifier, for which $\varepsilon_\Delta = 0$ or, possibly, $\varepsilon_\Delta \leq \theta$ (where θ is a *TG-error tolerance*). To control the degree of concavity, the TG-bases f_i are parameterizable by a *concavity weight* $w \geq 0$, where $w = 0$ makes every f_i the identity, i.e. $f_i(x, 0) = x$, while with increasing w the concavity of f_i increases as well (a more concave f_i decreases m_{nt}; it turns more triplets into triangular ones). In such a way any TG-base can be forced by an increase of w to minimize the TG-error ε_Δ (possibly to zero).

Among the TG-base candidates the optimal TG-modifier (f_i, w) is chosen such that $\rho(\mathbb{S}^*, d^{f^*(x,w^*)})$ is as low as possible. The TriGen algorithm (see Listing 1) takes advantage of halving the concavity interval $\langle w_{\mathrm{LB}}, w_{\mathrm{UB}} \rangle$ or doubling the upper bound w_{UB}, in order to quickly find the optimal concavity weight w for a TG-base f^*. To keep the computation scalable, the number of iterations (in each iteration w is improved) is limited to e.g. 24 (the iterLimit constant).

Listing 1 (the TriGen algorithm)

Input: semimetric d, set $\mathcal{F}$ of TG-bases, sample $\mathbb{S}^*$, TG-error tolerance θ, iteration limit iterLimit
Output: optimal f, w

```
f = w = null; minIDim = ∞                                                    1
sample m distance triplets into a set T (from S* using d)                    2
for each f* in F                                                             3
    w_LB = 0; w_UB = ∞; w* = 1; w_best = -1; i = 0                           4
    while i < iterLimit                                                      5
        if TGError(f*,w*,T) ≤ θ then w_UB = w_best = w* else w_LB = w*       6
        if w_UB ≠ ∞ then w* = (w_LB + w_UB)/2 else w* = 2 * w*               7
        i = i + 1;                                                           8
    end while                                                                9
    if w_best ≥ 0 then                                                       10
        idim = IDim(f*,w_best,T)                                             11
        if idim < minIDim then f = f*; w = w_best; minIDim = idim            12
    end if                                                                   13
end for                                                                      14
```

In Listing 2 the TGError function is described. The TG-error ε_Δ is computed by taking m distance triplets from the dataset sample $\mathbb{S}^*$ onto which the examined

TG-base f^* together with the current weight w^* is applied. The distance triplets are sampled only once – at the beginning of the TriGen's run – whereas the modified triplets are recomputed for each particular f^*, w^*.

The not-listed function IDim (computing $\rho(\mathbb{S}^*, d^{f^*(x,w^*)})$ makes use of the previously obtained modified triplets as well, however, the values in the triplets are used independently; just for evaluation of the intrinsic dimensionality.

Listing 2 (the TGError function)

Input: TG-base f^*, concavity weight w^*, set $\mathcal{T}$ of m sampled distance triplets
Output: TG-error ε_Δ

```
m_nt = 0                                                                          1
for each ot in T // "ot" stands for "ordered triplet"                             2
    if f*(ot.a, w*) + f*(ot.b, w*) < f*(ot.c, w*) then m_nt = m_nt + 1            3
end for                                                                           4
ε_Δ = m_nt / m                                                                    5
```

4.1 Sampling the Distance Triplets

Initially, we have n objects in the dataset sample $\mathbb{S}^*$. Then we create an $n \times n$ distance matrix for storage of pairwise distances $d_{ij} = d(O_i, O_j)$ between the sampled objects. In such a way we are able to obtain up to $m = \binom{n}{3}$ distance triplets for at most $\frac{n(n-1)}{2}$ distance computations. Thus, to obtain a sufficiently large number of distance triplets, the dataset sample $\mathbb{S}^*$ needs to be quite small. Each of the m distance triplets is sampled by a random choice of three among the n objects, while the respective distances are retrieved from the matrix. Naturally, the values in the matrix could be computed "on-demand", just in the moment a distance retrieval is requested. Since d is symmetric, the sub-diagonal half of the matrix can be used for storage of the modified distances $d^f_{ji} = f^*(d_{ij}, w^*)$, however, these are recomputed for each particular f^*, w^*. As in case of distances, also the modified distances can be computed "on-demand".

4.2 Time Complexity Analysis (simplified)

Let $|\mathbb{S}^*|$ be the number of objects in the sample $\mathbb{S}^*$, m be the number of sampled triplets, and $O(d)$ be the complexity of single distance computation. The complexity of $f(\cdot)$ computation is supposed $O(1)$. The overall complexity of TriGen is then $O(|\mathbb{S}^*|^2 * O(d) + \mathsf{iterLimit} * |\mathcal{F}| * m)$, i.e. the distance matrix computation plus the main algorithm. The number of TG-bases $|\mathcal{F}|$ as well as the number of iterations (variable iterLimit) are assumed as (small) constants, hence we get $O(|\mathbb{S}^*|^2 * O(d) + m)$. The size of $\mathbb{S}^*$ and the number m affect the precision of TGError and IDim values, so we can trade off the TriGen's complexity and the precision by choosing $|\mathbb{S}^*| = O(1), O(|\mathbb{S}|)$ and $m = O(1), O(|\mathbb{S}^*|)$, or e.g. $O(|\mathbb{S}^*|^2)$.

4.3 Default TG-Bases

We propose two general-purpose TG-bases for the TriGen algorithm. The simpler one, the *Fractional-Power TG-base* (or *FP-base*), is defined as $\mathrm{FP}(x, w) = x^{\frac{1}{1+w}}$,

see Figure 3a. The advantage of FP-base is there always exists a concavity weight w for which the modified semimetric becomes metric, i.e. the TriGen will always find a solution (after a number of iterations). Furthermore, when using the FP-base, the semimetric d needs not to be bounded. A particular disadvantage of the FP-base is that its concavity is controlled globally, just by the weight w.

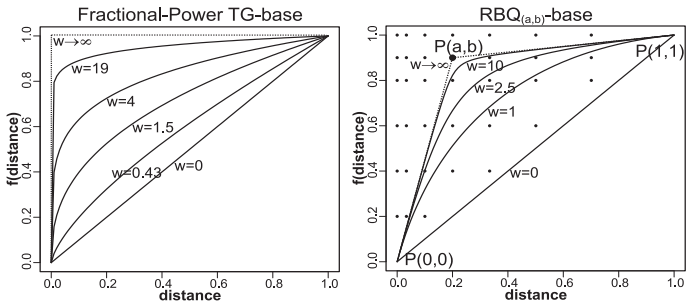

Fig. 3. (a) FP-base (b) $\text{RBQ}_{(a,b)}$-base

As a more flexible TG-base, we have utilized the Rational Bézier Quadratic curve. To derive a proper TG-base from the curve, the three Bézier points are specified as $(0,0),(a,b),(1,1)$, where $0 \leq a < b \leq 1$, see Figure 3b. The *Rational Bézier Quadratic TG-base* (simply RBQ-base) is defined as $\text{RBQ}_{(a,b)}(x,w) = -(\Psi - x + wx - aw) \cdot (-2bwx + 2bw^2x - 2abw^2 + 2bw - x + wx - aw + \Psi(1 - 2bw))/(-1 + 2aw - 4awx - 4a^2w^2 + 2aw^2 + 4aw^2x + 2wx - 2w^2x + 2\Psi(1-w))$, where $\Psi = \sqrt{-x^2 + x^2w^2 - 2aw^2x + a^2w^2 + x}$. The additional RBQ parameters a, b (the second Bézier point) are treated as constants, i.e. for various a, b values (see the dots in Figure 3b) we get multiple RBQ-bases, which are all individually inserted into the set $\mathcal{F}$ of TriGen's input. To keep the RBQ evaluation correct, a possible division by zero or $\Psi^2 < 0$ is prevented by a slight shift of a or w. The advantage of RBQ-bases is the place of maximal concavity can be controlled locally by a choice of (a,b), hence, for a given concavity weight w^* we can achieve lower value of either $\rho(\mathbb{S}^*, d^{f^*(x,w^*)})$ or ε_Δ just by choosing different a, b.

As a particular limitation, for usage of RBQ-bases the semimetric d must be bounded (due to the third Bézier point (1,1)). Furthermore, for an RBQ-base with $(a,b) \neq (0,1)$ the TG-error ε_Δ could be generally greater than the TG-error tolerance θ, even in case $w \to \infty$. Nevertheless, having the FP-base or the $\text{RBQ}_{(0,1)}$-base in $\mathcal{F}$, the TriGen will always find a TG-modifier such that $\varepsilon_\Delta \leq \theta$.

4.4 Notes on the Triangular Inequality

As we have shown, the TriGen algorithm produces a TG-modifier which generates the triangular inequality property for a particular semimetric d. However, we have to realize the triangular inequality is generated just according to the dataset sample $\mathbb{S}^*$ (to the sampled distance triplets, actually). A TG-modification d^f being metric according to $\mathbb{S}^*$ has not to be a "full metric" according to the entire dataset $\mathbb{S}$ (or even to $\mathbb{U}$), so that searching in $\mathbb{S}$ by a MAM could become only

approximate, even in case $\theta = 0$. Nevertheless, in most applications a (random) dataset sample $\mathbb{S}^*$ is supposed to have the distance distribution similar to that of $\mathbb{S} \cup \{Q\}$, and also the sampled distance triplets are expected to be representative.

Moreover, the construction of such a TG-modifier f, for which $(\mathbb{S}, d^f)$ is metric space but $(\mathbb{U}, d^f)$ is not, can be beneficial for the efficiency of search, since the intrinsic dimensionality of $(\mathbb{S}, d^f)$ can be significantly lower than that of $(\mathbb{U}, d^f)$. The above claims are verified experimentally in the following section, where the retrieval error (besides pure ε_Δ) and the retrieval efficiency (besides pure $\rho(\mathbb{S}, d^f)$) are evaluated. Nonetheless, to keep the terminology correct let us read a metric d^f created by the TriGen as a *TriGen-approximated metric*.

5 Experimental Results

To examine the proposed method, we have performed extensive testing of the TriGen algorithm as well as evaluation of the generated distances with respect to the effectiveness and efficiency of retrieval by two MAMs (M-tree and PM-tree).

5.1 The Testbed

We have examined 10 non-metric distance measures (all described in Section 1.6) on two datasets (images and polygons). The dataset of images consisted of 10,000 web-crawled images [30] transformed into 64-level gray-scale histograms. We have tested 6 semimetrics on the images: the COSIMIR measure (denoted **COSIMIR**), the 5-median L_2 distance (**5-medL2**), the squared L_2 distance (**L2square**), and three fractional L_p distances ($p = 0.25, 0.5, 0.75$, denoted **FracLp***p*). The **COSIMIR** network was trained by 28 user-assessed pairs of images.

The synthetic dataset of polygons consisted of 1,000,000 2D polygons, each consisting of 5 to 10 vertices. We have tested 4 semimetrics on the polygons: the 3-median and 5-median Hausdorff distances (denoted **3-medHausdorff**, **5-medHausdorff**), and the time warping distance with δ chosen as L_2 and L_∞, respectively (denoted **TimeWarpL2**, **TimeWarpLmax**). The **COSIMIR**, **5-medL2** and k-**medHausdorff** measures were adjusted to be semimetrics, as described in Section 3.1. All the semimetrics were normed to return distances from $\langle 0, 1 \rangle$.

5.2 The TriGen Setup

The TriGen algorithm was used to generate the optimal TG-modifier for each semimetric (considering the respective dataset). To examine the relation between retrieval error of MAMs and the TG-error, we have constructed several TG-modifiers for each semimetric, considering different values of TG-error tolerance $\theta \geq 0$. The TriGen's set of bases $\mathcal{F}$ was populated by the FP-base and 116 RBQ-bases parametrized by all such pairs (a, b) that $a \in \{0, 0.005, 0.015, 0.035, 0.075, 0.155\}$, where for a value of a the values of b were multiples of 0.05 limited by $a < b \leq 1$. The dataset sample $\mathbb{S}^*$ used by TriGen consisted of $n = 1000$ randomly selected objects in case of images (10% of the dataset), and $n = 5000$ in case of polygons (0.5% of the dataset). The distance matrix built from the respective dataset sample $\mathbb{S}^*$ was used to form $m = 10^6$ distance triplets.

In Table 1 see the optimal TG-modifiers found for the semimetrics by TriGen, considering $\theta = 0$ and $\theta = 0.05$, respectively. In the first column, best RBQ modifier parameters (best in sense of lowest ρ depending on a, b) are presented. In the second column, the achieved ρ for a concavity weight w of the FP-base is presented, in order to make a comparison with the best RBQ modifier. Among RBQ- and FP-bases, the winning modifier (with respect to lowest ρ) is printed in bold. When considering $\theta = 0.05$, **FracLp0.5**, **3-medHausdorff**, **5-medHausdorff** even need not to be modified (see the zero weights by the FP-base), since the TG-error is already below θ. Also note that for **L2square** and $\theta = 0$ the weight of FP-base modifier is $w = 0.99$, instead of $w = 1.0$ (which would turn **L2square** into L_2 distance). That is because the intrinsic dimensionality of the dataset sample $\mathbb{S}^*$ is lower than that of the universe $\mathbb{U}$ (64-dimensional vector space).

Table 1. TG-modifiers found by TriGen.

	$\theta = 0.00$				$\theta = 0.05$			
	best RBQ-base		FP-base		best RBQ-base		FP-base	
semimetric	(a, b)	ρ	ρ	w	(a, b)	ρ	ρ	w
L2square	**(0, 0.15)**	**3.74**	4.22	0.99	**(0, 0.05)**	**2.82**	3.02	0.59
COSIMIR	**(0, 0.45)**	**12.2**	27.2	4.33	**(0.005, 0.15)**	**3.19**	3.80	0.63
5-medL2	(0, 0.1)	37.7	**19.8**	**16.5**	(0, 0.05)	4.28	**3.17**	**3.88**
FracLp0.25	**(0, 0.45)**	**12.7**	15.2	2.29	(0.035, 0.05)	3.50	**3.30**	**0.30**
FracLp0.5	**(0, 0.05)**	**7.57**	8.37	0.87	**(0, 0.2)**	**3.28**	3.34	0.06
FracLp0.75	**(0, 0.75)**	**5.13**	5.69	0.30	**any**	**3.77**	**3.77**	**0**
3-medHausdorff	**(0, 0.05)**	**3.77**	5.11	0.60	**any**	**2.28**	**2.28**	**0**
5-medHausdorff	**(0, 0.05)**	**3.42**	4.12	0.35	**any**	**2.45**	**2.45**	**0**
TimeWarpL2	(0, 0.55)	10.0	**9.48**	**1.48**	**(0.035, 0.1)**	**2.72**	2.76	0.23
TimeWarpLmax	**(0.005, 0.3)**	**8.75**	9.69	1.52	**(0, 0.1)**	**2.83**	2.86	0.26

In Figure 4 see the intrinsic dimensionalities $\rho(\mathbb{S}^*, d^f)$ with respect to the growing TG-error tolerance θ (f is the optimal TG-modifier found by TriGen).

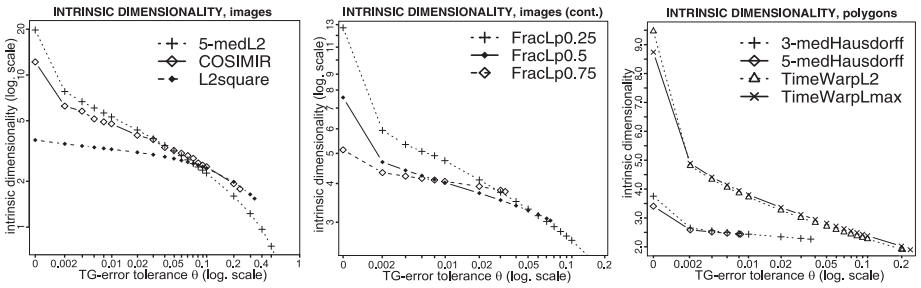

Fig. 4. Intrinsic dimensionality of images and polygons

The rightmost point $[\theta, \rho]$ of a particular curve in each figure means θ is the maximum ε_Δ value that can be reached; for such a value (and all greater) the concavity weight w becomes zero. Similar "endpoints" on curves appear also in other following curves that depend on the TG-error tolerance.

The Figure 5a shows the impact of m sampled triplets (used by TGError) on the intrinsic dimensionality, considering $\theta = 0$ and only the FP-base in $\mathcal{F}$. The more triplets, the more accurate value of ε_Δ and the more concave TG-modifier is needed to keep $\varepsilon_\Delta = 0$, so the concavity weight and the intrinsic dimensionality

grow. However, except for **5-medHausdorff**, the growth of intrinsic dimensionality is quite slow for $m > 10^6$ (and even slower if we set $\theta > 0$).

For the future we plan to improve the simple random selection of triplets from the distance matrix, in order to obtain more representative triplets, and thus more accurate values of ε_Δ together with keeping m low.

5.3 Indexing & Querying

In order to evaluate the efficiency and effectiveness of search when using TriGen-approximated metrics, we have utilized the M-tree [7] and the PM-tree [27].

For either of the datasets several M-tree and PM-tree indices were built, differed in the metric d^f employed – for each semimetric and each θ value a d^f was found by TriGen, and an index created. The setup of (P)M-tree indices is summarized in Table 2 (for technical details see [7, 26, 27]).

Table 2. M-tree and PM-tree setup

disk page size:	4 kB | avg. page utilization: 41%–68%
PM-tree pivots:	64 inner node pivots, 0 leaf pivots
image indices size:	1–2 MB (M-tree) 1.2–2.2 MB (PM-tree)
polygon indices size:	140–150 MB (both M-tree and PM-tree)
construction method:	MinMax + SingleWay (+ slim-down)

To achieve more compact MAM classes, the indices (both M-tree and PM-tree) built on the image dataset were post-processed by the *generalized slim-down algorithm* [26]. The 64 global pivot objects used by PM-tree indices were sampled among the n objects already used for the TriGen's distance matrix construction.

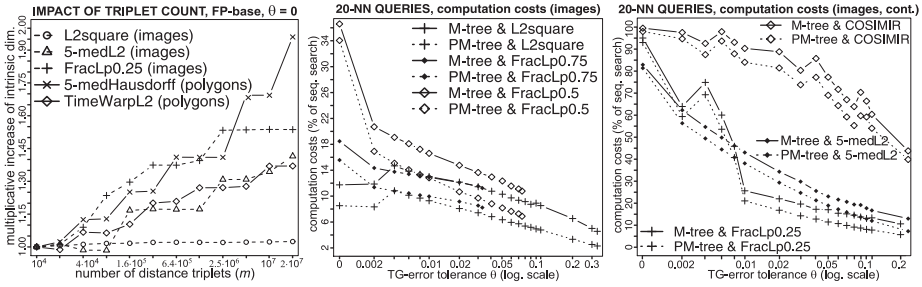

Fig. 5. Impact of triplet count; 20-NN queries on images (costs)

All the (P)M-tree indices were used to process k-NN queries. Since the TriGen-generated modifications are generally metric approximations (especially when $\theta > 0$), the filtration of (P)M-tree branches was affected by a *retrieval error* (the relative error in precision and recall). The retrieval error was computed as the *Jaccard distance* E_{NO} (or normed overlap distance) between the query result QR_{MAM} returned by a (P)M-tree index and the correct query result QR_{SEQ} (obtained by sequential search of the dataset), i.e. $E_{NO} = 1 - \frac{|QR_{MAM} \cap QR_{SEQ}|}{|QR_{MAM} \cup QR_{SEQ}|}$.

To examine retrieval efficiency, the computation costs needed for query evaluation were compared to the costs spent by sequential search. Every query was repeated for 200 randomly selected query objects, and the results were averaged.

In Figures 5b,c see the costs of 20-NN queries processed on image indices, depending on growing θ. The intrinsic dimensionalities decrease, and so the searching becomes more efficient (e.g. down to 2% of costs spent by sequential search for $\theta = 0.4$ and the TG-modification of **L2square**). On the other hand, for $\theta = 0$ the TG-modifications of **COSIMIR** and **FracLp0.25** imply high intrinsic dimensionality, so the retrieval deteriorates to almost sequential search.

In Figures 6a,b the retrieval error E_{NO} is presented for growing θ. In Figures 6c and 7a see the retrieval efficiency and error for 20-NN querying on the polygon indices. As supposed, the error grows with growing TG-error tolerance θ. Interestingly, the values of θ tend to be the upper bounds to the values of E_{NO}, so we could utilize θ in an *error model* for prediction of E_{NO}.

In case of **5-medL2**, **3-medHausdorff** (and partly **COSIMIR**, **5-medHausdorff**) indices, the retrieval error was non-zero even for $\theta = 0$. This was caused by neglecting some "pathological" distance triplets when computing the TGError function (see Section 4), so the triangular inequality was not preserved for all triplets, and the filtering performed by (P)M-tree was sometimes (but rarely) incorrect. In other cases (where $\theta = 0$) the retrieval error was zero.

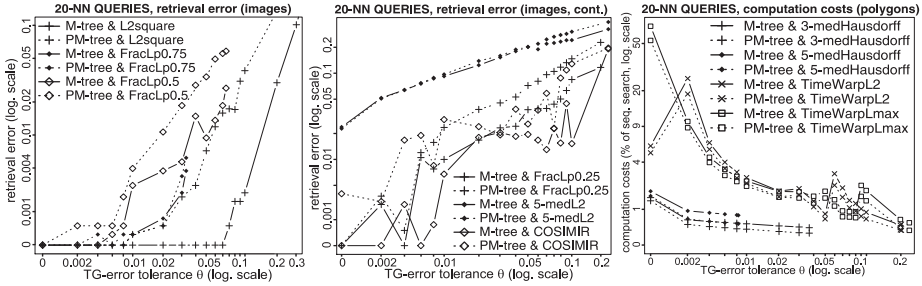

Fig. 6. 20-NN queries on images and polygons (retrieval error, costs)

The costs and the error for k-NN querying are presented in Figures 7b,c – with respect to the increasing number of nearest neighbors k.

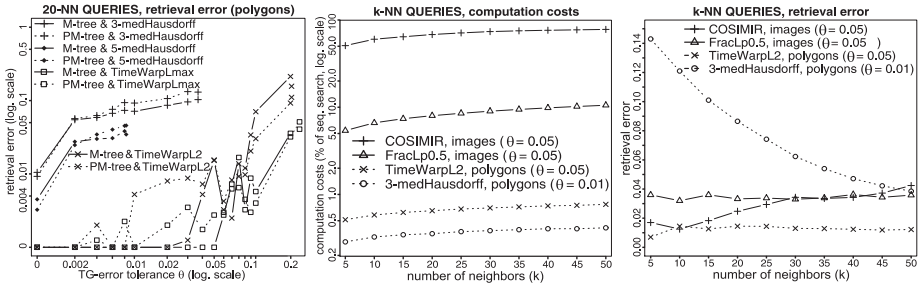

Fig. 7. 20-NN queries on polygons (retrieval error); k-NN queries (costs, retrieval error)

Summary. Based on the above presented experimental results, we can observe that non-metric searching by MAMs, together with usage of the TriGen algorithm as the first step of the indexing, can successfully merge both aspects, the

retrieval efficiency as well as the effectiveness. The efficiency achieved is by far higher than simple sequential search (even for $\theta = 0$), whereas the retrieval error is kept very low for reasonable values of θ. Moreover, by choosing different values of θ we get a trade-off between the effectiveness and efficiency thus, the TriGen algorithm provides a scalability mechanism for non-metric search by MAMs.

On the other hand, some non-metric measures are very hard to use for efficient *exact* search by MAMs (i.e. keeping $E_{NO} = 0$), in particular the **COSIMIR** and the **FracLp0.25** measures. Nevertheless, for *approximate* search ($E_{NO} > 0$) also these measures can be utilized efficiently.

6 Conclusions

In this paper we have proposed a general approach to non-metric similarity search in multimedia databases by use of metric access methods (MAMs). We have shown the triangular inequality property is not restrictive for similarity search and can be enforced for every semimetric (modifying it to a metric). Furthermore, we have introduced the TriGen algorithm for automatic turning of any black-box semimetric into metric (or at least approximation of a metric) just by use of distance distribution in a fraction of the database. Such a "TriGen-approximated metric" can be safely used to search the database by any MAM, while the similarity orderings with respect to a query object (the retrieval effectiveness) are correctly preserved. The main result of the paper is a fact that we can quickly search a multimedia database when using unknown non-metric similarity measures, while the retrieval error achieved can be very low.

Acknowledgements. This research has been supported by grants 201/05/P036 of the Czech Science Foundation (GAČR) and "Information Society" 1ET100300419 – National Research Programme of the Czech Republic. I also thank Július Štroffek for his implementation of backpropagation network (used for the COSIMIR experiments).

References

1. C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*. LNCS, Springer, 2001.
2. F. Ashby and N. Perrin. Toward a unified theory of similarity and recognition. *Psychological Review*, 95(1):124–150, 1988.
3. I. Bartolini, P. Ciaccia, and M. Patella. WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance. *IEEE Pattern Analysis and Machine Intelligence*, 27(1):142–147, 2005.
4. E. Chávez and G. Navarro. A Probabilistic Spell for the Curse of Dimensionality. In *ALENEX'01, LNCS 2153*, pages 147–160. Springer, 2001.
5. E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.
6. P. Ciaccia and M. Patella. Searching in metric spaces with user-defined and approximate distances. *ACM Database Systems*, 27(4):398–437, 2002.
7. P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *VLDB'97*, pages 426–435, 1997.

8. P. Corazza. Introduction to metric-preserving functions. *American Mathematical Monthly*, 104(4):309–23, 1999.
9. V. Dohnal, C. Gennaro, P. Savino, and P. Zezula. D-index: Distance searching index for metric data sets. *Multimedia Tools and Applications*, 21(1):9–33, 2003.
10. M. Donahue, D. Geiger, T. Liu, and R. Hummel. Sparse representations for image decomposition with occlusions. In *CVPR*, pages 7–12, 1996.
11. C. Faloutsos and K. Lin. Fastmap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In *SIGMOD*, 1995.
12. R. F. S. Filho, A. J. M. Traina, C. Traina, and C. Faloutsos. Similarity search without tears: The OMNI family of all-purpose access methods. In *ICDE*, 2001.
13. K.-S. Goh, B. Li, and E. Chang. DynDex: a dynamic and non-metric space indexer. In *ACM Multimedia*, 2002.
14. P. Hart. The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.
15. G. R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *IEEE Patt.Anal. and Mach.Intell.*, 25(5):530–549, 2003.
16. P. Howarth and S. Ruger. Fractional distance measures for content-based image retrieval. In *ECIR 2005*, pages 447–456. LNCS 3408, Springer-Verlag, 2005.
17. D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Patt. Anal. and Mach. Intell.*, 15(9):850–863, 1993.
18. D. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.
19. A. K. Jain and D. E. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Patt.Anal.Mach.Intell.*, 19(12):1386–1391, 1997.
20. O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *AVBPA*, pages 90–95. LNCS 2091, Springer-Verlag, 2001.
21. C. L. Krumhansl. Concerning the applicability of geometric models to similar data: The interrelationship between similarity and spatial density. *Psychological Review*, 85(5):445–463, 1978.
22. T. Mandl. Learning similarity functions in information retrieval. In *EUFIT*, 1998.
23. E. Rosch. Cognitive reference points. *Cognitive Psychology*, 7:532–47, 1975.
24. E. Rothkopf. A measure of stimulus similarity and errors in some paired-associate learning tasks. *J. of Experimental Psychology*, 53(2):94–101, 1957.
25. S. Santini and R. Jain. Similarity measures. *IEEE Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.
26. T. Skopal, J. Pokorný, M. Krátký, and V. Snášel. Revisiting M-tree Building Principles. In *ADBIS, Dresden*, pages 148–162. LNCS 2798, Springer, 2003.
27. T. Skopal, J. Pokorný, and V. Snášel. Nearest Neighbours Search using the PM-tree. In *DASFAA '05, Beijing, China*, pages 803–815. LNCS 3453, Springer, 2005.
28. A. Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.
29. A. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154, 1982.
30. Wavelet-based Image Indexing and Searching, Stanford University, `wang.ist.psu.edu`.
31. R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, 1998.
32. D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972.
33. B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE '98*, pages 201–208, 1998.