Data Management for Dynamic Multimedia Analytics and Retrieval

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Ralph Marc Philipp Gasser

Basel, 2022

Zusammenfassung

Abstract

Acknowledgements

Good luck.

This work was partly supported by the Swiss National Science Foundation, which is also thankfully acknowledged.

Contents

List of Figures

List of Tables

Part I

Introduction

1

Introduction

The term *multimedia* describes the combination of different forms of digital media – also called *modalities* – into a single, sensory experience that carries a higher level semantic. Those modalities include but are not limited to images and videos (visual), music, sound effects and speech (aural) or textual information. However, more exotic media types such as 3D models or signals produced by sensors can also be seen as modalities, even though experience by a human consumer may depend on pre-processing by specialized hard- and software.

Nowadays, people encounter digital media and multimedia on a daily basis when watching videos on Netflix or YouTube, when listening to music on Spotify or when browsing a private image collection on their laptop. (Multi-)media content makes up a large part of today's Internet and constitutes a major driving force behind its growth, as both volume and variety increases at an ever increasing pace. An important contributing factor are social media platforms, where users act both as consumers and producers of digital content. Current estimates suggest, that there are roughly 4.66 billion active Internet users worldwide, of which 4.2 billion can be considered active social media users[1]. Facebook alone contributed to 144 thousand uploaded images per minute in 2020. And many more of these platforms, such as *Instagram* or *Twitter*, serve millions of users with mixed, self-made content involving text, images, videos or a combination thereof. A similar study found, that by 2025 we will produce a yearly amount of 175 Zettabytes (i.e, 10^{21} bytes) worth of data[2].

Looking at these numbers, the need for efficient and effective tools for *managing*, *manipulating*, *searching*, *exploring* and *analysing* multimedia data corpora becomes very apparent, which has given rise to different areas of research.

[1] Source: Statista.com, "Social media usage worldwide", January 2021

[2] Source: Statista.com, "Big Data", January 2021

1.1 Working with Multimedia Data

On a very high level, multimedia data collections consist of individual multimedia items, such as video, image or audio files. Each item, in turn, comprises of *content*, *annotations* and *metdata*. Unlike traditional data collections that contain only text and numbers, the content of the multimedia item itself is unstructured on a data level, which is why *feature representations* that reflect a media item's content in some way and that can be handled by data processing systems are required [ZW14]. Traditionally, such feature representations have often been numerical vectors $f_i \in \mathbb{R}^d$. However, in theory, any mathematical object that can be processed by a computer can act as a feature.

It is important to emphasize, that a media item can comprise of all of the aforementioned components and that a multimedia collection may contain items of different types. Furthermore, when looking at a media item's lifecycle, all of the aforementioned aspects are not static; annotations, metdata and features may either be generated upon the item's creation (e.g., for technical metadata), as a result of data-processing and analysis or by manually adding the information at some stage. Hence, any data management system must be able to cope with changes to that information. Those requirements are formalized in Section 4.1.

1.1.1 Multimedia Analysis

Multimedia analysis has its roots in *computer vision* and *pattern recognition*, which started in the early 1960s and deal with the automated, computer-aided analysis of visual information found in images and later videos. In the early days of computer vision, a lot of effort went into the engineering of feature representations that captured certain aspects of a media item's content, such as the colour distribution, texture or relevant keypoints [Low99; BTVG06] in an image. Once such features have been obtained, they can be used to perform various tasks such as classification, clustering or statistical analysis. With the advent of deep learning, the extraction of such features could largely be automated through neural network architectures such as the *Convolutional Neural Network (CNN)* and sometimes even be integrated with the downstream analysis [GBC16].

Obviously, such analysis is not restricted to the visual domain and can be applied to other types of media such as speech, music, video or 3D models with specific applications, such as, speech recognition, audio fingerprinting in music, movement detection in videos or classification of 3D models, all of which fall into the broader category of multimedia analysis.

1.1.2 Multimedia Analytics

Multimedia analytics aims at generating new knowledge and insights from multimedia data by combining techniques from multimedia analysis and visual analytics. While multimedia analysis deals with the different media types and how meaningful representations and models can be extracted from them, visual analytics deals with the user's interaction with the data and the models themselves [CTW+10; KKE+10]. Simply put, multimedia analytics can be seens as a back and forth between multimedia (data) analysis and visual analytics, wheras analysis is used to generate models as well as visualisations from data which are then examined and refined by the user and their input. This is an iterative process that generates new knowledge and may in and by itself lead to new information being attached to the multimedia items in the collection.

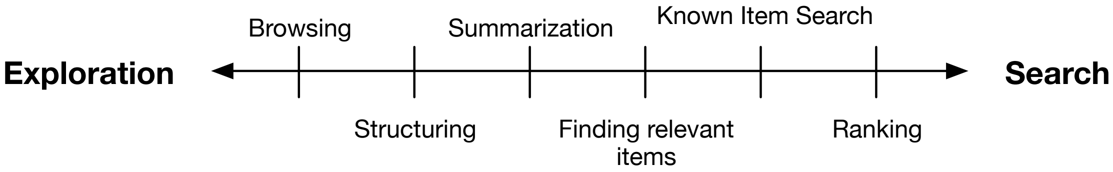

Figure 1.1 Exploration-search axis of multimedia analytics [ZW14].

For analytics on a multimedia collection, Zahalka et al. [ZW14] propose the formal model of an *exploration-search axis*, which is depicted in Figure 1.1. The model is used to characterize the different types of tasks carried out by the user. The axis specifies two ends of a spectrum, with *exploration* marking one end – in case the user knows nothing about the data collection – and *search* marking the other end – in case the user knows exactly which specific items of a collection they're interested in. During multimedia analytics, a user's activities oscillate between the two ends of the spectrum until the desired knowledge has been generated. Unsurprisingly, all of the depicted activites come with distinct requirements on data transformation and processing.

As data collections become large enough for the relevant units of information – i.e., feature representations, annotations and metadata – to no longer fit into main memory, multimedia analytics and the associated data processing quickly becomes an issue of scalable data management [JWZ+16]. This data management aspect becomes very challenging when considering the volume and variety of the multimedia data, the velocity at which new data is generated and the inherently unstructured nature of the media data itself.

1.1.3 Multimedia Retrieval

Traditionally, multimedia retrieval or content-based retrieval could be seen as a special niche within the multimedia analysis domain. It constitutes a dedidcated field of research that deals with searching and finding items of interest within a large (multi-)media collection. Even though this may sound like the main function of a database, it is a very different task for multimedia than it is for structured data [BdB+07]. On the one hand, given the structure of a relational database and languages like SQL, a user can specify exactly what elements from the database should be selected using predicates that either match or don't match the items in a collection. For example, when considering a product database that contains price information for individual items, it is trivial to formulate a query that selects all items above a specific price threshold.

Retrieving multimedia data, on the other hand, comes with indirections due to the unstructured nature of the content, the feature representations used as a proxy for it and the *semantic gap* associated with these representations. A very popular model to work with the feature representations involves calculation of (dis-)similarity scores from the features and sorting and ranking of items based on this score. This is commonly referred to as the *vector space model* of multimedia retrieval and similiarty search. Over the years, many different combinations of features and ranking models have been proposed to facilitate content-based retrieval of different media types, such as, images, audio or video as have been different types of query formulation, such as *Query-by-Sketch*, *Query-by-Humming* or *Query-by-Sculpting* [CWW+10; GLC+95; BGS+20].

Add sources: Survey for each modality

Furthermore, when looking at concrete system implementations that facilitate interactive multimedia retrieval for an end-user today, the lines between multimedia retrieval and multimedia analytics quickly start to blur. This is because, in addition to the extraction of appropriate features and the conception of effective ranking algorithms, multimedia retrieval systems today also concern themselves with aspects such as query (re-)formulation and refinement, results presentation and efficient exploration [LKM+19]. In addition, multimedia retrieval systems do not simply operate on features representations anymore but combine *similarity search* on features and *Boolean retrieval* on annotations and metadata [RGL+20]. Therefore, one could argue that multimedia retrieval systems perform a very specific type of multimedia analytics task, which is that of finding unknown items that satisfy a specific information need. This makes all the arguments made about data processing and data management requirements for multimedia analytics applicable to multimedia retrieval as well.

1.2 Research Gap and Objective

It has been pointed out by Jonson et al. [JWZ+16] (p. 296) that "Multimedia analytics state of the art [...] has up to now [...] not explicitly considered the issue of data management, despite aiming for large-scale analytics.". Despite recent advances and the development of concrete architecture models for multimedia database management [GS16; Gia18] and multimedia retrieval systems that refactor data management into distinct components [Ros18], that statement, to some extent, still holds true today. While [Gia18] makes important contributions towards a unified data-, query- and execution model required for effective search and exploration in multimedia collections, scalability aspects and the need for near real-time query performance, especially in the face of dynamic data, are not systematically considered. On the contrary, the proposed models – despite being seminal for data management in certain multimedia retrieval applications – postulate assumptions, that have considerable impact on the practical applicability of data management systems implementing them.

The starting point for the research described in this thesis is therefore the current state-of-the-art for data management in multimedia retrieval and analytics as briefly touched upon in the previous sections. Starting from and inspired by the models and solutions proposed in [GS16; Gia18] and motivated by the "Ten Research Questions for Scalable Multimedia Analytics" [JWZ+16], this thesis challenges three basic assumptions currently employed and operated upon in multimedia data management and explores the ramifications of doing so, with the higher level goal of bridging certain gaps between research conducted in multimedia retrieval, analysis and analytics on the one hand, and classical data management and databases on the other. These assumptions are namely:

Assumption 1: Staticity of data collections Most multimedia retrieval systems today make a distinction between an *offline* phase during which media items are analysed, features are generated and derived data is ingested into a data management system, and an *online* phase, during which queries of the data management system take place. Usually, no changes to the data collection are being made during the online phase. This model is proposed by both [Gia18] and [Ros18] and to the best of our knowledge, most existing multimedia retrieval and analytics systems implement this either explicitly or implicitly. This simplification allows for time consuming processes related to feature extraction and indexing to take place separated from any concurrent query activities and eases requirements on transaction isolation.

Assumption 2: Nearest neighbor search The vector space model used in multimedia retrieval relies on a notion of similarity search that is usually expressed as finding the k nearest neighboring feature vectors $\vec{v}_{i\in[1,k]} \in \mathrm{C}$ to a query vector $\vec{q} \in \mathbb{R}^d$ in a collection $\mathrm{C} \subset \mathbb{R}^d$ given a certain distance function. Very often, metrics such as the Euclidean or the Manhattan distance are employed for this comparison. While this model is very concise, computationally efficient and rather simple, it merely allows for the ranking of potential results and, given that the underlying model and the query is precise enough, finding the relevant or desired item(s).

Assumption 3: User defines execution Database management systems usually evaluate and select the execution plan for an incoming query during a step that is refered to as *query planning*. The underlying assumption here is that the database system has all the information required to determine the most effective execution path in terms of cost parameters such as required I/O, CPU and memory usage. In multimedia retrieval, this is not the case since, for example, index selection relies on a lot of different aspects that, to some extent, can be parametrized by the client issuing a query or that may be subject to change. Therefore, the index used for executing a query is often selected explicitly by the user issuing the query.

It is worth noting, that Assumption 1 and 3 both go against well-established design principles usually found it modern database systems [Pet19]. While it may be convenient from a perspective of system design, to assume a data collection to be static, such a mode of operation is utterly limiting when considering data that is subject to change, as is the case, for example, when doing analytics or when having an application with CRUD support. A similar argument can be made for manual index selection. Such an assumption may be simplifying the process of query planning but assumes, that a user is always a technical expert. Furthermore, it limits the amount of optimization that can be applied by the data management system especially in the face of non-static data collections, where indexes are changing, or changing query workloads.

As for Assumption 2, one can state that the described model is only able to accommodate the search-end of the *exploration-search axis*, assuming that features are, in fact, real valued vectors. It quickly becomes unusable for tasks such as browsing, structuring and summarization, delegating the required data processing to upper-tier system components. Refering to [JWZ+16], it would however be desirable to offer such primitives at the level of the data management system.

1.2.1 Research Questions

Challenging the aforementioned assumptions raises very specific questions that fundamentaly impact the design of a *multimedia data management system*. These questions are briefly summarized in Table 1.1.

Table 1.1 List of research questions (RQ) resulting from challenging assumptions(AS) one, two and three.

RQ	Question	Related to
1	Which commonly used, secondary index structures for NNS (e.g., VA [WSB98], LSH [IM98], PQ [JDS11] based indexes) can cope with changes to data and to what extent?	AS 1
2	Can we estimate and quantify deterioration of retrieval quality of index structures from RQ1 as changes are being made to the underlying data collections?	AS 1
3	How can we handle index structures from RQ1 for which to expect deterioration during query planning and execution?	AS 1
4	Can we devise a model that (temporarily) compensates deterioration of retrieval quality of index structures?	AS 1
5	How can user knowledge about the the retrieval task at hand be factored into query planning without forcing the user the make explicit choices about how a query should be executed?	AS 1 & 3
6	How would a cost model that factors in desired retrieval accuracy look like and can it be applied during query planning?	AS 3
7	Assuming the cost model in RQ6 exists, at what levels of the system can it be applied (globally, per query, context)?	AS 3
8	Is there a measurable impact (e.g., on query execution time vs. accuracy) of having such a cost model?	AS 3
9	Can we generalize the model for similarity search (i.e., the vector space model) and what is the consequence of doing so?	AS 2
10	Do the existing applications and use-cases justify a generalization?	AS 2

RQ1 to RQ5 address the issue of index structures for NNS, which are mostly unable to cope with data that is subject to change, since their correctness deteriorates as data is modified. The focus of these questions are whether deterioration can be quantified and how it can be handled by a system. We argue, that both is necessary for practical application in dynamic data management.

RQ6 to RQ8 explore the possibility of a cost model, that takes accuracy of the produced results into account. Since most techniques for fast NNS rely on approximation, inaccuracy is an inherent factor for such operations. Assuming such a model exists, it can be used by a user or system administrator to make explicit choices between either accuracy or execution peformance. In addition, such a cost model can be put to use when deciding what indexes to use in face of deteriorated retrieval quality due to changing data.

And finally, RQ9 and RQ10 address the issue of a more generalized model for similarity search and the impact of such a model on all the different system components. Most importantly, however, they explore and justify the need for such a model, which is not self-evident, based on concrete use-cases and applications.

1.3 Contribution

In this thesis, we try to address the research gap identified and described in Section 1.2 and thereby try to bridge the disparity between the fields of databases and retrieval systems. The contribution of this thesis can be summarized as follows:

- We examine the impact of challenging the *data staticity assumption* on index structures commonly used for nearest neighbor search. Most importantly, we describe a model to *quantify* the effect of changing data for commonly used structures and to *expose* that information to the data management system.

- We describe an *adaptive index management* model by which a data management system can compensate errors introduced at an index level due to changes to the underlying data. The main design goal for that mechanism is, that it can be employed regardless of what type of index is used underneath.

- We propose a *cost-model that factors-in accuracy* of generated results in addition to common performance metrics, such as IO-, CPU-, or memory-usage and, based on that model, derive mechanisms for the user to express their preference for either accuacy or speed at different levels of the system.

- We postulate a more *generalized model for similarity search* and explore implications of such a model on aspects, such as, query planning.

- We introduce a working implementation that implements the aforementioned models, in the form of *Cottontail DB* [GRH+20].

- We present an evaluation of the impact of the model changes on real world datasets to provide a basis upon which their applicability can be assessed.

The described contributions are presented in four parts: The first part, to which this introduction belongs, introduces the problem and provides motivating use-cases and applications (Chapter 2) as well as an overview of relevant research done in the fields of multimedia retrieval, multimedia analytics and data management (Chapter 3).

The second part, gives a brief summary of the theoretical foundation in multimedia analysis & retrieval (Chapter 4) and databases (Chapter 5) required to understand the remainder of this thesis. The function of these chapters is that of a refresher for readers not familiar with either domain.

The third part introduces the theoretical models that make up the aforementioned contributions (Chapter 6) and introduces our reference implementation Cottontail DB (Chapter 7).

The fourth and final part presents the evaluation using Cottontail DB as an implementation (Chapter 8) and discusses the conlusions and potential future work (Chapter 9)

2

Applications and Use Cases

2.1 Use case 1: Multimedia Retrieval System

vitrivr, vitrivr VR with focus on search and exploration and data mangement & query implications (e.g., for SOMs, staged querying etc.). It remains to be seen how changes to data can be motivated here.

2.2 Use case 2: Analysis of Social Media Streams

Online analysis in Pythia, Delphi. Mainly as a motivating use case for why data may be subject to change.

Demo paper!

2.3 Use case 3: Magnetic Resonance Fingerprinting (MRF)

MRF as a concrete example why the classical NNS is too limited for certain use cases and an extension should be considered.

Paper!

3

Related Work

Part II

Foundations

4

On Multimedia Analysis and Retrieval

4.1 Multimedia Data and Multimedia Collections

Formalisation of what multimedia data is and what forms it can take (video, audio, images, text + metadata etc.). This formal model has the potential of being an original contribution, since we will make very explicit assumptions about what aspects of a multimedia item there are and which ones are mutable or immutable (e.g, content vs. annotations, metadata, features etc.)

4.2 Multimedia Retrieval

4.2.1 Similarity and the Vector Space Model

4.2.2 Approximate Nearest Neighbor Search

Describe techniques for approximate nearest neighbor search (ANN). Focus on a more conceptual overview of the types of algorithms rather than just enumerating concrete examples; this can be used as a build-up for discussing properties of different index structures later.

4.2.3 Beyond Similarity Search

Retrieval and analytics techniques that go beyond simple similarity search (e.g. SOM, summarization, clustering)

4.3 Online Multimedia Analysis

Introducing an online analysis pipeline (e.g., Pythia / Delphi).

4.4 Multimedia Analytics

Describe how the combination of analysis

4.4.1 Beyond Similarity Search

5

On The Design of a Database Management System

Digression into design considerations of a database management system (storage, locking, query planning, execution model etc.)

5.1 The Relational Data Model

5.2 Storage, Indexes and Caching

5.3 Operational Primitives

5.4 Query Parsing, Planning and Execution

5.5 Architectual Considerations

Part III

Dynamic Multimedia Data Management

6

Modelling a Database for Dynamic Multimedia Data

6.1 Generalized Similarity Operations

As has been argued in Chapter 4, there are two important assumptions for similiarity based operations, such as similarity search in multimedia retrieval. These assumptions can be summarized as follows:

- For every object o_i in a collection C, there exists a feature transformation $\phi\colon C \rightarrow \mathcal{F}$, that maps the object $o_i \in C$ to a feature space $\mathcal{F}$.
- The feature space $\mathcal{F}$ and a to be defined distance function $\delta\colon \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$, constitute a metric space $(\mathcal{F}, \delta)$, thus satisfying the identity of indiscernibles, symmetry and subadditivity condition.

For all similarity based operations, the output of δ – i.e., the calculated distance d – acts as a proxy for (dis-)similiarty between two objects o_i, $o_j \in C$ given the feature transformation ϕ. Hence, the closer two objects o_i, o_j appear under the transformation, the more (dis-)similar they are. It must be pointed out for the sake of completeness, that whether similarity is directly or inversely proportional to the distance is a matter of definition and depends on the concrete application. Also, in practice, multiple feature transformations ϕ_n may exist for a given media collection, leading to different feature spaces $\mathcal{F}_n$ for a collection C that must be considered jointly. Both these aspects are usually addressed by additional correspondence and scoring functions. Since both these aspects are not relevant for the discussion at hand, they will for now be ignored.

Using the relationship between (diss-)similarity and distance, it has been shown by Giangreco et al., that for a database to be able to support similarity-based search given the relational model for databases [Cod90], one can extend the data domain $\mathcal{D}$ by $\mathbb{R}^{dim}, dim \in \mathbb{N}$ and postulate the existence of a relational similarity operator $\tau_{\delta(\cdot,\cdot),a,q}(R)$ that *"performs a similarity query under a distance $\delta(\cdot,\cdot)$ applied on an attribute a of relation R and compared to a query vector q."* ([Gia18], p. 138). Such an operation introduces an implicit attribute in the underlying relation R, which in turn induces an ascending ordering of the tuples. Using this operation, the authors then go on to define two concrete implementations, namely $\tau^{kNN}_{\delta(\cdot,\cdot),a,q}(R)$, and $\tau^{\epsilon NN}_{\delta(\cdot,\cdot),a,q}(R)$, which limit the number of retrieved results by their cardinality k or a maximum cut-off distance ϵ respectively.

6.1.1 Revisiting Distance Computation

Considerding the definition provided in Chapter 4, we identify the following (implicit) constrained of the postulated model:

1. The codomain (i.e., the output) of the distance function δ is assumed to be $\mathbb{R}$, hence, the distance value we generate is a real number.

2. The domain (i.e., the input) of the distance function δ is assumed to be $\mathbb{R}^{dim} \times \mathbb{R}^{dim}$, hence, we restrict ourselves to real-valued vectors and the distance function is assumed to be a binary function.

Upon further examination, one can see that there is very good reason to assume the codomain of δ to be in $\mathbb{R}$. On the one hand, it is obviously convenient both for the underlying mathematics as well as from a programming prespective. More importantly, however, real numbers – unlike, for example, complex numbers or vectors – come with a natural, total ordering, which is required for the sorting that is part of the relational similarity search operation. If, however, we turn to the use cases presented in Section 2.1 and Section 2.3, one can see that both the domain and the arity of a regular distance function are often too limited, as is shown in examples Example 6.1 and Example 6.2.

Example 6.1 Maximum Inner Product Search (MIPS) for MRF

In MRF, we try to find the signal vector $f_i \in \mathcal{F}$ from a dictionary $\mathcal{F} \subset \mathbb{C}^{dim}$ so that it maximizes the inner product to a query vector $q \in \mathbb{C}^{dim}$. In this case, the distance function δ has the form $\delta \colon \mathbb{C}^{dim} \times \mathbb{C}^{dim} \to \mathbb{R}$ with $dim \in \mathbb{N}$.

Example 6.2 Distance Between a Vector and a Hyperplane

In the example introduced in Section 2.1, want to find positive/negative examples for features in $\mathcal{F} \subset \mathbb{R}^{dim}$ given a linear classifier, e.g., provided by a SVM. Mathematically, such a classifier can be defined by a hyperplane $\mathbf{w}^T\mathbf{x} - b = 0$ with $\mathbf{w}, \mathbf{x} \in \mathbb{R}^{dim}$ and $b \in \mathbb{R}$. The distance function can then be defined as $d\colon \mathbb{R}^{dim} \times \mathbb{R}^{dim} \times \mathbb{R} \to \mathbb{R}$ as both $\mathbf{w}$ and b become parameters of the function, in addition to the attributes $f_i \in \mathcal{F}$ for which a distance is obtained. Hence, the distance function is no longer a binary but a ternay function with attributes $\mathbf{f_i}$, $\mathbf{w}$ and b.

In order to address these limitations, we propose the extension of a the distance function to the notion of a *similarity proxy function (SPF)* following Definition 6.2. As the name implies, such a function acts as a proxy for (dis-)similarity in similarity based operations.

Definition 6.1 Similarity Proxy Function (SPF)

A *similarity proxy function (SPF)* $\delta\colon \mathcal{F} \times \mathcal{F} \times \mathcal{D}_1 ... \times \mathcal{D}_n \to \mathbb{R}$ is an n-ary but at least binary function that outputs a distance $d \in \mathbb{R}$ between an attribute $f_a \in \mathcal{F} \subset \mathcal{D}_i$ and a query $f_q \in \mathcal{D}_i$ using a well defined number of *support arguments* from the data domains $\mathcal{D}_j$ with $i, j \in \mathbb{N}$.

In simple terms, a SPF can be seen as a distance function that is being parametrized by its support arguments. A very simple and widely used example would be the Manhattan (L1) and the Euclidean (L2) distance, which are both parametrized versions of the more general Minkowski distance $\delta_M\colon \mathbb{R}^{dim} \times \mathbb{R}^{dim} \times \mathbb{N} \to \mathbb{R}$ with

$$\delta_{L1}(\mathbf{q}, \mathbf{f}) = \sum_{i=1}^{n} \mid q_i - f_i \mid = \left(\sum_{i=1}^{n} \mid q_i - f_i \mid^p \right)^{\frac{1}{p}}, p = 1 \tag{6.1}$$

$$\delta_{L2}(\mathbf{q}, \mathbf{f}) = \sqrt{\sum_{i=1}^{n} \mid q_i - f_i \mid^2} = \left(\sum_{i=1}^{n} \mid q_i - f_i \mid^p \right)^{\frac{1}{p}}, p = 2 \tag{6.2}$$

Definition 6.2 Similarity Proxy Function (MSPF)

A SPF $\delta_{\mathcal{F}}: \mathcal{F} \times \mathcal{F} \times \mathcal{D}_1 ... \times \mathcal{D}_n \rightarrow \mathbb{R}$ that fulfills the identity of indiscernibles, symmetry and subadditivity conditions with respect to $f_a \in \mathcal{F} \subset \mathcal{D}_i$ and $f_q \in \mathcal{D}_i$ is called a *metric similarity proxy function (MSPF)*. It induces a metric on the vector space $\mathcal{D}_j$.

6.1.1.1 Parametrized dimensionality

As an aside, we must address the role of dimensionality in the case of vector spaces such as $\mathbb{R}^{dim}$ or $\mathbb{C}^{dim}$. One could argue, that the dimensionality of such a vector space can also be seens as a parameter of the SPF. Nevertheless, we consider the dimensionality to be a structural property of the underlying data domain as, for example, the data type. This means, that dimensionality as well as the type are well-defined and most importantly constant properties for a given relation.

6.1.2 Extending the Relational Model

Assuming the existence of an SPF as described in Definition 6.2, one can start to integrate this into the relational algebra model. While postulating a new, relational operation $\tau^{kNN}_{\delta(\cdot,\cdot),a,q}(R)$ or $\tau^{\epsilon NN}_{\delta(\cdot,\cdot),a,q}(R)$ as proposed by [Gia18] has a certain elegance to it, it also comes with limitations that become apparent once we disect the structure of τ. In its postulated form, τ addresses several functions at once:

1. It specifies the distance function and its parameters.
2. It generates an implicit distance attribute on the underlying relation R.
3. It implies an implicit, ascending ordering on the tuples.
4. It enforces a limit on the result set.

While being very specific and thus straightforward to implement and optimize, the amalgamation of all this functionality into a single operation is very specifically tailored to the use-case of similarity search and only of limited value when considering more general similarity-based query operations. If we, for example, want to obtain the k farthest neighbours rather than the k nearest neighbours, as necessary when doing MIPS or obtaining negative examples, we would have to either change the distance function or extend the definition of τ.

Another important issue with the definition of τ in its current form is givn in Examples 6.3 and 6.4.

Example 6.3 Shortcomings of $\tau^{kNN}_{\delta(\cdot,\cdot),a,q}$

Given a relation R with $SCH(R) = \{a_1, a_2\}$, we consider the two relational expressions $\pi_{a1,d}(\tau^{kNN}_{\delta(\cdot,\cdot),a_2,q}(\sigma_{a_1=true}(R)))$ and $\pi_{a1,d}(\sigma_{a_1=true}(\tau^{kNN}_{\delta(\cdot,\cdot),a_2,q}(R)))$.
The keen observer will notice that for small numbers of k with respect to the number of tuples that match the selection σ, the two expressions will not return the same resultset due to the order of execution. The first expression will always return a result set of size k given that at least k tuples in R satisfy σ. The second expression, however, is likely to produce a resultset smaller than k since the selection takes place on a top k ranked subset of R.

Example 6.4 Shortcomings of $\tau^{\epsilon NN}_{\delta(\cdot,\cdot),a,q}$

Given a relation R with $SCH(R) = \{a_1, a_2\}$, we consider the two relational expressions $\pi_{a1,d}(\tau^{\epsilon NN}_{\delta(\cdot,\cdot),a_2,q}(\sigma_{a_1=true}(R)))$ and $\pi_{a1,d}(\sigma_{a_1=true}(\tau^{\epsilon NN}_{\delta(\cdot,\cdot),a_2,q}(R)))$, which select all entries that pass predicate σ and whose distance $d \geq \epsilon$.
Again, the keen observer will notice that the two result sets are going to be equal in this case. This can be attributed to the fact that the limiting of the resultset is formally introduced by another selection, i.e., the expression can be restated as $\pi_{a1,d}(\sigma_{d\geq\epsilon}(\tau_{\delta(\cdot,\cdot),a_2,q}(\sigma_{a_1=true}(R))))$. It is perfectly valid to move the inner selection to the outside due to the commutativity of the selection operation. The only limitation is, that $\sigma \geq \epsilon$ requires prior execution of $tau_{\delta(\cdot,\cdot),a_2,q}$.

With these example, we have demonstrated the limitations of τ and the fact that the two proposed implementations for kNN and ϵNN actually behave very differently despite their common core. We therefore propose to decompose τ into distinct extensions to the relational model, with a clear focus on separation of concerns.

6.1.2.1 Extended Project and SPF

First, we consider the SPF to be part of an *extended projection* π, which has been defined as part of the extended relational algebra as described in Section 5.1. In addition to projection on attributes, the extended projection allows for the projection on general algebraic expressions involving attributes, constants and function invocations. Given the extended projection, the invocation of a SPF can be expressed as follows.

Definition 6.3 Similarity Proxy Function (SPF) in Extended Projection

Let $\delta \colon \mathcal{F} \times \mathcal{F} \times \mathcal{D}_1 ... \times \mathcal{D}_{n-2} \rightarrow \mathbb{R}$ be an SPF and R be a relation with $SCH(R) = \{a_1, a_2, ...a_n\}$. The *extended projection* $\pi_{\delta(a_1,a_2,...,a_n)}$(R) describes the execution of the n-ary SPF δ using attributes $a_1, a_2, ...a_n$ from relation R as parameters. Note that $\pi_{\delta(a_1,a_2,...,a_n)}$ introduces a new, calculated distance attribute $a_d \in \mathbb{R}$ on each tuple $t_i \in R$, i.e., $SCH(\pi_{\delta(a_1,a_2,...,a_n),a_1,a_2,...,a_n}) = SCH(R) \cup \{a_d\}$.

Obviously, the combination of multiple SPF in a single, extended projection or the combination of simple attribute projection with SPFs are also allowed. Hence, the following expressions are valid examples of the extended projection on relation R with $SCH(R) = \{a_1, a_2, a_3, a_4\}$: $\pi_{\delta_1(a_1,a_2),\delta_2(a_2,a_3,a_1)}(R)$ or $\pi_{\delta(a_1,a_2),a_3,a_4}(R)$ or $\pi_{a_1,a_3,a_4}(R)$.

- NNS: Scan -> (Predicate) -> Distance Function -> Sort -> Limit -> (Predicate); no need for dedicated language feature aside from distance function
- Distance function is a binary function $D(q, v) \longrightarrow d$, q and v can be elements of $\mathbb{R}^d, \mathbb{C}^d$ or some other object (e.g. matrices)
- Different types of distance functions, depending on parameters they accept; e.g. distance between point & point or point & plane etc.
- Systems perspective 1: How enable planner to reason about distance function execution? Possible optimizations?
- Systems perspective 2: Concrete applications in multimedia retrieval and analytics?

6.2 Cost Model for Retrieval Accuracy

Describe cost model for execution plans with following properties:

- Cost as a function of atomic costs: $f(a_{cpu}, a_{io}, a_{memory}, a_{accuracy}) \longrightarrow C$
- Means to estimate results accuracy and associated considerations from execution path (e.g., when using index) based on properties of the index

- Means to specify importance of accurate results (e.g., global, per-query, context-based i.e. when doing 1NN search) in comparison to other factors
- Systems perspective 1: How can such a cost model be applied during query planning and optimization?

6.3 Adaptive Index Management

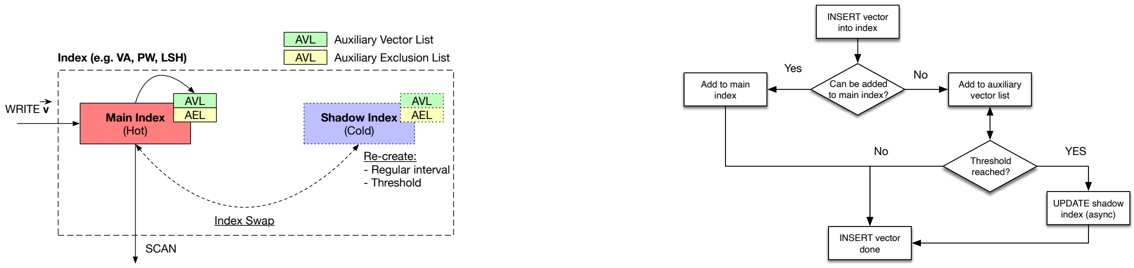

Figure 6.1 Adaptive index structures overview.

Describe model for index management in the face of changing data (adaptive index management):

- Reason about properties of secondary indexes for NNS (e.q., PQ, VA, LSH) with regards to data change
- Derivation of error bounds possible (e.g., usable for planning)?! Use in query planning?
- Systems perspective 1: How to cope with "dirty" indexes? Proposal: hot vs. cold index, auxilary data structure, offline optimization, see Figure 6.1
- Systems perspective 2: On-demand index based on query workload?

6.4 Architecture Model

Putting everything together into a unified systems model (base on previous work + aforementioned aspects).

7

Cottontail DB

Implementation chapter for Cottontail DB

Part IV

Discussion

8

Evaluation

8.1 Adaptive Index Management

Brute force vs. plain index vs. index with auxilary data structure

8.2 Cost Model

Benchmark effect of cost model in different settings (e.g. based on use cases from chapter 2)

9

Conclusion & Future Work

Appendix

Bibliography

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[BdB+07] Henk M Blanken, Arjen P de Vries, Henk Ernst Blok, and Ling Feng. *Multimedia Retrieval*. Springer, 2007.

[BGS+20] Samuel Börlin, Ralph Gasser, Florian Spiess, and Heiko Schuldt. 3D Model Retrieval Using Constructive Solid Geometry in Virtual Reality. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 373–374. IEEE, 2020.

[CWW+10] Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. MindFinder: interactive sketch-based image search on millions of images. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1605–1608, New York, NY, USA. Association for Computing Machinery, 2010. ISBN: 978-1-60558-933-6. DOI: `10.1145/1873951.1874299`.

[CTW+10] Nancy A. Chinchor, James J. Thomas, Pak Chung Wong, Michael G. Christel, and William Ribarsky. Multimedia Analysis + Visual Analytics = Multimedia Analytics. *IEEE Computer Graphics and Applications*, 30(5):52–60, 2010. DOI: `10.1109/MCG.2010.92`.

[Cod90] Edgar F. Codd. *The Relational Model for Database Management: Version 2*. Addison-Wesley Longman Publishing Co., Inc., 1990.

[GRH+20] Ralph Gasser, Luca Rossetto, Silvan Heller, and Heiko Schuldt. Cottontail DB: An Open Source Database System for Multimedia Retrieval and Analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4465–4468, 2020.

[GLC+95] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C Smith. Query by Humming: Musical Information Retrieval in an Audio Database. In *Proceedings of the Third ACM International Conference on Multimedia*, pages 231–236, 1995.

[Gia18] Ivan Giangreco. *Database Support for Large-Scale Multimedia Retrieval*. PhD thesis, University of Basel, Switzerland, August 2018.

[GS16] Ivan Giangreco and Heiko Schuldt. ADAMpro: Database support for big multimedia retrieval. *Datenbank-Spektrum*, 16(1):17–26, 2016. DOI: `10.1007/s13222-015-0209-y`.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[JDS11] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product Quantization for Nearest Neighbor Search. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):117–128, January 2011. DOI: `10.1109/TPAMI.2010.57`.

[JWZ$^+$16] Björn Þór Jónsson, Marcel Worring, Jan Zahálka, Stevan Rudinac, and Laurent Amsaleg. Ten Research Questions for Scalable Multimedia Analytics. In *International Conference on Multimedia Modeling*, pages 290–302. Springer, 2016.

[KKE$^+$10] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. Mastering the information age: Solving problems with visual analytics, 2010.

[LKM$^+$19] Jakub Lokoč, Gregor Kovalčík, Bernd Münzer, Klaus Schöffmann, Werner Bailer, Ralph Gasser, Stefanos Vrochidis, Phuong Anh Nguyen, Sitapa Rujikietgumjorn, and Kai Uwe Barthel. Interactive Search or Sequential Browsing? A Detailed Analysis of the Video Browser Showdown 2018. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(1):1–18, 2019.

[Low99] David G Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[Pet19] Alex Petrov. *Database Internals*. en. O'Reilly Media, Inc., 2019. ISBN: 978-1-4920-4034-7.

[Ros18] Luca Rossetto. *Multi-Modal Video Retrieval*. PhD thesis, University of Basel, Switzerland, September 2018.

[RGL+20] Luca Rossetto, Ralph Gasser, Jakub Lokoc, Werner Bailer, Klaus Schoeffmann, Bernd Muenzer, Tomas Soucek, Phuong Anh Nguyen, Paolo Bolettieri, Andreas Leibetseder, et al. Interactive Video Retrieval in the Age of Deep Learning - Detailed Evaluation of VBS 2019. *IEEE Transactions on Multimedia*, 2020.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *VLDB*, volume 98, pages 194–205, New York City, NY, USA. Morgan Kaufmann, 1998.

[ZW14] Jan Zahálka and Marcel Worring. Towards interactive, intelligent, and integrated multimedia analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 3–12, 2014. DOI: 10.1109/VAST.2014.7042476.

Curriculum Vitae

Name	Ralph Marc Philipp Gasser Brunnenweg 10, 4632 Trimbach
Date of Birth	28.03.1987
Birthplace	Riehen BS, Switzerland
Citizenship	Switzerland

Education

since Jan. 2000 Ph. D. in Computer Science under the supervision of Prof. Dr. Heiko Schuldt, Databases and Information Systems research group, University of Basel, Switzerland

Sept. 1997 – Aug. 1999 M. Sc. in Computer Science, University of Basel, Switzerland

Sept. 1994 – Aug. 1997 B. Sc. in Computer Science, University of Basel, Switzerland

Employment

since Jan. 2000 Research and teaching assistant, Databases and Information Systems research group, University of Basel, Switzerland

Publications

1937

- **turing:1937**.

Hand-in in thesis and separately

Declaration on Scientific Integrity

includes Declaration on Plagiarism and Fraud

Author

Ralph Marc Philipp Gasser

Matriculation Number

2007-050-131

Title of Work

Data Management for Dynamic Multimedia Analytics and Retrieval

PhD Subject

Computer Sciences

Declaration

I hereby declare that this doctoral dissertation *"Data Management for Dynamic Multimedia Analytics and Retrieval"* has been completed only with the assistance mentioned herein and that it has not been submitted for award to any other university nor to any other faculty at the University of Basel.

Basel, DD.MM.YYYY

Signature

Hand-in separately