Where are you heading, metric access methods? A provocative survey.

Tomáš Skopal
SIRET research group
Charles University in Prague, FMP, Department of Software Engineering,
Malostranské nám. 25, 118 00 Prague, Czech Republic
http://siret.ms.mff.cuni.cz

ABSTRACT

In this paper the impact of the metric indexing paradigm on the real-world applications is discussed. We pose questions whether the priorities in research of metric access methods (MAMs) established in the past decades reflect the actual needs of practitioners. In particular, we formulate the following pragmatic questions: Are the established MAM cost measures relevant? Isn't the metric space model too general when the majority of real-world applications use L_p spaces? On the other hand, isn't the metric model too restrictive with respect to the growing community of practitioners using non-metric distances? Are the simple similarity queries competitive enough? Have the real-world similarity search engines ever used a general metric access method, or do they use specific indexing? Is there a real demand for content-based similarity search or will the annotations and keyword search win the game? We present justification of these questions, investigating relevant literature and search engines. Finally, we try to transform the questions into answers and suggestions to the future research on MAMs.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

algorithms, performance, design

Keywords

metric access methods, MAM, similarity, content-based search

1. INTRODUCTION

The content-based search in multimedia and other unstructured data becomes steadily more important nowadays, while the similarity search concept provides a general and intuitive model. Given a database of descriptors (i.e., features extracted from the original multimedia objects), a query descriptor is submitted such that objects similar to the query are returned as an answer. Hence, such content-based retrieval model separates the actual retrieval algorithm from the semantic model (the similarity of content). In a narrow sense, the pair-wise similarity function is often required to fulfill the metric postulates which are the basic properties that allow to index the database for efficient (fast) query processing. Here we talk about a class of database methods called *metric access methods* (MAMs), or *metric indexes*.

In this paper, we pragmatically analyze both, the experimental practices in the MAM research, and also the potential "market" for metric indexing – applications in content-based multimedia retrieval. Unlike other survey papers, we do not give a systematic overview of the achievements in the research area, but we try to critically discuss the weak points of MAM research. The aim of the constructive critique is to prevent the research of MAM from isolating into a bubble of theoretical and artificial achievements, and to motivate the research to a constant and tight connection with the real-world applications.

1.1 Questions

In particular, we ask the following six provocative questions, more or less rhetorical, but all aimed at impact of the MAM research on practical applications:

Q1: Isn't the metric space model too general?
Q2: Are the established MAM cost measures relevant?
Q3: Is there a real demand for general metric indexing?
Q4: Are the simple similarity queries competitive enough?
Q5: Have the real-world search engines ever used a MAM?
Q6: Isn't the metric model too restrictive?

The set of questions was not assembled by an arbitrary thought process of the author, but it came out as an impression of the analysis we will present in the following text. In addition to identifying the questions in Sections 2 and 3, in Section 4 we discuss possible answers and future solutions.

1.2 Metric access methods

Before we start the survey, we need to remember the very basic mission of the metric access methods.

$$MAM =$$
Set of algorithms and data structure(s) providing efficient (fast) similarity search under the metric space model.

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.
SISAP 2010, September 18-19, 2010, Istanbul, Turkey.
Copyright 2010 ACM 978-1-4503-0420-7/10/09 ...$10.00.

The metric space model itself is determined by its mathematical foundations. The database $\mathbb{S} \subset \mathbb{U}$ to be searched is considered as a set of unstructured (black-box) descriptors, so that only a distance function $\delta(x, y)$ is defined between any two descriptors x, y from the descriptors universe $\mathbb{U}$. The distance is required to be a metric distance, i.e., δ is non-negative, identical ($\delta(x, y) = 0 \Leftrightarrow x = y$), symmetric ($\delta(x, y) = \delta(y, x)$) and triangular ($\delta(x, y)+\delta(y, z) \geq \delta(x, z)$).

There were many metric access methods developed so far (see the literature referenced in the following section), addressing various data management aspects, namely: main vs. secondary memory index, static vs. dynamic database, exact vs. approximate search, continuous vs. discrete metric distances, centralized/serial vs. distributed/parallel implementation, etc.

2. EXPERIMENTAL PRACTICES IN MAM RESEARCH

In this section we present the results of analysis within the 40 years old database-oriented research on similarity search. In particular, we have analyzed papers cited in the major "bibles" for the MAM community – the classic survey [6], the classic monographs [26, 20] and a recent book chapter [13]. Among the hundreds of papers referenced in the mentioned sources, we selected 77 for our research, that propose *general* metric access methods and prove their contribution in an *experimental evaluation*. Hence, we did not consider distance-specific indexing methods, and also theoretical papers. In addition to the 77 papers, we analyzed also 18 relevant papers from the SISAP 2008 and 2009 conferences, giving us the total number of 95 analyzed papers.

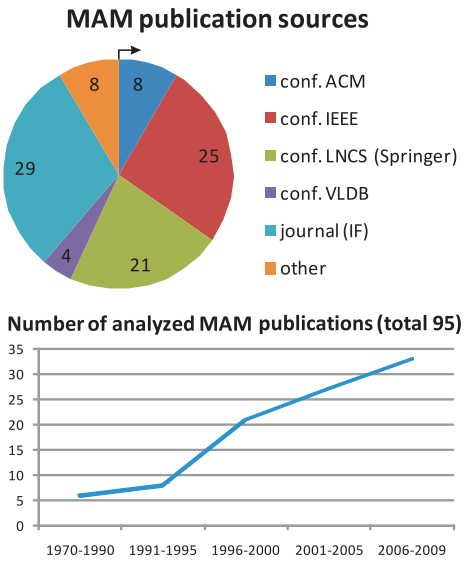

Figure 1: Analyzed experimental papers on MAMs.

The structure of the papers and their distribution in time is shown in Figure 1. We can observe that the majority of papers was published in renowned journals and in proceedings of international conferences, while the majority of the contributions was published in the past ten years. It should be also mentioned that 49.5% of the papers were co-authored by somebody from the SISAP program committee (12 people through the years 2008-2010). However, this fact does not support a possible biased paper selection, it simply points out that SISAP conference gathers a significant proportion of people interested in MAM research.

2.1 Distances & Databases

In the first part of the MAM papers analysis, we focused on the metric space instances and testbeds used in the papers' experiments. In particular, we aggregated the information about the type of space, the size of databases used, and especially the distance metric used.

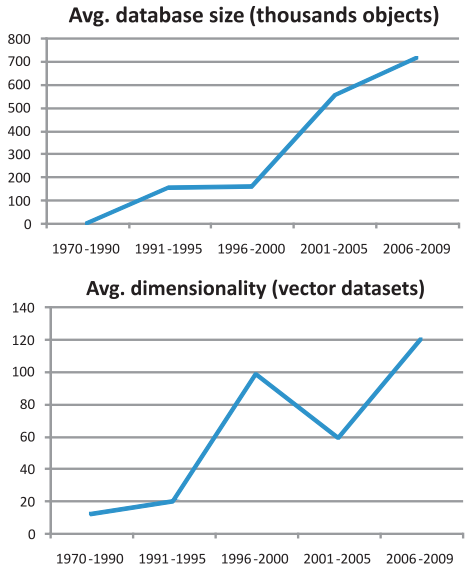

Figure 2: Databases used through the years.

In Figure 2, see the evolution of database size and dimension (when vector space used) in time. In 1970-1990 we can hardly speak about "databases", as the numbers of objects were around 1000. In the past ten years, however, the database sizes got to volumes of almost a million objects per database (on average). In such volumes the sequential search already becomes a bottleneck in the retrieval process, so the indexing efforts pay off. In case of vector spaces (equipped by a metric distance), the dimension grows from less than ten to more than one hundred (on average). This growth not only increases the volumes of databases, but also indicates growing complexity of descriptors (high-dimensional vectors).

The most interesting result of this subsection is shown in Figure 3, where the usage of different types of metric spaces is summarized. As expected, the majority of papers evaluate their contribution on vector spaces, which mostly means the Euclidean space (+ several L_∞, L_1 spaces), followed by a few others, like Hamming or angle space. Since low-level descriptors represented as vectors of independent dimensions are very popular over many domains, the employment of Euclidean distance seems natural. The second most frequent is the string space under the edit distance. The instances of string databases used are, however, far less multifarious than the Euclidean ones, as they mostly include English and Spanish vocabularies (+ several others, like biological sequences). Other types of metric spaces are quite rare, including also several non-vectorial databases (sets of elements, time series, geometries). The expensive metric

distances (i.e., of complexity $\geq O(n^2)$) mostly reduce to the mentioned edit distance, followed by several others, like Hausdorff distance, quadratic form distance, or variations on the edit distance (sequence/string alignments solved by dynamic programming)[1].

An alarming fact, that denies the common assumption that metric distances are *expensive*, is shown on the last line of the figure, saying that almost 50% papers use only cheap distances ($O(n)$) in their experiments!

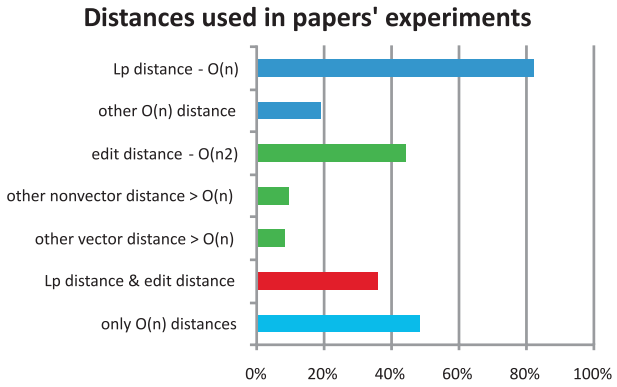

Figure 3: Distance spaces used in experiments.

Had we interpret the observations resulting from Figures 2 and 3, we could speculate about two alternative possibilities:

- The experimental part of the papers (on *general* MAM) is considered as a toy problem, while the main focus is given to the description of the method itself, rather than proving its properties on *general* metric spaces.
- The experiments in the papers are correct and reflect the actual application needs. However, that would mean there is actually only little demand for general MAM, while the attention should be given to more specific access methods, or simply indexes for L_2 or generally (combinations of) L_p distances.

Either of the two speculations indicates a problem. In the first case, the MAM research does not stick to real-world experiments in general spaces, focusing on the narrow L_2 space or edit distance stereotype. The second case is worse, as it leads to the first question of this paper, *Q1: Isn't the metric space model too general?* In other words, isn't the universality of the metric model just a technical simplification for indexing much more specific spaces, e.g., L_p?

2.2 Cost measures

In the second part of the MAM papers' analysis, we discuss the methodologies of measuring a MAM's performance (efficiency) in terms of cost types used in experiments. The overall picture is shown in Figure 5, showing the proportion of each cost type utilized in the papers' experiments. Before we analyze the consequences of Figure 5 in Section 2.2.5, we discuss all of the relevant cost types in the following sections.

2.2.1 Distance computations

From the very dawn of MAM research, the number of distance computations spent during indexing/querying (*DC cost*) has established as the most respected and frequently used performance cost. To justify the employment of DC cost, it is generally assumed that evaluation of single distance $\delta(\cdot,\cdot)$ is *computationally expensive*, so that other types of cost become marginal. The advantage of DC cost is its independence on code optimization, programming language, software and hardware platform, thus allowing to separate the essence of the proposed MAM's efficiency from the irrelevant runtime factors.

However, the assumption on expensive distance is critical, while experiments using the DC cost alone (without showing also other types of cost) are only appropriate when:

1. an expensive distance is used (i.e., having time complexity $\geq O(n^2)$ and/or large n),
2. rather small database is used (e.g., fits main memory),
3. contribution of other cost type to realtime is negligible, e.g., internal time/space cost, I/Os, networking, synchronization of parallel/distributed processing, etc.

Hence, the DC cost is a performance measure useful to analyze the qualitative behavior of the MAM (including tuning of internal parameters or comparing MAMs), rather than to be presented as the objective cost to the "end-users".

2.2.2 I/O cost

The *I/O cost*, mostly interpreted as the number of random accesses to disk pages (reads/writes), has been established in the pioneer times of database research when the hard disk drives (HDDs) were the biggest bottleneck of the data management. In particular, the research field of *spatial access methods* [4] – the closest relative to MAM research – still widely uses I/Os as the major cost type. Although the HDD technology has improved tremendously over the years, the seek time component in an I/O operation has not changed much, so the I/O cost is still relevant for methods requiring random access to disk.

However, in the context of MAM research, the I/O cost has to be used carefully. In particular, experiments using the I/O cost alone are only appropriate when:

1. I/O time dominates the other types of cost,
2. the competing MAMs share the same I/O model

The latter condition is especially important, because improper usage of I/O cost could be totally misleading. In particular, consider sequential search (as trivial MAM) and a hierarchical MAM, like the M-tree [8]. The sequential search can be easily optimized such that only single seek operation is needed to process the whole file. On the other hand, such an optimization cannot be implemented for the M-tree (without heavy disk prefetching leading to sequential search), since metric data in hierarchical indexes cannot be (natively) linearly ordered, and so random access I/Os are necessary for them.

In the following (silly) example we illustrate the danger of incorrect I/O cost usage. Let us have two 100 MB indexes (sequential file and M-tree), 4 kB disk page file system (i.e., 25,600 pages per index), seek time 8 ms and read time 50 MB/s (i.e., today low-cost HDD). Let us also suppose that an M-tree query needs to access just 1% of pages, while the sequential file needs to access, of course, 100% pages.

[1]The variable n is the size of a descriptor.

The realtime needed for M-tree is 256 I/Os, that is $0.008 \times 256 + 0.1 = 2.148$ seconds. An improper (random access) implementation of sequential search would take $204.8 + 2 = 206.8$ seconds, however, optimized (one seek) variant would take only 2.008 seconds![2]

Anyways, as the HDD technology will be soon (hopefully) defeated by the SSD technology that removes the seek time overhead, we can expect a renaissance of random access methods (and better performance of hierarchical MAMs).

2.2.3 *Internal cost*

Apart from distance computations (and I/Os), i.e., cost measures universally applicable to every MAM (managing secondary-memory index), the time/space requirements of a particular MAM could be also measured by a specific *internal* cost measure. The internal cost is not very frequently presented in the experiments, however, it could be crucial to the overall MAM's efficiency, especially when the DC cost is not dominant (i.e., cheap metric is used). To illustrate the impact of internal cost, let us discuss two examples:

- The methods based on Pivot tables, e.g., the classic LAESA [18], use a matrix consisting of distances from the database objects to a set of pivots. When a query is evaluated, the distance matrix is sequentially processed (either entire using single pass, or just a part but using multiple passes), which constitutes a significant internal overhead. For instance, consider 128 dimensional vector space under L_2 distance and 128 pivots. Then the one-pass LAESA query processing of distance matrix under L_∞ distance is equivalent to the sequential search in the original L_2 space. Here the DC cost, dramatically higher for the sequential search, is not appropriate due to the internal cost of LAESA.

- The incremental kNN algorithm [15] by Hjaltason and Samet can be implemented in any MAM. Although the algorithm was proved as optimal in terms of DC cost (i.e., equivalent to range query with radius equal to the distance to the kth neighbor), it suffers from high internal cost. Specifically, the algorithm utilizes a heap that contains the set of not-yet-processed regions of a MAM's index. Depending on the intrinsic dimensionality of the space [5] and the MAM used, the heap is usually largely inflated until the first neighbor is found, followed by a rapid heap reduction.

2.2.4 *Realtime cost*

Finally, we get to the very objective type of computation cost – the "dirty" *realtime cost* (or wall-clock time), measured in seconds or processor cycles spent in the MAM's process. The realtime cost is not very popular in analyzing the MAMs' efficiency because it is hardware-, platform-, language- and compiler-dependent, it requires proper optimizations of the code, etc. It mixes many different costs into a single aggregate, making hard to recognize the underlying causes of a MAM's (in)efficiency.

On the other hand, only the realtime cost is the moment of truth for an end user that wants to be oriented in the jungle of various MAMs. The realtime cost means "no cheating is allowed" – a MAM is either fast or slow (given a particular database context). To demonstrate a possible discrepancy between the DC cost and realtime, see an experiment in Figure 4, where a database of peptides (pieces of proteins) was indexed as 32-dimensional vectors under a linear variant of the Hausdorff distance (intrinsic dim. ≈ 3). The left-hand figure shows an expected superior performance of Pivot tables (LAESA) in terms of DC cost, being on 5% of sequential search. However, in terms of realtime (right-hand figure) the M-tree and even the sequential search run faster for databases over 1.5 million objects (2.5, resp.), due to the internal cost of Pivot tables and the cheap distance.

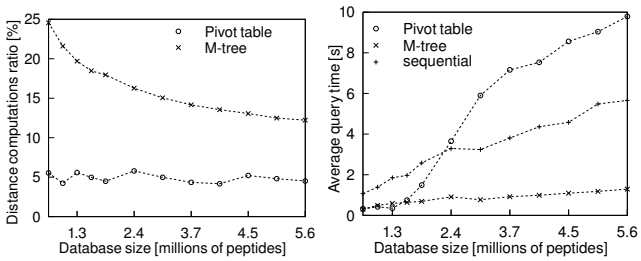

Figure 4: Example: DC cost vs. realtime

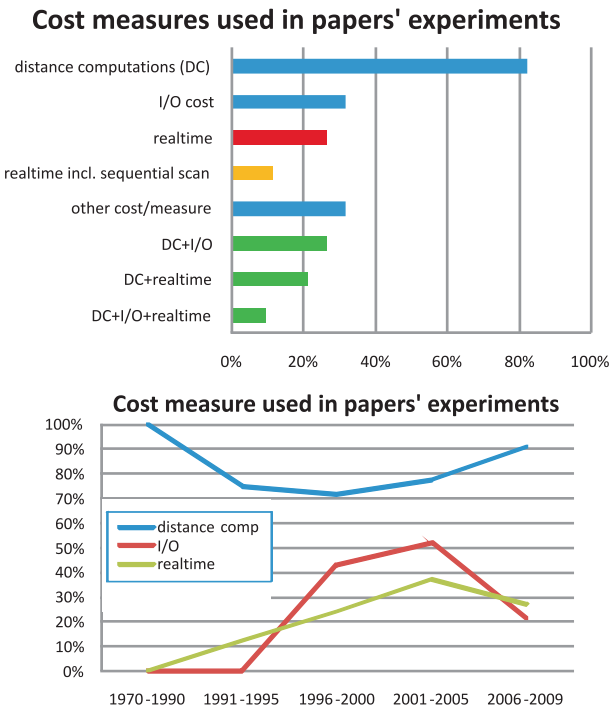

Figure 5: Cost measures used in experiments.

2.2.5 *Discussion*

In Figure 5 see the structure of cost measures used in the examined papers' experiments. Among other, it shows the DC cost was the most popular (in more than 80% papers), the realtime cost was used in 25% of papers and all costs (DC+I/O+realtime) were used in 10% papers. Some 12% papers provided a realtime comparison with the sequential search, which is seemingly an unimportant detail, but it prevents from commending an expensive MAM over even more

[2]Note that here we consider just the I/O cost contribution to realtime, while the overall realtime could be significantly different (due to possibly expensive distance computations).

expensive competitors as "really fast".

In summary, the structure of cost measures used in the papers is quite rich, however, it is not very convincing for the practitioners that need a MAM for their application. This is documented not only by the relatively small number of papers using realtime, but also by the alarming observation that 21% papers used only DC cost and, at the same time, only cheap ($O(n)$) distances in their experiments! Hence, we end up the discussion with the second question of this paper, *Q2: Are the established MAM cost measures relevant?*

3. APPLICATIONS

In this section, we leave the area of "intra-MAM" research, and move to applications that are expected to profit from metric indexing (in the future). In particular, we investigate trends in content-based image retrieval, the existing multimedia search engines, and the recent attempts to employ similarity measures more general than metric distances.

3.1 Content-based image retrieval

As a representative summary of the advances in content-based image retrieval (CBIR), we chose the respected survey by Datta et al. [10], referencing almost 300 papers related to CBIR. Apart from presenting the recent feature extraction techniques, belonging more to Computer vision than to the area of Image retrieval, we summarize several observations interesting for applications of MAMs in CBIR.

3.1.1 Modeling vs. indexing in CBIR

The situation in CBIR could be uncovered slightly by two citations from the survey: *"... we do not have yet a universally acceptable visual model for content-based search..."*, and *"... the indexing techniques were largely overshadowed by research on similarity modeling ..."*

The good news for MAM applications in CBIR is the uncompromising interpretation of the content-based retrieval as a similarity search based on image features. Hence, there is a clear distinction between the model (similarity function) and retrieval algorithm (an access method). The bad news for MAMs is the indexing for efficient search was either not solved at all (implicit sequential search), or the semantic model itself was prepared to reuse a well-known indexing technology. Following the latter way, a popular concept is an automatic annotation of the image content (including even more popular segmentation) by tags/keywords. The images described by tags are subsequently indexed using the well-known boolean or vector model of information retrieval [2], leading to inverted files at the implementation level. At this moment we come out with the third question of this paper, *Q3: Is there a real demand for general metric indexing?*

Nowadays, MAMs suffer from many limitations that prevent their straightforward (naïve) utilization in CBIR. Although the metric space model is quite general, from the complex CBIR point of view the metric indexing brings many obstacles. In particular, MAMs rarely allow modifications to the metric space during their indexes' lifetime, making hard to learn/tune the similarity measure, to rearrange the structure of descriptors, or to include user preferences. Furthermore, as metric distances must fulfill the triangle inequality, they are limited in measuring *local* similarity that usually leads to nonmetric behavior (discussed in Section 3.3). Nevertheless, even if MAM will not become the core of image retrieval techniques, its role in CBIR could be substantial (as discussed later in Sections 3.1.3 and 4.2).

3.1.2 Similarity measures

When modeling the distance space in CBIR, the complexity is more propagated into the descriptor semantics (see also the later discussion in Section 4.1), rather than into the distance measure. In turn, the most popular distances measuring similarity in CBIR are the usual Euclidean or L_1 distance, statistical (non)metric distances (e.g., Kullback-Leibler divergence), while some approaches use more expensive distances measuring histogram similarity, like quadratic form distance or earth mover's distance. As in the previous section, also these observations do not indicate inevitable benefits of general metric indexing.

Fortunately, when taking another citation from the survey, *"...the richness in the mathematical formulation of signatures (descriptors) grows alongside the invention of new methods for measuring similarity ..."*, the need for more sophisticated similarity measuring could lead to more expensive distance measures that would require general (metric) indexing.

3.1.3 Retrieval models

Had we classify the retrieval models used in CBIR, we could distinguish three design levels:

- **Pseudo-CBIR** – proprietary add-ons of text-based image search engines (surveyed in Section 3.2). The usual search of images based on keywords extracted from the surrounding web page is augmented by a limited CBIR functionality. For example, the images are additionally labeled by tags representing certain extracted features, like "contains face", "is illustration", "mostly red color", etc. At query time, the user can select some of the tags as an additional filter to the keyword query. There is no room for MAMs at all.

- **Single-model similarity search** – a true content-based search, where the retrieval procedure is based on similarity search using single-descriptor representation and single distance. Although this model would position a "simple" MAM into the prominent role of the core technology inside a CBIR system, it is not very likely to happen due to the limitations mentioned in Section 3.1.1. On the other hand, MAMs could be utilized for particular tasks, as suggested in Section 4.2.

- **Hybrid-model similarity search** – a true content-based search, where the complex retrieval procedure is split into a hierarchy of simple similarity searches. In particular, an image is represented by multiple local subdescriptors (e.g., image segments), where each subdescriptor could be modeled in its own distance space. A query image is modeled the same, so that multiple similarity searches are performed for a single query. The obtained intermediate results (ordered lists of subdescriptors) are finally ranked by an aggregating function, e.g., the top-k operator [12] or reranking [16] based on user preferences/feedback, etc. The MAMs could be utilized in the separate local searches, provided the local distances are metric and static. Here the above mentioned incremental kNN algorithm by Hjaltason and Samet is suitable due to unknown k required by the aggregating function.

Based on the observations discussed in this section (i.e., tags-based search, multiple local searches + aggregation), we come with the fourth question of the paper,
Q4: Are the simple similarity queries competitive enough? In other words, should the MAM research focus also on a native support of more complex similarity queries than the simple range/kNN?

3.2 Search engines

After the more or less academic discussion on MAMs in CBIR systems, the following analysis investigates the impact of MAMs on real-world engines. However, because most of the engines were not much documented and/or patented, this part of the paper should be considered with caution.

3.2.1 Mainstream multimedia search engines

At first, we have focused on 32 mainstream web sites providing multimedia retrieval, including search engines, hosting servers, and stock servers.

The *multimedia search engines* do not constitute standalone solutions, they are rather add-ons extending the classic web search engines. In particular, we considered web sites for image search (Google Image Search, Bing Image Search, AllTheWeb, PicSearch), video search (Bing Video Search, Lycos, AOL Video Search, SearchForVideo, BlinkX) and audio search (KaZaA, FindSounds, Skreemr, Yahoo Music Search). In addition to search engines, we included also hosting servers for images (Flickr, PhotoBucket, ImageShack, Google Picasa, DeviantArt) and videos (YouTube, DailyMotion, Yahoo Video, MySpace, MetaCafe, Google Video, MSN Video). Finally, we included major (micro)stock servers (Corbis, Getty, iStockPhoto, ShutterStock, Fotolia, DreamsTime, Alamy, Veer) that offer a paid multimedia content (image, video, audio, vector, flash) to professional designers.

Among all the listed web sites, just 7 (Google, Bing, PicSearch, FindSounds, Flickr, Picasa, ShutterStock) support a kind of content-based retrieval. However, only FindSounds supports true similarity search (though information on the similarity and index is not available), while the rest of the sites provide a kind of Pseudo-CBIR (see Section 3.1.3).

3.2.2 Content-based image retrieval engines

In order to increase the number engines providing similarity search, we have analyzed CBIR systems listed at Wikipedia [24], namely, Elastic Vision, Gazopa, Imense, Imprezzeo, Incogna, Like.com, MiPai, idee Visual Search Lab, Empora, Shopachu, TinEye, Tiltomo, eBay More Like This, ALIPR, Anaktisi, BRISC, Caliph & Emir, CIRES, FIRE, GNU Image Finding Tool, ISSBP, img(Rummager), imgSeek, IKONA, MUVIS, PIRIA, RETIN, Retrievr, SIMBA, TagProp, MUFIN. Among the 29 engines, 25 use similarity search concept, while 7 of them certainly use a metric distance (for the rest of engines the information was not available.) Only MiPai and MUFIN were identified as MAM-based. Anyways, as there is not much evidence (or even promotion) that current content-based search engines use MAMs, we ask the fifth question of this paper,
Q5: Have the real-world search engines ever used a MAM?

3.3 Beyond the metric space model

As pointed out in a recent survey [22], the playground for similarity search is much larger than the usual area of multimedia retrieval. In particular, similarity search tasks become even more common in areas like biometric databases, various scientific databases (bioinformatics, chemoinformatics, medical data), social networks, etc. Moreover, it was shown that domain experts develop constantly more complex similarities that have to reflect higher demands on retrieval effectiveness, leaving the simple distances like L_p metrics or edit distance. The new complex distances are often being generalized in order to become better parameterizable for a given domain. Due to such extensive similarity modeling, the new distances often lose their closed form (i.e., concise mathematical formula) and become heuristic algorithms. In consequence, the more complex distance, the more likely it will violate the metric postulates, so it becomes a *nonmetric*. As an example we name the (nonmetric) Smith-Waterman alignment [23], generalizing the edit distance to better model functional similarity of proteins (including scoring matrices, local alignment and gap penalizations).

The domain experts often do not care whether their distance is a metric or is not, because their similarity search tasks are usually not (yet) large-scale and the sequential search is sufficient for them at the moment. On the other hand, in case a model gets matured in the particular domain ("surviving" a certain period), the demand for better scalability could reach a higher priority, so that a kind of nonmetric indexing will be needed. Apparently, the MAMs cannot be directly utilized here, as they require metric distances. This observation leads us to the last question of the paper, *Q6: Isn't the metric space model too restrictive?*

3.3.1 "Metric nonmetric" indexing

Nevertheless, there appear transformational approaches that put the MAMs back into the game also for the purpose of nonmetric similarity search. This could bring fascinating opportunities for indexing by similarity, not yet discovered by the database community. In particular, the proposed TriGen algorithm [21] constitutes a mapping of a semi-metric space into an (approximation of) metric space, so that MAMs can be used without limitations. The mapping is achieved by finding suitable concave function, so that the triangle inequality becomes satisfied while the intrinsic dimensionality of the new space is kept as low as possible.

3.3.2 Alternative indexing

There appear also alternative approaches that completely abandon the metric space model and propose a qualitatively different mathematic formalism for general similarity indexing. For example, the recently introduced concepts of *ptolemaic indexing* [14] (replacing triangle inequality by ptolemaic inequality) or *indexing fuzzy similarity* [11] (replacing metric properties by fuzzy logic operators) represent the first attempts to natively nonmetric indexing.

4. DISCUSSION AND SUGGESTIONS

Based on the observations summarized in the previous sections, in the following text we give some suggestions to the future MAM research from the application point of view. Our aim is to strengthen the competitiveness of metric access methods in the context of content-based retrieval, by addressing the questions formulated in this paper.

As a leitmotif, an active attitude of the MAM community to domain problems is necessary, in order to achieve a larger success of MAM research in "enterprise" software ap-

plications. Hence, in addition to the "formal" experiments, (some of) the MAM proposals should dive into the real-world problems, showing that MAMs can really contribute to the performance of complex data management.

4.1 Balancing the model complexity

To address the questions Q1, Q2, Q6, the structure of the similarity model complexity deserves an increased attention. The formulation of questions Q1 and Q2 was motivated by the almost exclusive use of cheap $O(n)$ metric distances (mostly Euclidean vector spaces) and by the closely related problem of cost measures relevancy. In the following text, we discuss the two conceptual possibilities when modeling a similarity – either a low-level descriptor space and a complex distance, or a high-level descriptor and a simple distance.

4.1.1 Complex distance + low-level descriptor

The most promising opportunity for MAMs would be seeking for applications that require *complex* and *expensive* metric distances. The advantages for MAMs are two-fold, first, for complex metric distances (often non-vectorial) the alternative indexing methods (e.g., spatial access methods) cannot be efficiently employed, and second, for expensive distances the DC cost (being the MAMs' optimization priority) becomes relevant due to the negligible contribution of the other cost types.

From the semantic point of view, this "complex distance" concept assumes most of the retrieval logic lies inside a complex distance function, while the descriptor is large and contains rather low-level (raw) features produced by some elementary feature extraction procedure. In fact, the complex distance algorithm is supposed to finish the feature extraction at the moment of distance evaluation, however, during the evaluation also the second descriptor is available. Hence, such "online" feature extraction is able to integrate both descriptors into the process, allowing thus their better comparison.

As an example, we could consider the time series matching using the dynamic time warping (DTW) distance [17]. Instead of applying just the Euclidean distance on the time series, the DTW distance at first finishes the feature extraction by aligning the closest value pairs between the two series, while the resulting Euclidean distance is computed on this optimal alignment.

Unfortunately, as the complex and expensive distances are often not metrics, a preprocessing step that maps the nonmetric space into metric space, e.g., the TriGen algorithm, is needed (just the case of the DTW distance). It is also questionable, whether the "cleverness" of the complex distance could pay off the computational expensiveness (when compared with the opposite approach, that follows).

4.1.2 Simple distance + high-level descriptor

Nowadays, it seems the "complex distance" concept is less popular than the inverse concept that supposes a *simple* and *cheap* distance. From the semantic point of view, the "simple distance" concept aims to put the essence of the retrieval logic right into the descriptors. Hence, the descriptor (often vector) contains rather high-level features produced by a sophisticated feature extraction procedure[3]. The distance is "degraded" to simple aggregation of internal distances within the particular descriptor features. In most cases, this leads to the popular model of vector space with non-correlated dimensions + an L_p distance. Apparently, in the "simple distance" concept the position of MAMs is not as advantageous as in the "complex distance" concept.

To demonstrate the properties of both concepts in the same domain, we consider, again, the example of time series matching, but now using the "simple distance" concept [25]. Instead of low-level features (e.g., the time series itself), the time series could be modeled as a linear combination (or concatenation) of some representative subseries. Hence, the time series becomes a high-level vector modeled in space of subseries, while the Euclidean distance is used as similarity.

4.1.3 Complex vs. simple distance

When reasoning pragmatically, the "simple distance" concept promises more benefits for the practitioners (which is rather bad news for MAMs). In particular, the increased cost needed for the high-level feature extraction procedure is amortized within the frequently repeated search by cheap distance. Here we can see a motivation similar to the very purpose of indexing, where an expensive one-shot indexing phase is paid by multiple efficient searches.

Nevertheless, to give MAMs a better prospect, we can formulate the following question. Can always be the model complexity put into "canonized" descriptors within the "simple distance" concept, or do there exist (important) problems requiring inherently a complex distance? A possible positive answer to this question is suggested in Section 4.3.

4.2 MAMs in search engine architectures

In order to address the questions Q3, Q4, Q5, in the following section we discuss the role of MAMs in various architectures of similarity search engines (as categorized in Section 3.1.3).

4.2.1 MAM as single-model engine

As the single-model engine assumes single-descriptor space under a complex similarity, a MAM could be used as the core technology, provided the similarity is (mapped to) a metric distance. Although the single-model engines have clear semantics of the search (described by a rigorous model), they are quite limited in flexibility as discussed in Section 3.1.1. Hence, because there is not much room for adjusting the distance function after the indexing phase, the expressive power of the retrieval could be increased by offering a larger portfolio of similarity queries.

In addition to the simple kNN/range queries that allow just two parameters (a single example + a radius or k), there have been more complex query types proposed, providing more detailed query specification, yet remaining fully consistent with the single model (e.g., multi-example queries [9], metric skylines [7]).

4.2.2 MAM as a part of hybrid-model engine

In the hybrid-model engine, a MAM is nested deeper in the architecture (say, at "middleware" level), but still representing the most important part. It combines several similarity searches into an aggregated output, providing thus more flexible retrieval. As an example, we mention a shape retrieval method [3], where multiple queries are performed on M-tree indexes, while the results are finally aggregated by a nonmetric ranking.

[3]To be complete, there are many approaches (even the majority?) using simple distance *and* low-level descriptors.

Note the final aggregation could not only combine output of several metric indexes, but it can incorporate also results of searches that are not content-based, e.g., the popular keyword search.

4.2.3 MAM as a tool

In some implementations of similarity-search engines, the MAMs cannot compete with specific indexing models. For example, consider a CBIR based on the model of visual words [19], where the descriptor of an image is modeled as a sparse 10^6-dimensional vector (i.e., million of visual words). The dimensions of the vector correspond to tf-idf weights of visual words, which is a concept adopted from the vector model of information retrieval [2]. The distance between two vectors is evaluated as the well-known cosine similarity. It is also well-known that for searching a collection of sparse vectors under cosine similarity, the inverted file is extremely efficient (due to only traversing the lists corresponding to nonzero weights in the query vector). Any MAM in such an extremely high-dimensional space is condemned to fail.

When creating the image descriptor, for each image segment[4] (for its 128-dimensional SIFT vector, respectively) the most similar visual word (also SIFT vector) is found, while all the visual words are organized in a vocabulary. Hence, the database and query images are transformed into the space of visual words, using the nearest neighbor search (under L_2 or L_1 distance) in the vocabulary. Since the vocabulary (million dense 128D vectors) needs to be efficiently searched, an index is necessary, while this is an opportunity for MAMs. Hence, the MAM within the CBIR engine could not only provide the retrieval of images, it could serve as a particular tool speeding the descriptor preparation (the feature extraction, respectively). Moreover, the role of MAM as a tool is not limited to retrieval engines, as it could be applicable in other areas, such as multimedia data mining.

4.3 Bidirectional motivation

We end up the discussion with a highly speculative meditation on how to bring closer the interests of MAM research and domain-specific research. Apparently, the worlds of databases and various applied sciences requiring management of data are separated. The gap caused by different concerns of each world is even magnified by different terminology, where for a database researcher it is often difficult to identify a possible similarity function within a proprietary retrieval algorithm. The popular BLAST method used in proteomics research could be an example [1], where measuring the similarity of protein sequences is mixed with the access method (a search tree).

Had we establish a picture of usual thinking stereotype of a domain expert when modeling a content-based retrieval technique, we could consider two variants (see Figure 6 top).

4.3.1 All-in-one stereotype

In the first one, the domain expert does not distinguish between the content-based semantics and the access method (e.g., the BLAST example), which turns out in a monolithic retrieval solution (usually a heuristics without a rigorous formal model)[5].

[4]An image is segmented into more than 3000 segments, so we get 3000 nonzero weights per each million-dimensional representation of an image (i.e., 99.7% vector sparsity).

[5]Actually, BLAST aims to approximate the similarity model of Smith-Waterman alignment, but it is still a heuristics.

4.3.2 Separated similarity + sequential search

Second, the expert views the retrieval task as sequential search, where a similarity function is used to check the relevancy of database instances against the query instance. Hence, this variant is more suitable for a database application, as the sequential search could be replaced by a more efficient access method, for example, a MAM.

4.3.3 Modeling augmented by indexing

The previous variant assumes one-directional motivation, where a database research is motivated by an already formulated domain-specific retrieval problem. However, because the domain expert considered a future efficient access method only as optional, he/she naturally tended to design the retrieval problem as simple as possible, in order to minimize the cost used by sequential search. In consequence, this thinking stereotype pressurizes the domain expert to employ only cheap distances that are not sophisticated (exhibiting low precision and recall in the retrieval).

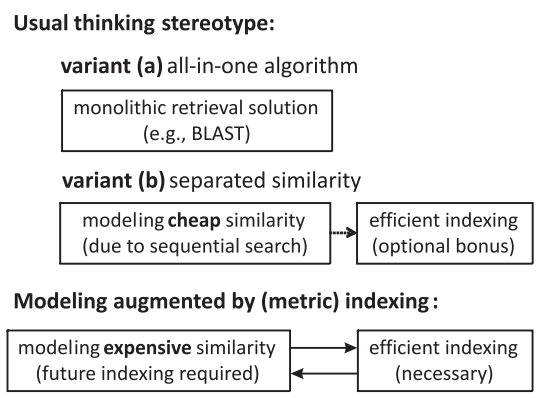

Figure 6: Bidirectional motivation.

Thus, we suggest a *bidirectional-motivation* thinking (see also Figure 6 bottom), that requires both worlds to tightly cooperate. The idea is based on "modeling augmented by indexing", where the domain expert banks on necessary indexing in her/his retrieval task. Hence, the prior knowledge of faster than sequential search enables the expert to model the similarity more generously, using expensive (metric) distance functions. The application of MAMs within such a framework is obvious, while the expensiveness of metric distances increases the likelihood of MAMs' success.

5. CONCLUSIONS

In this paper we have discussed benefits and the impact of metric access methods (MAMs) on the real-world applications and search engines. We asked six questions related to the correctness of intra-MAM research and to the applicability of MAMs in content-based retrieval. In the broad context of content-based retrieval, we have suggested that MAMs have to fight for their success, as waiting for an impulse of demand from outside the database community appears as rather naïve. Anyways, despite the scepticism intentionally (provocatively) invoked throughout the paper, we believe the metric access methods have solid foundations that promise successful applications of MAMs in many domains.

5.1 One more provocation at the end

At the very end, we would like to mention the topic of correct experimental comparison, however, a detailed analysis would deserve a standalone survey. Because the number of papers on various MAMs grows to a substantial volume, the credibility of experimental results needs to increase as well. While in the "ancient" times of only several papers on MAMs a particular result was rather easy to verify, nowadays, in the multitude of proposals such a verification is not easy due to increasing MAMs' complexity. At the same time, we often read claims that "our method beats the competitors by an order of magnitude", so one has a feeling that the power of modern MAMs, being transitively "by many orders of magnitude faster than the others", is almost infinite. Unfortunately, this is not true, and it points to the importance of correct experimentation, including *repeatable experiments* (renowned testbeds and algorithms, e.g., the SISAP library), *measuring realtime* (see Section 2.2), and *fair comparison* (optimization of the competing algorithms and not twisting the experimental setup to handicap the others).

Acknowledgments

This research has been supported in part by Czech Science Foundation project Nr. 201/09/0683.

6. REFERENCES

[1] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing, 1999.

[3] S. Berretti, A. D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, 2000.

[4] C. Böhm, S. Berchtold, and D. Keim. Searching in High-Dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases. *ACM Computing Surveys*, 33(3):322–373, 2001.

[5] E. Chávez and G. Navarro. A Probabilistic Spell for the Curse of Dimensionality. In *ALENEX'01, LNCS 2153*, pages 147–160. Springer, 2001.

[6] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[7] L. Chen and X. Lian. Efficient processing of metric skyline queries. *IEEE Trans. on Knowl. and Data Eng.*, 21(3):351–365, 2009.

[8] 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.

[9] P. Ciaccia, M. Patella, and P. Zezula. Processing complex similarity queries with distance-based access methods. In H.-J. Schek, F. Saltor, I. Ramos, and G. Alonso, editors, *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings*, volume 1377 of *Lecture Notes in Computer Science*, pages 9–23. Springer, 1998.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[11] A. Eckhardt, T. Skopal, and P. Vojtáš. On fuzzy vs. metric similarity search in complex databases. In *Proc. 8th Conference on Flexible Query Answering Systems (FQAS'09)*, volume 5822 of *LNAI*, pages 64–75. Springer, 2009.

[12] R. Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[13] M. L. Hetland. The basic principles of metric indexing. In *Swarm Intelligence for Multi-objective Problems in Data Mining*. Springer, 2009.

[14] M. L. Hetland. Ptolemaic indexing. *CoRR*, abs/0911.4384, 2009.

[15] G. Hjaltason and H. Samet. Incremental similarity search in multimedia databases, Comp. Science Dept. TR-4199, Univ. of Maryland, College Park, 2000.

[16] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Reranking methods for visual search. *IEEE Multimedia*, 14:14–22, 2007.

[17] E. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos. LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 882–893. VLDB Endowment, 2006.

[18] M. L. Mico, J. Oncina, and E. Vidal. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recogn. Lett.*, 15(1):9–17, 1994.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition. CVPR '07*, pages 1–8, 2007.

[20] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[21] T. Skopal. Unified framework for fast exact and approximate search in dissimilarity spaces. *ACM Transactions on Database Systems*, 32(4):1–46, 2007.

[22] T. Skopal and B. Bustos. On Nonmetric Similarity Search Problems in Complex Domains. *ACM Computing Surveys 44(3)*, 2012, issue tentative, available at http://siret.ms.mff.cuni.cz/skopal/pub/nmsurvey.pdf.

[23] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[24] Wikipedia. List of content-based image retrieval engines `http://en.wikipedia.org/wiki/List_of_CBIR_engines`, June 16, 2010.

[25] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956, New York, NY, USA, 2009. ACM.

[26] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.