Temporal Multimodal Video and Lifelog Retrieval

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Silvan Heller

Basel, 2023

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Heiko Schuldt
Erstbetreuer

Prof. Dr. Malte Helmert
Zweitbetreuer

Prof. Dr. Fabio Crestani
Externer Experte

Basel, 28.03.2023

Prof. Dr. Marcel Mayor
Dekan

For all those who see potential in others
and give them the opportunity to grow

Det er ganske sandt, hvad Philosophien siger, at Livet maa forstaaes baglænds. Men derover glemmer man den anden Sætning, at det maa leves forlænds.

It is really true what philosophy tells us, that life must be understood backwards. But with this, one forgets the second proposition, that it must be lived forwards.

— Søren Kierkegaard
Journalen JJ:167 (1843)

Abstract

The past decades have seen exponential growth of both consumption and production of data, with multimedia such as images and videos contributing significantly to said growth. The widespread proliferation of smartphones has provided everyday users with the ability to consume and produce such content easily. As the complexity and diversity of multimedia data has grown, so has the need for more complex retrieval models which address the information needs of users. Finding relevant multimedia content is central in many scenarios, from internet search engines and medical retrieval to querying one's personal multimedia archive, also called lifelog. Traditional retrieval models have often focused on queries targeting small units of retrieval, yet users usually remember temporal context and expect results to include this. However, there is little research into enabling these information needs in interactive multimedia retrieval.

In this thesis, we aim to close this research gap by making several contributions to multimedia retrieval with a focus on two scenarios, namely video and lifelog retrieval. We provide a retrieval model for complex information needs with temporal components, including a data model for multimedia retrieval, a query model for complex information needs, and a modular and adaptable query execution model which includes novel algorithms for result fusion. The concepts and models are implemented in vitrivr, an open-source multimodal multimedia retrieval system, which covers all aspects from extraction to query formulation and browsing. vitrivr has proven its usefulness in evaluation campaigns and is now used in two large-scale interdisciplinary research projects. We show the feasibility and effectiveness of our contributions in two ways: firstly, through results from user-centric evaluations which pit different user-system combinations against one another. Secondly, we perform a system-centric evaluation by creating a new dataset for temporal information needs in video and lifelog retrieval with which we quantitatively evaluate our models.

The results show significant benefits for systems that enable users to specify more complex information needs with temporal components. Participation in interactive retrieval evaluation campaigns over multiple years provides insight into possible future developments and challenges of such campaigns.

Acknowledgements

First, I want to thank Prof. Dr. Heiko Schuldt. Starting with the programming project and reaffirmed in subsequent lectures and projects, it was clear to me that the Databases and Information Systems Group was one where there is a focus on excellence and commitment in teaching, supervision, research, and building systems. Thanks for your continued support and guidance during these years.

I would also like to thank Prof. Dr. Fabio Crestani for his willingness to review this thesis as external expert, my second supervisor Prof. Dr. Malte Helmert, especially for the collaboration throughout the years in the various committees, and Prof. Dr. Isabel Wagner for chairing my defense.

Over the past years, I had the great pleasure of working with current and former members of the DBIS group. My first thanks here go to Ivan Giangreco and Luca Rossetto, for — already during my studies — always having an open door and countless hours of discussions. Ralph Gasser, Mahnaz Parian-Scherb, Loris Sauter, and Florian Spiess were great collaborators on various projects, some of which have made their way into this dissertation. Lukas Probst, Alexander Stiemer, and Marco Vogt were always available for a question on databases, a discussion on teaching, or anything else the PhD student life required. In general, I want to thank all current and former members of the DBIS Group, namely Rahel Arnold, Sein Coray, David Lengweiler, Ashery Mbilinyi, Dina Sayed, Philipp Seidenschwarz, and Shaban Shabani for the shared time throughout the years, in discussions, meetings, teaching, hikes, retreats, and lunches.

I have worked with many talented students during my dissertation project and would like to thank Simon Peterhans, Manuel Rickli, Paul Höft, Cristina Illi, Sein Coray, Kalthoum Nemmour, Maurizio Pasquinelli, Timo Castelberg, Viktor Gsteiger, Sanja Popovic, Vera Benz, and Nico Aebischer for choosing me as their supervisor for theses and projects, and Cristina Illi, Kalthoum Nemmour, Simon Peterhans, Sebastian Philipp, Jan Schönholz, Rahel Arnold, Nikodem Kernbach, Tim Bachmann, Esther Mugdan, Luka Obser, Claire Walzer, Vera Benz, Flurina Fischer, Colin Saladin, Luc Heitz, Raphael Waltenspül, Maria Desteffani, and Sascha Maibach who have worked with me as TAs in lectures. You all have bright futures ahead of you. A special mention goes to all the contributors of the vitrivr system, especially the students during Google Summer of Code.

I would also like to thank all the people I have collaborated with over the

years on publications, in particular Jakub Lokoč for interesting discussions on all things VBS and video retrieval, Björn Þór Jónsson for the in-depth input and feedback during the writing of my first journal publication, Lucia Vadicamo for discussing nuances and pitfalls of statistical analysis and data visualization, Werner Bailer for designing fun VBS tasks, Klaus Schoeffmann for his efforts in organizing VBS, and Cathal Gurrin for his efforts in organizing LSC.

Having done my studies and my PhD at the Department of Mathematics and Computer Science in Basel, I benefited from the work of all the people who make research and teaching possible in visible and invisible ways, be they administration, facility management, IT, or management. Thank you!

On a more personal note, I want to thank friends and family who have supported me throughout these years, listened to success stories and complaints, and made my life immeasurably better. You know who you are.

Finally, I want to thank the love of my life, Viviane Alexandra Blatter-Heller, for everything.

In addition to feedback given during progress reports, presentations, at conferences and workshops, and in other informal settings, specific parts of this thesis have benefited from reviewing work by Rahel Arnold (Section 6.1), Viviane Alexandra Blatter-Heller (Abstract and Chapter 1), Ralph Gasser (Chapters 1 to 4), Ivan Giangreco (Chapters 1, 2 and 8), Heiko Schuldt (Chapters 1 and 4), Florian Spiess (Chapters 3 to 5), and Marco Vogt (Chapters 1, 2 and 7 and Section 6.2).

Illustrations in this thesis are made with either Seaborn[1] (using JupyterLab[2], Pandas[3], and Numpy[4]) or Diagrams[5].

This work was partly supported by the Swiss National Science Foundation (project "Polypheny-DB: Cost- and Workload-aware Adaptive Data Management", contract no. 200021_172763) and by the Nachwuchsförderpreis of the Ferdinand Neeracher-Pfrunder Foundation, which is thankfully acknowledged.

[1] `https://seaborn.pydata.org`
[2] `https://jupyter.org`
[3] `https://pandas.pydata.org`
[4] `https://numpy.org`
[5] `https://www.diagrams.net`

Contents

List of Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
ASR	Automatic Speech Recognition
AVGSSA	Average Segment Scoring Algorithm
AVS	Ad-Hoc Video Search
CLI	Command-Line Interface
DBMS	Database Management System
DCF	Distance-Combining Function
DCG	Discounted Cumulative Gain
EDA	Exponential Decay Algorithm
GSOC	Google Summer Of Code
HCI	Human-Computer Interaction
IR	Information Retrieval
KIS	Known Item Search
LNA	Log Normal Decay Algorithm
LSC	Lifelog Search Challenge
MAP	Mean Average Precision
MAXSSA	Maximum Segment Scoring Algorithm
MR	Multimedia Retrieval
MRF	Magnetic Resonance Fingerprinting
NDA	Normal Distribution Algorithm

OCR	Optical Character Recognition
RF	Relevance Feedback
SCF	Similarity-Combining Function
SOM	Self-Organizing Map
SSM	Self-Sorting Map
T-KIS	Textual Known Item Search
TREC	Text Retrieval Conferences
UI	User Interface
V-KIS	Visual Known Item Search
VBS	Video Browser Showdown
VR	Virtual Reality

List of Symbols

Retrieval Foundations

$u \in \mathcal{U}$	user of a retrieval system
$\mathcal{U}$	set of all users
$in \in \mathcal{IN}$	information need of a user
$\mathcal{IN}$	set of all information needs of users
$q \in \mathcal{Q}$	query of a user
$\mathcal{Q}$	set of all user queries
$\mathrm{QF} : \mathcal{IN} \times \mathcal{U} \to \mathcal{Q}$	query formulation function
$o \in \mathcal{O}$	multimedia object
$\mathcal{O}$	set of all multimedia objects
$f \in \mathcal{F}$	feature; compact representation of something
$\mathcal{F}$	set of all features
$\mathrm{F}_o : \mathcal{O} \to \mathcal{F}$	object transformation function
$\mathrm{F}_q : \mathcal{Q} \to \mathcal{F}$	user query transformation function
$\mathrm{REL} : \mathcal{F} \times \mathcal{F} \to [0,1]$	relevance function
$\mathrm{DS} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$	dissimilarity function comparing two elements
$\delta \in \mathbb{R}_{\geq 0}$	distance between two elements
$\overline{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$	list of distances
$\mathrm{C} : \mathbb{R}_{\geq 0} \to [0,1]$	correspondence function
$\mathrm{C}_{lin}\,(\delta, \mathtt{max}) \mapsto 1 - \frac{\delta}{\mathtt{max}}$	linear correspondence function
$\mathrm{C}_{hyp}\,(\delta, \mathtt{div}) \mapsto \frac{1}{1+\frac{\delta}{\mathtt{div}}}$	hyperbolic correspondence function
$\hat{\delta} : (\mathbb{R}_{\geq 0})^n \to \mathbb{R}_{\geq 0}$	distance-combining function
$w \in [0,1]$	weight for a query component
$\overline{w} = (w_1, w_2, \ldots, w_n)$	weights for components of a query

Data Model

`oid`	object identifier; uniquely identifies an object
$s \in \mathcal{S}$	segment
$\mathcal{S}$	set of all segments across all multimedia objects
`sid`	segment identifier; uniquely identifies a segment
$\tau \in [0,1]$	relevance score

$\overline{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$	list of relevance scores
$\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$	scored segment
$\hat{\mathcal{S}}$	set of all scored segments
$\text{SEG} : O \rightarrow 2^{\mathcal{S}}$	segmentation function
f	retrieval feature
$\overline{f} = (f_1, f_2, \ldots, f_n)$	list of retrieval features
$f_s : \mathcal{S} \rightarrow \mathcal{F}$	segment transformation function
$f_{\texttt{qt}} : QT \rightarrow \mathcal{F}$	query transformation function
$f_r : QT \rightarrow 2^{\hat{\mathcal{S}}}$	feature retrieval function

Query Model

$\texttt{qt} \in QT := \left\langle \texttt{data}, \overline{f} \right\rangle$	query term; atomic query element
QT	set of all query terms
$\overline{\texttt{qt}} = (\texttt{qt}_1, \texttt{qt}_2, \ldots, \texttt{qt}_n)$	list of query terms
$\hat{\varrho} : ([0,1])^n \rightarrow [0,1]$	Similarity-Combining Function (SCF)
$\texttt{csq} := \left\langle \overline{\texttt{qt}}, \hat{\varrho} \right\rangle$	complex similarity query
$\overline{\texttt{csq}} = (\texttt{csq}_1, \texttt{csq}_2, \ldots, \texttt{csq}_n)$	list of complex similarity queries
$\texttt{tsq} := \left\langle \overline{\texttt{csq}}, \overline{\phi}, \omega, \overline{\texttt{qt}_{\texttt{tsq}}}, \hat{\varrho}_{\texttt{tsq}} \right\rangle$	temporal similarity query
$\phi \in \Phi$	desired distance between two subqueries
Φ	set of all user-specified distances
$\overline{\phi} = (\phi_1, \phi_2, \ldots, \phi_n)$	list of distances between subqueries
$\omega \in \Omega$	desired maximum length of a result
Ω	set of all possible ω
$\overline{\texttt{qt}_{\texttt{tsq}}} = \left(\texttt{qt}_{\texttt{tsq}_1}, \texttt{qt}_{\texttt{tsq}_2}, \ldots, \texttt{qt}_{\texttt{tsq}_q}\right)$	query-level constraints of a temporal query
$\hat{\varrho}_{\texttt{tsq}}$	SCF describing how query-level constraints are merged with subquery results

Result Model

$\overline{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$	list of scored sequences
$\tilde{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$	candidate sequence; list of scored sequences
$r = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$	result; scored segment list
$r_f = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$	result of a retrieval feature, scored list
$r_{\texttt{sqi}} = \left(r_{\texttt{qt}_1}, r_{\texttt{qt}_2}, \ldots, r_{\texttt{qt}_u}\right)$	intermediate results per query term
$r_{\texttt{csq}} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$	result of a complex similarity query, scored list
$\overline{r_{\texttt{csq}}} = \left(r_{\texttt{csq}_1}, r_{\texttt{csq}_2}, \ldots, r_{\texttt{csq}_n}\right)$	list of subquery results

$r_{\texttt{tsq}} = (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_x)$	result of a temporal similarity query, scored list

Fusion Functions

$\hat{\varrho}_{lc} : [0,1]^n \times [0,1]^n$	linear combination SCF
$\hat{\varrho}_{min} : [0,1]^n \rightarrow [0,1]$	minimum SCF
$\hat{\varrho}_{max} : [0,1]^n \rightarrow [0,1]$	maximum SCF
$\hat{\varrho}_{nr}$	SCF enabling absolute negative relevance feedback
$\hat{\varrho}_{r} : [0,1]^n \rightarrow [0,1]$	SCF combining remaining similarities of a $\hat{\varrho}_{nr}$

Temporal Query Execution

$\varsigma \in \mathfrak{S} := \langle \texttt{oid}, \texttt{start}, \texttt{end} \rangle$	a sequence inside an object
$\mathfrak{S}$	set of all sequences
$\hat{\varsigma} \in \hat{\mathfrak{S}} := \langle \varsigma, \tau \rangle\,,\ \tau \in [0,1]$	scored sequence
$\hat{\mathfrak{S}}$	set of all scored sequences
$\hat{\varrho}_{scs} : \left(\hat{\mathfrak{S}}\right)^n \times (\Phi)^{n-1} \rightarrow [0,1]$	SCF to score a candidate sequence
$\text{REW} : \mathfrak{S} \times \mathfrak{S} \times \Phi \rightarrow [0,1]$	reward function; adherence to user-specified distance
$\text{D}_{\varsigma} : \mathfrak{S} \times \mathfrak{S} \rightarrow \mathbb{R}$	distance between two sequences

Evaluation

$\mathcal{Q}_e$	set of all queries in the evaluation dataset

Dear Sir or Madam, will you read my book?
It took me years to write, will you take a look?

— The Beatles,
Paperback Writer

1 Introduction

The question of how to best organize and make available stored information has occupied humanity for millennia. Starting with the storage of clay tablets, to papyrus, novel ways to store information in an accessible manner are still an active area of research, for example DNA storage [DSP+20; BCQ+21]. This is necessitated by the ever-growing amounts of data which is consumed and produced in the past decades, with multimedia data such as images and videos contributing significantly to said growth.

The widespread availability of smartphones and wearables means that individuals produce data and content about themselves at an ever-growing rate. The rise of social networks has also blurred the line between consumers and producers of media, leading to a prosumer [Tof84] ecosystem where people leave large digital traces both in the public eye as well as on their private devices passively and actively.

Today, the most popular social networks [Ang21] are centered around experiencing multimedia data. Of those, YouTube[1] and TikTok[2] are built entirely for video consumption, albeit with different user interaction modalities. On Instagram[3], every post by a user must contain a picture or video, which can be augmented with text. TikTok, the fastest-growing social network is built entirely around algorithmic curation of user-generated videos.

In a professional context, applications relying on multimedia data are plentiful. Consider for example libraries with a mix of physical and digital media where retrieval is central to their utility, or museums which are also interested in novel ways of presenting and interacting with both ancient physical and novel,

[1] `https://youtube.com`
[2] `https://tiktok.com`
[3] `https://instagram.com`

digital-only exhibits [BMR+07; TLS+16; PSS+22]. Journalists writing articles require background information from archives or illustrative examples for their stories [MS98], companies have internal archives of meeting recordings, documentation, and work artifacts. Additional applications might include archaeology [KHZ09; BVL+22] and retrieval in medical imaging databases [MMB+04; SBL+16; IMP+22]

In this data abundance, people from all walks of life, researchers, librarians, doctors and creatives have motivated the quest for making this data accessible and searchable. One of the foundational vision texts for the digital age is Vannevar Bush's essay *As We May Think* [Bus45]. In it, he makes the case that after focusing their efforts on the second world war, scientists should tackle the challenge of making human knowledge and wisdom more accessible to individuals, as this is not only essential for advanced research in all fields, but also more general to wield knowledge for the "true good" [Bus45]. He envisions a device called "Memex" which can to store all information relevant to an individual, "books, records, and communications" [Bus45]. This is relevant for all types of media and applications, ranging from the more general such as video to narrower domains such as searching one's personal multimedia archive, or *lifelog* [DK07].

Research has tackled the task of finding relevant things under various labels such as *Information Retrieval* [Sal89; BR11][4] and *Multimedia Retrieval* [BBF+07], and as the complexity and diversity of multimedia data has grown, so has the need for more complex retrieval models. One of the driving forces behind this trend is the recognition that users have various information needs and enabling them to express those in a suitable way or *modality* is key. Additionally, traditional retrieval models have often focused on queries targeting small units of retrieval such as an image or a specific point in time of a video, a point also made in literature: "limitations [...] include retrieving shots only rather than larger units" [Sme07]. Similar to the need for more complex retrieval models, there is also a clear need for users to express complex and rich information needs easily. This is especially the case for information needs with a temporal context, which are common for both video and lifelogs, which are inherently temporal.

Additionally, the proposition that information retrieval should be "regarded as an inherently interactive/evolving process [Bat89; BMC93]" [LBB+22] has led

[4]The precise origin of the information retrieval field is described differently in various sources, one of the first books was [BH63], Mooers is said to have coined the term [Moo50], Salton [Sal68] is often cited as one of the most significant researchers in the early years [SC12], and Rijsbergen's book [van79] is often cited as an authoritative source of the early years.

to an increasing interest in the *interactive* aspect of multimedia retrieval, starting with the Relevance Feedback (RF) paradigm [RHO+98] and has paved the path for modern interactive and user-centric systems [KJR+20]. Against this backdrop of more complex retrieval models and interactive retrieval systems, the biggest change in the past decade has been the rise of Deep Learning, which has fundamentally changed the landscape of multimedia analysis [KSH17; ZZX+19] and retrieval [LXY+19; LVM+21; RGL+21; HGB+22] and other areas ranging from Chemistry [JEP+21] to board games such as Chess and Go [SHS+18]. Deep Learning has however not changed the fundamental problem of multimedia retrieval, which considers a user looking for the proverbial needle in the haystack.

Traditional algorithmic evaluation on state-of-the-art datasets, and system evaluations considering scaling behavior or other properties are useful tools which have served the community well when evaluating research contributions in this domain. Considering the trend toward interactive retrieval and the importance of understanding the whole user journey from query formulation to result browsing, these evaluations are complemented and enhanced by user-centric evaluations such as the Video Browser Showdown (VBS)[5] [HGB+22] and Lifelog Search Challenge (LSC)[6] [GJS+22], where user-system combinations are evaluated, and thus all aspects from user interface, to retrieval model and system efficiency are considered. Benchmarking and competition results also influence and facilitate the evolution of systems and methods, it is thus particularly important that they reflect sensible and realistic scenarios, as to avoid over-optimization. The information needs in those competitions ideally also reflect realistic queries users may make to system, for instance by including different presentation or communication modalities, and including temporal context in a way that is reflective of human perception and recall.

1.1 Focus and Significance of Research

As discussed, existing research has often focused on scenarios where a query is formulated in a single modality (e.g., text) and targets a specific unit of retrieval which is defined before the query (e.g., a shot in a video, or an image in a collection). In contrast, more complex retrieval models consider combinations of modalities and query models which enable users to express also temporal rela-

[5] VBS is a yearly evaluation campaign, we cite the 2022 review here which references the reviews from previous years.

[6] We also only cite the most recent LSC overview here.

tions. The need for such models has been well-established, such as in [HXL+11]: "Description of temporal relations between different kinds of information from multiple models, dynamic weighting of features of different models, fusion of information from multiple models that express the same theme, and fusion of multiple model information in multiple levels are all difficult issues in the fusion analysis of integrated models.". Earlier literature in the field of video retrieval has also explicitly called for research in this area, for example "As a unit of information a shot is [...] often sufficient for a user, but often it is not" [Sme07]. Looking at complex retrieval models which enable different modalities and concepts, there is a large body of work on combining multiple query modalities for multimedia retrieval [CWW+10; PT16; LZM18; Ros18; LKS+19b]. Textual embeddings using deep learning dominate retrieval benchmarks in recent years [LVM+21; HGB+22], yet successful systems allow to combine them with different modalities such as sketches [LKS20; LKS+19a].

Recent benchmarking campaigns have shown complex retrieval models which enable temporal context to be specified to be successful for interactive retrieval. One example is the analysis of VBS 2020: "The results reveal that the top two systems mostly relied on temporal queries before a correct frame was identified" [LVM+21] and 2021 "[...] almost all top performing systems [...] enable specification of temporal context in queries" [HGB+22]. However, there is little common terminology in the multimedia retrieval community around conceptual models on query formulation, execution, and evaluation of temporal information needs.

Tackling these challenges not only in theory, but in practice by building useful retrieval systems has been identified as a key challenge, e.g., in [DJL+08], where one of the four key questions for retrieval is "How can useful systems be built [...]?", or [JSG06], where the need for "human-centered multimedia systems" is motivated. Working with and advancing a fully-fledged retrieval system has also the advantage of being able to integrate state-of-the-art advances or develop novel retrieval methods.

In this thesis, we focus on two applications for our conceptual and implementation contributions, namely video and lifelog retrieval. The research community has long identified interactive video retrieval as a relevant field of research [AY99; HC04; SW09; Sch19], and systems meeting these needs are an active subfield. Lifelog retrieval is also a vibrant field where progress is driven through various benchmarking campaigns, such as the NTCIR-Lifelog task [GJH+19], ImageCLEF [NLZ+20] and the Lifelog Search Challenge

(LSC) [GLN+20; GJS+21; GJS+22]. We specifically choose video and lifelog retrieval because the temporal aspect of information needs is inherent and prominent in both those media types. Video retrieval also serves as an example of a broadly relevant media type relevant for many applications, whereas lifelogs represent a narrower and more specialized field. We focus on the visual domain as it dominates media consumption and production, and define our model in a generic way such that it is easily extendable to other media types which have temporal progression, and implement it in a multimedia retrieval engine, where it also works for audio content.

1.2 Contributions

Based on the previously identified research gaps and considerations, this thesis attempts to push the frontiers of multimedia retrieval and multimedia retrieval systems by presenting a retrieval model for complex information needs in interactive multimodal multimedia retrieval, which is based on an implementation in vitrivr, an open-source multimodal multimedia retrieval system. To evaluate our contributions, we have participated with vitrivr at interactive evaluation campaigns numerous times, and will present, contextualize and analyse insights from these evaluations in addition to a more traditional system-centric evaluation. In particular, this thesis makes the following key contributions:

- We present a retrieval model including data and query model for query formulation and execution in interactive multimodal video and lifelog retrieval. The model considers multimodal information needs with temporal components targeting different units of retrieval and includes novel algorithms for result fusion.

- The model is based on an implementation in a modular manner in vitrivr, a multimodal multimedia retrieval system which covers the entire user journey including extraction, query formulation and result presentation. vitrivr is now used in multiple large-scale interdisciplinary research projects [Wel22; LFF22], and has a healthy open-source ecosystem around it.

- To demonstrate the effectiveness of our model and implementation, we show results and insights from user-centric evaluations, where we have also made contributions to evaluation methodology. Additionally, we per-

form a system-centric evaluation, based on real-world data used in evaluation benchmarks, which looks at different model and system configurations.

The software contributions made in this thesis are fully open-sourced[7] with the aim of making them accessible to other researchers and practitioners.

The rest of thesis is structured along our key contributions. Chapter 2 will introduce our motivating scenario and its applications in video and lifelog retrieval, derive requirements for systems and model wishing to holistically address these scenarios, and map our contributions to these requirements which sets the scene for the remainder of the thesis. Afterwards, Chapter 3 will cover relevant foundations of multimedia retrieval and retrieval systems. The first contribution, our retrieval model for query formulation and execution of temporal multimodal multimedia retrieval queries is then described in Chapter 4. The chapter includes the data model, query model, execution model for multimodal queries, and execution model for temporal queries. During this dissertation project, significant contributions have been made to the multimedia retrieval system vitrivr. We describe the system architecture and implementation as our second contribution in Chapter 5. We evaluate different aspects of the model and its implementation in vitrivr in Chapter 6 and show results and insights from interactive benchmarking competitions, which marks our third key contribution. Related work is discussed in Chapter 7, and Chapter 8 concludes and gives an outlook to future work.

[7] `https://github.com/vitrivr`, `https://vitrivr.org`

Dreaming, after all, is a form of planning

— Gloria Steinem

2

Motivating Scenario

In this chapter, we will consider the two exemplary scenarios of information needs with temporal components in our two key applications, video and lifelog retrieval. They are both ad-hoc, spur-of-the-moment, instantaneous, information needs, where users are looking for a specific item in the collection. Both scenarios focus on *interactive* retrieval, defined as "[Retrieval] with users" [Kel09]. Modern information retrieval systems are interactive, and users can change their queries based on the interaction with the system. The interactive nature of the system also may make users reconsider their queries or even information need based on the results.

Users with such information needs are also referred to as *Searchers* in literature, who "are very clear about what [they] are searching for, [and their] session would typically be short, with coherent searches leading to an end-result" [DJL+08]. In the *search-exploration* axis [ZW14] of the *Multimedia Analytics* field [CTW+10], our motivating scenarios are clearly situated on the search side.

The first scenario is interactive video retrieval in Section 2.1, where a user might remember several parts of a desired video, and the second one is lifelog retrieval in Section 2.2, where information needs are based on memories and thus have an inherently temporal context. In Section 2.3, we derive requirements for a multimedia retrieval system addressing those scenarios, which we map to the contributions of this thesis in Section 2.4.

2.1 Video Retrieval

Consider the example of a documentary filmmaker, who is doing a documentation to shed light on the exploitation of wildlife in deserts. They remember

(a) A lion lying down

(b) A giraffe eating

(c) A group of elephants in a lake

Figure 2.1 Example sequence from video 07119 of the V3C [RSA+19] collection

having filmed a sequence where there were multiple animals visible in succession: first, a lion, then a giraffe, and the sequence closed on a group of elephants, as shown in Figure 2.1[1]. However, they cannot remember where the footage is located in their collection and thus would significantly benefit from a system which indexes their collection, and enables them to find this particular sequence without having to manually annotate all the footage. They may formulate their query in different ways, for example, using metadata such as the location where the footage was taken, textual descriptions ("a lion lying down in the desert"), and example images or sketches of the animals they are looking for. The filmmaker may also wish to specify the temporal context of their information need (i.e., the order in which animals appear in the footage), which should be considered during retrieval. This point is also made in literature: "It is the *combination* of shots what describes the story element, and each shot uses the context of the surrounding shots to convey its message" [Sme07]. Additionally, the filmmaker might not have a perfect memory of the sequence and remember it with partial context. In this context, the system should account for imperfect queries, and serve relevant results based on the specified query and context. To iterate on their query, the filmmaker should be able to reformulate it across all modalities and be able to seamlessly combine them, for example by specifying textual information from the metadata. They may also query based on other content-based features such as Optical Character Recognition (OCR) data or Automatic Speech Recognition (ASR) data.

After query formulation, the system should return relevant results which fulfill the information need. Presenting the results in a manner which makes browsing and exploration efficient for a wide range of users is an often under-investigated task. In addition to a ranking by relevance, our filmmaker might wish to order query results by video or as a timeline. The interface could also organize the results in a semantically meaningful way, for example in a hierar-

[1]Stills from this video [V3c] are used for examples in multiple chapters of this thesis

chical clustering, and enable filtering by metadata such as duration or content such as the number of animals. The system might also provide feedback to the user on how their query could be improved, or provide additional multimedia data from other domains such as related articles or images.

Given that the filmmaker has a very specific sequence in mind, the relevance of results is highly subjective. Another user might perceive results from other videos as relevant to the given queries, or formulate the queries when looking for the same sequence differently.

2.2 Lifelog Retrieval

The vision of having all of one's memories, and communications available for retrieval has been set forth and discussed by different researchers [SW10; GSD14; RTN22] and in popular culture [Wel11; Nai04]. In particular, the MyLifeBits [GBL+02] project which aimed to fill Bush's vision of the Memex [Bus45] discussed in Chapter 1 is widely considered to be the starting point for the field [GLN+20].

The motivation for lifelogging can be separated into two groups of users — so-called *lifeloggers* and people simply using digital services and devices. Lifeloggers wear specialized cameras to record every aspect of their life, which introduces considerable challenges for both research and everyday interactions such as anonymization and consent. More broadly, most individuals who have smartphones or wearables generate huge amount of data traces. Even just using E-Mail and online services means that a significant amount of personal data accumulates over the years. Children's upbringing is extensively documented by many parents, resulting in teenagers who have extensive documentation about their personal life. This means there is both interest in the narrow lifelogging community and from the broader public in efficient and privacy-preserving ways to search one's personal data. In the past decade, modern devices have made searching one's lifelog much more accessible by automatically adding sensor data such as GPS, identifying objects or scenery such as „beach“, and detecting faces [CZK+21], which makes queries like "show me all images from the beach in Barcelona" accessible to everyday users [App22; Goo22]. Lifelogs have a few special properties. They almost always consist of different media types, ranging from the visual to sensor metadata and inherently exhibit a temporal progression. Not only vary information needs significantly based on the user, but also the way in which they can best express their query, and the rel-

evance results depends significantly on the individual which is using a lifelog retrieval system [GSD14].

Beyond the visual lifelog, audio also plays an important role when remembering moments. While audio processing is its own research domain, for retrieval there are two important aspects: transcription (Speech-to-Text) and annotation (e.g., speaker recognition, conversation segmentation). These two enable querying not just for content, but also context of audio recordings. If audio and video is recorded jointly, similar and even stronger arguments as were made in the previous section apply. Processing and doing retrieval in both domains jointly enables much more expressive queries and thus more useful functionality.

One active research problem in lifelog retrieval is retrieving days or events based on more complex descriptions, such as "I had breakfast at Starbucks, then walked to the office and afterwards had a two-hour meeting. It was the week after I came home from Canada". These can also be considered ad-hoc information needs with a temporal component, similar to the video retrieval scenario. Particularly for lifelog retrieval, enabling users to formulate their information needs using the modalities most suited to them personally is critical. Additionally, the retrieval model should be robust to incomplete descriptions and queries, as humans forget things over time [Ebb85; MD15].

2.3 Deriving Requirements

From these two example scenarios and the context of multimedia, we can derive requirements for systems and models which wish to address the multimedia retrieval problem. These requirements are additionally based on those already identified in literature [BBF+07; BR11; CHM+19].

Multimedia: As already specified in its name, a multimedia retrieval system should support different types of media, ranging from video and images to audio or 3D. A system could also consider composite objects consisting of different media types.

Different Modalities: Different types of media call for different ways of query expression. Different users have different capabilities and preferences when it comes to expressing their information need. A multimedia retrieval system should therefore enable users to express their information need in different ways, and map those expressions to a unified query model which combines

those modalities seamlessly. Both the expression of information needs and the mapping to queries should keep in mind various gaps between information need and abstract representation, for example the semantic gap between the interpretation of a user and the machine-extractable information.

Temporal Context: Most multimedia data, such as video and audio, exhibits a temporal progression. If the multimedia data itself does not exhibit one, multimedia collections have temporal aspects such as creation, modification, and deletion dates, or they are clustered by events which have temporal context. Beyond the data, information needs of users also have temporal context. An example of this would be a lifelog collection, which may consist mostly of images which do not have a temporal aspects, but their context and the collection itself is inherently temporal. A multimedia retrieval system should therefore enable users to specify their information needs in a semantically meaningful way, and consider temporal aspects in result presentation.

Targeting Different Levels of Abstraction: Multimedia data is inherently composed of different elements, and can be completely unstructured or have structure defined by metadata or automated content analysis. A video consists of shots, which in turn consists of frames. A single frame or image itself can be described at various abstraction levels: color, semantic content, spatial relationships between semantic content. Audio has temporal and potentially spatial aspects, and different forms of audio (conversations, music, long-running sensor audio) have different levels of abstraction. Text itself can be structured in different ways, and cross-reference other textual or multimedia data. This means that a multimedia retrieval system should enable retrieval modules which target various levels of abstraction, and users to compose their queries accordingly.

Notion of Relevance: In contrast to traditional Boolean queries which have a binary notion of relevance, queries concerning multimedia data rely on a notion of relevance which depends on both the user and the method used to compare a query to a multimedia element. Therefore, a multimedia retrieval system must support a non-binary notion of relevance at all levels, from query formulation to query execution and result presentation.

User Journey: Users come with an information need to a system, formulate their query, browse results and iteratively continue their process until they are satisfied. Expert users may also wish to configure which features should

be extracted for their data, as they possess insights into which methods may be relevant for their scenario. A multimedia retrieval system should therefore support the whole user journey including extraction, query formulation, query execution, and result browsing.

Interactive Retrieval: Even in a Known Item Search (KIS) scenario, the retrieval process is an inherently interactive one as users refine their queries to find the desired item. Beyond KIS, users might wish to explore a collection or summarize it partially, in which case interaction is even more important. Multimedia retrieval systems should thus enable their users to refine queries efficiently, and ideally have mechanisms for relevance feedback.

Additionally, we could consider requirements such as scalability, privacy, integration with external sources and learning over all user interactions, but we do not elaborate on those in more detail as they are beyond the scope of this thesis and the motivating scenario presented here.

2.4 Mapping Requirements to Contributions

Looking at our contributions described in Section 1.2, we can map them to the requirements which we have outlined in the previous section.

Model for Query Formulation and Execution: Our model has at its heart a *notion of relevance* or similarity, and the entire retrieval model is based around the question of how to best determine and combine similarity. It combines *different modalities* through different fusion schemes, enabling users like our filmmaker or people searching their personal data to express queries in a manner of their choosing. The model is agnostic to segmentation and thus can *target different levels of abstraction*. It deals with *temporal context* of *multimedia* data at query level by enabling users to specify temporal context, at query execution level through a formalization of problem setting and algorithms for late fusion, and considers the result presentation step. This is relevant for both scenarios, as both videos and lifelogs are inherently temporal.

The model does not cover all requirements and aspects of the scenarios exhaustively, for example textual data and metadata-heavy collections are not considered in depth. Our focus is on video, which has an inherent temporal progression, and visual lifelogs which are segmented not by event, but by time. This means that non-visual media is not the focus and the temporal

aspects of multimedia collections are not covered in depth. Our model is however generic enough to also support audio in the implementation. Additionally, a theoretical model for the *user journey* which considers interaction and past queries for query execution and scoring is beyond the scope of this work. We demonstrate how relevance feedback and user preferences can be incorporated into our query and execution model, but do not investigate this in more depth.

Implementation in vitrivr: vitrivr covers the *user journey* from extraction, to query formulation and browsing and enables *interactive retrieval* through easy (re-)formulation of queries and late filtering of result sets. This means it offers an end-to-end solutions for users seeking to index and query their personal collections. The implemented retrieval model combines *relevance* for content-based queries with Boolean queries. vitrivr supports through a combination of prior work, improvements made in this dissertation project, and contributions by others *different modalities* such as query-by-sketch, query-by-pose, textual queries, Boolean queries and others. These modalities can be combined to query with *temporal context*. It also supports different *multimedia* types, namely video, images, audio, and 3D [GRS19b].

The existing segmentation for video [RGS14] and other segmentations enable querying for *different levels of abstractions*, but the implementation currently does not cleanly support multiple segmentations for one multimedia object. Additionally, the limitations from the model section with regard to a user model for personalized information retrieval are also applicable in the implementation.

Evaluation: The evaluation contributions touch all requirements, with the user-centric evaluation being specifically meaningful for the *user journey* and *interactive retrieval* in *multimedia*, and the system-centric evaluations showing the feasibility and effectiveness of model and implementation for the *notion of relevance* and *temporal context*. Through this combination, we combine a more traditional evaluation with results from benchmarking campaigns focusing on a more holistic comparison and involving both expert and novice users.

Errors like straws upon the surface flow;
He who would search for pearls must dive below

— John Dryden,
All for Love. Prologue [p. V]

3

Foundations of Multimodal Multimedia Retrieval

In this chapter, we discuss the theoretical models behind multimodal multimedia retrieval and systems. Generally speaking, multimedia retrieval is a subfield of *Information Retrieval (IR)*, with the *IR Problem* being defined as follows [BR11, p. 4]:

> [T]he primary goal of an IR system is to retrieve all the documents that are relevant to a user query while retrieving as few non-relevant documents as possible.

Definitions of the *Multimedia Retrieval (MR)* problem are often very similar and additionally emphasize the multimedia nature of the documents [BR11, p. 588]

Both this chapter and the next chapter of this thesis follow the fundamental aim of an Information Retrieval (IR) model, which is "producing [...] a function that assigns scores to documents with regard to a given query" [BR11, p. 57]. Over the chapters, we consider different abstractions for *document* both from a query and result perspective, and the term *function* is loosely used, as complex models are not particularly suited to a singular function representation. The aim of this chapter is to lay the groundwork for our model chapter, which means the focus and notation is tailored toward relevance in the context of this thesis.

In this chapter, we begin by going from the user to a formulated query in Section 3.1, have a look at different retrieval models in Section 3.2, provide a high-level conceptual view on retrieval systems in Section 3.3, show different query modalities in Section 3.4, and wrap up by discussing foundations of how different query modalities and retrieval models can be fused in Section 3.5.

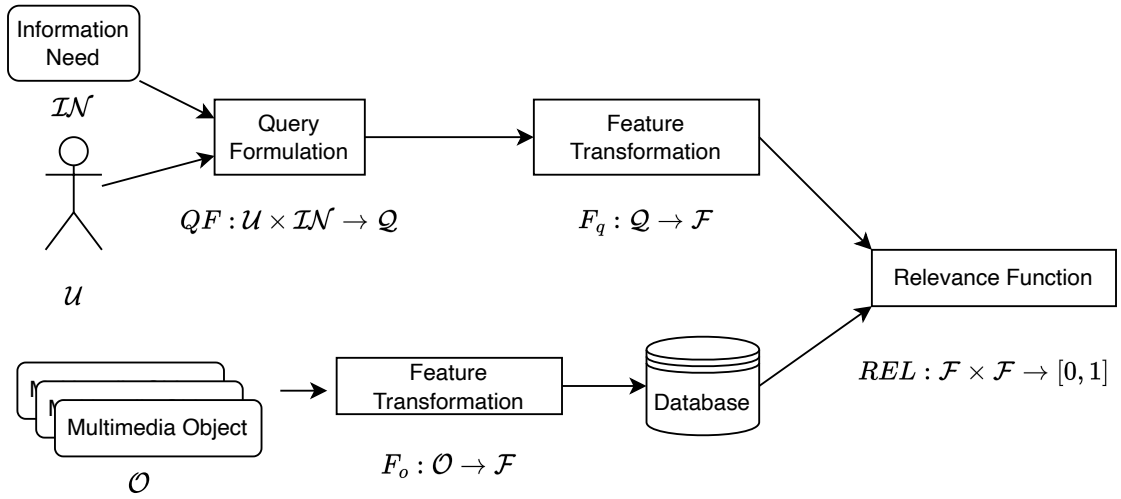


Figure 3.1 Visual overview of the retrieval process

3.1 From Information Need to Query Expression

A visual overview of the process discussed in this and the next section is shown in Figure 3.1. Fundamentally, a user formulates a query, which is along with the multimedia objects transformed into a common space, where relevance is evaluated. The transformation of objects into features and their storage in a database is often described as the *offline* part of retrieval, whereas the query formulation and relevance evaluation is *online*. We will revisit this separation later when we take a system perspective on retrieval.

We start with a user $u \in \mathcal{U}$ who wishes to utilize a retrieval system and their information need. The literature does not have a consensus on the definition of information need, see [CG16, p. 68–83] for an extensive discussion of the term and its various meanings. For our purposes, we define an information need in Definition 3.1.

Definition 3.1 Information Need

An information need $in \in \mathcal{IN}$ is "the actual, unexpressed, need for information" [Tay62] with which a user approaches the retrieval system.

Given an information need, it is up to the user to transform their information need into a representation which is understood by the retrieval system. We call this a *query*, see Definition 3.2.

Definition 3.2 Retrieval Query

A query $q \in \mathcal{Q}$ is a representation of a user's information need. It is used as a request to an IR system with the goal of retrieving relevant results.

The information need is transformed into a query in the query formulation step described in Definition 3.3.

Definition 3.3 Query Formulation

The query formulation step QF : $\mathcal{IN} \times \mathcal{U} \rightarrow \mathcal{Q}$ maps an information need $in \in \mathcal{IN}$ by a user $u \in \mathcal{U}$ to a query $q \in \mathcal{Q}$.

The resulting query depends thus on the information need, the user's ability to express their information need, and the interaction modalities offered by a retrieval system.

A query can be of arbitrary complexity, containing and combining different modalities (e.g., free text, Boolean predicates, color sketch). We discuss different query formulation modalities later in Section 3.4, and fusion for those in Section 3.5. The query formulation step from information need to query is fundamentally subject to various gaps between information need and the actual query, as briefly reviewed in Literature Discussion 3.1.

Literature Discussion 3.1 Gaps From Information Need to Query

[SWS+00; DJL+08] identify two key gaps: the sensory gap (between the real-world object and its computational description) and the semantic gap: "The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation." [SWS+00]. [Gia18] identifies an additional *expressive gap* between "the content a user perceives, [...], the concepts which are [...] detectable by the system, and [...] ability to express these concepts", which could be argued represent two different gaps. [Ros18] additionally identifies a *perceptive gap* (perception of a situation), *interpretative gap* (interpretation of a situation), and *mnemonic gap* (between the actual situation and the memory of it).

Because of those gaps and the complexity associated with bridging them, most retrieval models and systems have as their starting point the actual *query*.

From an interaction perspective, users might update their queries or include relevance feedback in them. This leads us to the next section, where we discuss different retrieval models and how they determine relevance.

3.2 Retrieval Models

In this section, we first define basic notations for retrieval models based on existing work, and then briefly review two prominent retrieval models, vector space retrieval and Boolean retrieval together with relevant examples. For the purposes of this section, we focus on queries which only contain one modality and within this modality are focused on one domain within which relevance can be determined. Complex queries with different modalities are discussed later in Section 3.5.

3.2.1 Overview

In multimedia retrieval, we define the set of all multimedia objects (e.g., images, videos) O which are stored in the system and a single multimedia object $o \in O$.

Our definitions of both a query and the IR Problem had the term *relevance* at their heart. To determine the relevance of an element w.r.t. a query, we thus require a compact representation of both the query and the element we wish to evaluate its relevance for. To this end, we define a feature $f \in \mathcal{F}$ in Definition 3.4.

Definition 3.4 Feature

A feature $f \in \mathcal{F}$ is defined as "derived characteristics" [BBF+07] of an element.

As a simple example for a feature, consider a vector representing the average color of an image. To determine relevance between an object and a query, we need to transform both into their common domain. This transformation is defined in Definition 3.5.

Definition 3.5 Feature Transformation

A query is transformed to a feature with the query transformation function $\mathrm{F}_q : Q \to \mathcal{F}$, and an object with the object transformation function $\mathrm{F}_o : O \to \mathcal{F}$.

To make this chapter concise, we consider an object as the unit for which relevance is determined. As discussed in the previous chapter, more sophisticated

retrieval models enable different levels of abstraction which can be queried and returned as results. Having mapped both query and media object into a common domain, we define a relevance function in Definition 3.6.

Definition 3.6 Relevance Function

A relevance function REL : $\mathcal{F} \times \mathcal{F} \rightarrow [0,1]$ determines the relevance of one feature w.r.t. the other. 0 indicates no relevance, 1 indicates perfect relevance.

This means that any notion of relevance in retrieval relies on the representations of the queries and objects and not on the queries and documents themselves [CLVR+98].

Since relevance is hard to define, we often first calculate dissimilarity or distance as it can be easily expressed as a mathematical function for many features.

Definition 3.7 Dissimilarity Function

A dissimilarity function DS : $\mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}_{\geq 0}$ indicates how far apart two features are in the feature domain.

Afterwards, we apply a correspondence function [CPZ98] as defined in Equation (3.1) to our dissimilarity.

$$C : \mathbb{R}_{\geq 0} \rightarrow [0,1] \tag{3.1}$$

In these cases, we define $\text{REL}\left(f_q, f_o\right) \mapsto C(\text{DS}(f_q, f_o))$. Correspondence functions need to have the following two properties [CPZ98]:

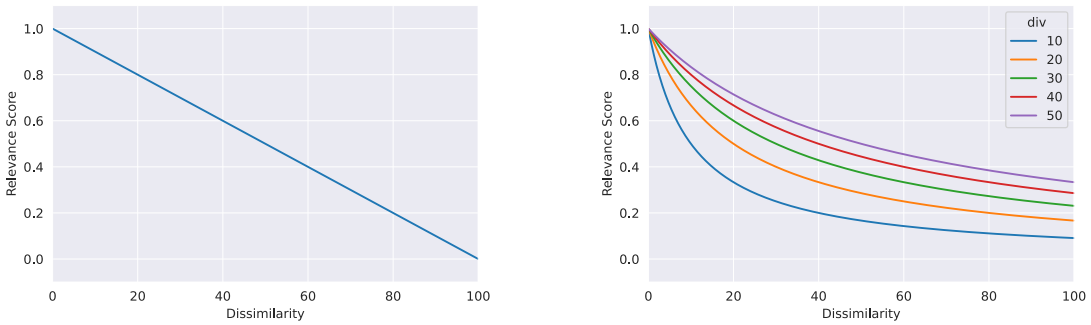
$$C(0) = 1$$

$$x_1 \leq x_2 \Rightarrow C(x_1) \geq C(x_2) \qquad \forall x_1, x_2 \in \mathbb{R}_{\geq 0}$$

We show two examples of correspondence functions in Example 3.1.

Example 3.1 Correspondence Functions

Assuming we are given a dissimilarity $\delta \in \mathbb{R}_{\geq 0}$, two examples of correspondence functions $C : \mathbb{R}_{\geq 0} \to [0,1]$ are the linear correspondence function $C_{lin}(\delta, \texttt{max}) \mapsto 1 - \frac{\delta}{\texttt{max}}$ and the hyperbolic correspondence function $C_{hyp}(\delta, \texttt{div}) \mapsto \frac{1}{1+\frac{\delta}{\texttt{div}}}$. We visualize them in Figure 3.2, with the dissimilarity on the x-axis and the resulting relevance score on the y-axis.



(a) C_{lin} **with** max = 100 **(b)** C_{hyp} **with varying** *div*

Figure 3.2 Visualizations of two different correspondence functions

As intuitively expected, both result in a lower relevance score for increasing dissimilarity. C_{hyp} is especially useful for cases where close proximity is essential, for example with geospatial features [BS16; SRS18] as used in the lifelog retrieval modules implemented in the course of this dissertation project [HGP+21].

As the goal is often to get all or the most relevant elements, retrieval can be sped up using index structures which do not have to evaluate relevance per item, but instead use heuristics. We will not discuss these here in detail, as our model in the next chapter is agnostic to whether index structures exist for a given retrieval module or not. Before ending this section, we review other perspectives on retrieval models found in literature in Literature Discussion 3.2.

Literature Discussion 3.2 Other Perspectives on Retrieval Models

[BR11, p. 58] define an IR-model as a quadruple $\langle D, Q, F, R(q_i, d_j) \rangle$. D is a set of representations of the documents in a collection (O in our notation). Q is a set of representations of the user information needs called queries (Q in our notation). F is a framework for modeling representations, queries and everything else (the transformation functions associated with a feature in our

notation), and $R(q_i, d_j)$ is a ranking function that associates a real number with a query representation $q_i \in Q$ and a document representation $d_j \in D$. This ranking defines an ordering with regard to the query q_i.
[SWS+00, p. 1365–1369] define a query space with the following components: documents (images), features, a similarity function, and a set of labels for goal-dependent semantics, the first three of which are closely aligned to our notation.
Fuhr [Fuh92] has the elements $\langle D, Q, r, \alpha, \beta \rangle$ where D are the documents, Q the queries, and r all possible relevance judgments ($[0, 1]$ in our notation). α derives representations from queries and documents (which would be the Q and O in our notation) and β derives descriptions from these representations, on which a retrieval function can be applied. These descriptions are the features $\mathcal{F}$ in our model.
Most retrieval models found in literature are similar in nature to very early work [Coo76], which defines $\langle I, R, V, T \rangle$ where I is a set of representations of all documents (in our notations, the features that were derived from O), R all "search prescriptions" (queries), V the set of "retrieval status values" a system can produce (relevance judgments, [0,1] in our notation) and $T : R \times I \rightarrow V$ the retrieval function (relevance function in our notation).
There is also work on probabilistic models, in which relevance is additionally estimated based on the probability of relevance to a query. We do not consider these in the scope of this thesis, and refer interested readers to surveys [CLVR+98] and more recent work [BL17].

Most of the presented retrieval models assume that the unit of relevance is a document, and do not explicitly consider the case in which the documents in the collection, the unit for which a query is formulated, the unit for which relevance is determined, and the unit which is presented, may differ. As discussed in the previous chapter, enabling users to query different layers of abstraction according to their information needs could have significant benefits for more complex information needs, and we will revisit this requirement in our model in Chapter 4.

In the remainder of this section, we discuss two common retrieval models which are also used in our retrieval model and implementation, and how they determine relevance along with simple examples. We begin with vector space retrieval in Section 3.2.2 and continue with Boolean retrieval in Section 3.2.3.

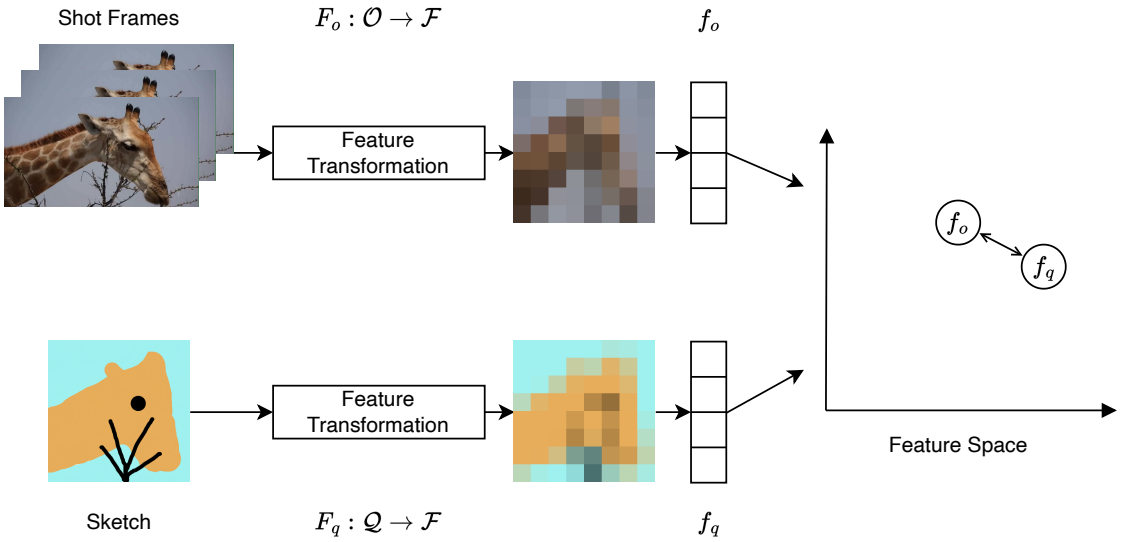


Figure 3.3 Illustration of how vector similarity is used and evaluated in a simple feature for a user-provided sketch. Sketch from [Gst21]

3.2.2 Vector Space Retrieval

In the vector space model [SWY75], features are high-dimensional vectors, with $f \in \mathbb{R}^n$. This has the advantage that dissimilarity between two vectors can be easily expressed as a mathematical function, with a wide range of options for the dissimilarity function $\mathrm{DS} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$ such as the Euclidean distance, which is defined for a query vector f_q and an object vector f_o as follows:

$$\mathrm{DS}_{l2}\left(f_q, f_o\right) \mapsto \sqrt{\sum_{i=1}^{n}\left(f_{q_i} - f_{o_i}\right)}$$

We give an example for a simple color sketch feature which relies on the vector space retrieval model in Example 3.2, which is illustrated in Figure 3.3.

Example 3.2 Sketch-based Video Retrieval

To find a sequence containing a giraffe, a user provides a sketch. A simple feature for sketch-based retrieval shown in Figure 3.3 reduces both sketch and most representative frame are to an 8-by-8 grid, where the distance between the two vectors is computed using the Manhattan distance function which is defined as follows:

$$\mathrm{DS}_{l1}\left(f_q, f_o\right) \mapsto \sum_{i=1}^{n}\left|f_{q_i} - f_{o_i}\right|$$

The vector space model was originally developed for textual data, where textual documents are mapped to a vector in which each element represents a specific term and its frequency, which is potentially normalized [Dom08, p. 163]. It can be also applied to various types of multimedia and features, such as textual embeddings of visual data [SGH+22], and color sketches [Ros18]. Different dissimilarity functions are suited to different applications and feature spaces, for a comprehensive evaluation of dissimilarity functions for multimedia retrieval, we refer the reader to [Ros18].

We do not discuss the vector space retrieval model for text in further detail as our conceptual model in Chapter 4 is agnostic to which feature is used for text retrieval and the implementation discussed in Chapter 5 uses one of the most popular libraries, Lucene [Fou21]. Lucene combines Boolean retrieval with vector space retrieval by pre-filtering for Boolean matches before scoring with vector space retrieval. This is extremely useful both for textual metadata as present in our Lifelog scenario, and multimedia analysis modules which output textual data such as OCR [SBY17; arg22] or ASR [HCC+14; Moz22; RKX+22].

3.2.3 Boolean Retrieval

The Boolean retrieval model is the foundational model of information retrieval, and has been criticized and extended countless times by researchers. At its core are three basic operators: AND, OR and NOT. Both the multimedia objects and the query are represented as a set of terms, and the results contain all objects which match the specified terms and their respective operators [Dom08, p. 126].

As a simple example, consider metadata about an audio collection, specifically a textual description per song containing information about title, band, and release year. If we wish to retrieve all songs by "The Beatles" or "The Rolling Stones" but specifically not "Paperback Writer", our query would be ("The Beatles" $\vee$ "The Rolling Stones") $\wedge \neg$"Paperback Writer". It is easy to see how this model has many disadvantages, with the two main being the inability to rank documents and the rigidity of the operators [BBF+07], yet it also has the benefit of clarity and being easily expandable with other operators and wildcards for text retrieval.

There are extensions to the Boolean retrieval model which enable ranking such as the p-norm extended Boolean model [SFW83] or the probabilistic model [BBF+07, p. 106–110], which we will not cover in further depth.

In the context of our work, we will also be using the terminology of Boolean retrieval when talking about more traditional database queries as known from

relational algebra [Cod70] or SQL. This is specifically useful when we consider structured or semi-structured metadata as available in the context lifelog retrieval or most modern multimedia collections. We will not fully introduce the relational data model here, for recent comprehensive overviews with examples in the context of multimedia retrieval we refer to [Gia18; Gas23]. Our conceptual model is largely independent of the specific semantics of a query term, and in the context of our implementation contributions [RGH+19; HPG+20], we use standard operators such as $=, \leq, \geq$, or IN for structured metadata and LIKE for full-text retrieval.

3.3 A Conceptual View of Retrieval Systems

Having discussed the path from information need to query and different retrieval models, we turn to a systems perspective on multimedia retrieval. As discussed in the previous chapter, support for the user journey and interactive retrieval necessitate a system view, and the other requirements also lend themselves to being considered from this perspective. In this section, we discuss the fundamentals of multimedia retrieval systems, which will lead into the next sections with a conceptual discussion of query formulation methods and fusion for complex queries.

In general, retrieval systems consist of three core components: a user interface for formulating queries and browsing results, an application layer and a data management system. We show an architectural view in Figure 3.4. This separation makes sense from a variety of perspectives: It follows the traditional three-tier architecture of information systems [Sch18], is in line with retrieval system ideas developed on the basis of the MPEG-7 standard [HBH+04; KAG10], and is also widely advocated in literature [Fuh12; Fuh14; JWZ+16; Gia18; Ros18; Gas23].

Traditionally, a separation has been made between an *offline* phase where data is ingested, parsed, analyzed and stored in the database layer and an *online* phase, during which data is retrieved. While this assumption has been significantly challenged in recent years due to the dynamic nature of the applications in which multimedia retrieval systems are used [Gas23], we still include it in this chapter as it also helps to understand the different parts of a retrieval system, but will talk of parts instead of phases to highlight that ingestion, extraction and retrieval can happen in parallel.

In the offline part, data which should be retrievable is provided to the sys-

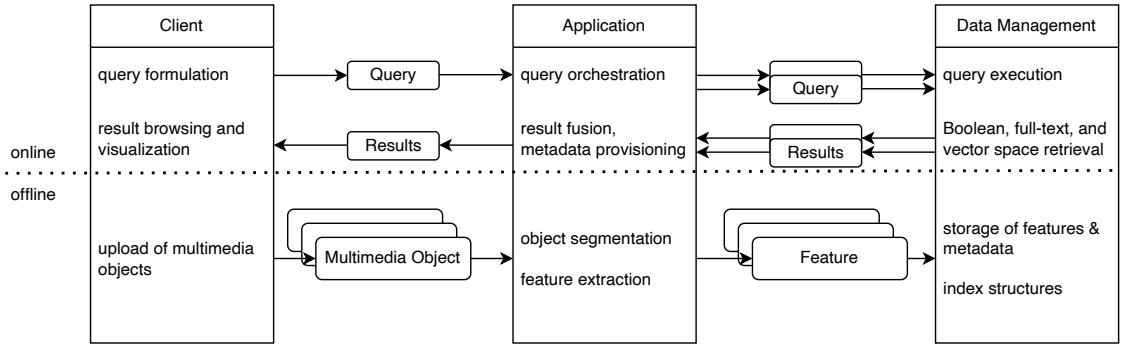


Figure 3.4 Model of a general purpose multimedia retrieval system. Based on [Gia18; HGG+23; HSS23]

tem through an Application Programming Interface (API), Command-Line Interface (CLI) or User Interface (UI). Features are then extracted from the multimedia objects and stored along metadata in the database. The data management layer should support both structured and semi-structured data, relevant retrieval models such as vector space retrieval, full-text search and Boolean retrieval in addition to traditional Database Management System (DBMS) capabilities [Gia18; Gas23].

In the online part, information needs are mapped to a system query in the query formulation process as defined in the previous section. This can be done through various modalities, which will be described in Section 3.4. The query formulation options offered by the system not only determine the available modalities, but through those also the retrieval models and the ability of the user to express their information need. The application layer (also called retrieval engine in this thesis) is responsible for executing queries in an efficient and effective manner, and returning meaningful results to the client. This necessitates a clear model of determining relevance, and supporting complex queries.

Conceptually, this thesis introduces data, query and retrieval models in Chapter 4 and thus touches all three areas of multimedia retrieval systems, with a focus on the application layer. In the implementation presented in Chapter 5, the contributions are mainly in the user interface and application layer, with minor contributions to the data management layer. While the conceptual contributions of this thesis are independent of the user interface modality itself (e.g., desktop, mobile) and the implementation contributions are in a traditional desktop UI, we will later briefly review different interface options in Section 7.1.2.

3.4 Query Modalities

The literature has no unified definition of terminology and distinction of query modalities, which could be defined as *different ways to express an information need*. In this section, we cover different ways to formulate queries from a user perspective. We briefly review some categorizations from relevant literature in Literature Discussion 3.3, then discuss textual, sketch and Boolean queries in more detail, and review novel query modalities at the end of this section.

Literature Discussion 3.3 Query Modalities

[DJL+08] distinguishes between *modalities* and categorizes them into keywords, free-text, image, graphics (sketch), and composite queries, which are grouped together with interactive queries. [HXL+11] distinguishes between *query types* with query-by-example, query-by-sketch, query-by-object, query-by-keyword, query-by-natural language and combination based queries (also called multimodel search) as examples. [Sme07] distinguishes between metadata and browsing keyframes, text (ASR, OCR), keyframe matching (query-by-example), semantic features, object-based video retrieval and a combination of the above. Most recently, in comparison of interactive evaluation campaigns [LVM+21; HGB+22], the categories free-text search, object / concept detection, image search, sketch search, fusion of modalities, temporal queries and relevance feedback are being used to categorize the different participating systems, with the first five having a mapping to aforementioned literature.
Generally speaking, literature often distinguishes between query-by-example and query-by-sketch, but we do not review query-by-example in this section as it is not a prominent feature of modern retrieval systems anymore. Object/concept detectors remain a popular feature [HGB+22], and keyword queries are either directly mapped to the output of such detectors or are then used as a free-text field for OCR, ASR, or textual embeddings.

3.4.1 Textual Queries

Text is often described as "the universal interface" [Ray03], so it is unsurprisingly an extremely popular and often used input modality for queries. From web-scale search engines to library search systems, both large- and small-scale systems offer users the option to simply write text and process it in such a way that it produces relevant results. In the multimedia retrieval context, systems

sometimes offer the user to explicitly distinguish between the meaning of the text, for example if it is referring to OCR, ASR, or textual embeddings [LXY+19; BNV+21]. Other systems try to parse free-text fields and map them to different retrieval features (e.g., by separating Boolean filters such as the weekday from content-based queries [TNN+22]).

The actual retrieval model which is used when evaluating a query with textual input differs based on the feature, for example OCR is often used with text indexing models as described previously, while textual embeddings use vector-based similarity search as introduced in Section 3.2.2. As discussed previously in Literature Discussion 3.1 though, there are various gaps between a user's mental model of their information need, their ability to translate said need into textual form and the ability of a system to make use of this expression. This means that other query modes might be better choices for specific scenarios.

3.4.2 Sketch Queries

Drawing a sketch of an information need is a relatively intuitive modality, and sketch-based image retrieval is an established field of research [RHC99; DJL+08; LL18].

Sketches serve as an excellent illustration of the various gaps between information need and query formulation mentioned earlier. Similarity to a sketch can be interpreted on the basis of shape [Can86; PJW00], color [CMN04], implicit semantics [SSX+16; SDM17] or even explicit semantics [RGS19]. Beyond the gaps mentioned previously, there are sketch-specific gaps [LL18], namely visual cues (sketches only have symbolic colors and little details on shape) and content imbalance (sketches have in most cases no background). Additionally, sketches can be representations of 3D objects [BGS+20], opening up interesting questions for multimedia retrieval beyond video, images, audio and text. For a more in-depth discussion on the fundamentals and limits of query-by-sketch, we refer the reader to [Ros18, p. 10–16]. In our example for vector space retrieval (Example 3.2), we showed how a sketch query could be used to express an information need.

3.4.3 Boolean Queries

When discussing differences between the world of Databases and Information Retrieval, we often make a distinction between *matching* and *relevant* results or *partial* and *exact* matches [van79]. However, sophisticated retrieval systems often

offer users the ability to enhance or complement their queries with components which have a binary relevance judgment. Examples for such queries might be based on metadata (e.g., "only images within the last year"), or checkboxes (e.g., "only videos") or Safe Search functionality in Google, which hides content deemed explicit). This is especially relevant in lifelog retrieval, where content-based information needs are complemented through metadata-based ones (e.g., "I was at a conference in Korea last year"), and also underscores the need for query models which enable a combination of modalities.

3.4.4 Novel Query Modalities

In the past decades, various novel and innovative query modalities have been considered for multimedia retrieval. One example is querying by motion [KPZ+04] (e.g., specifying that a bird flies to the right in a video [RGS+15]), other examples include querying for the constructed pose of a person [CHS+19; HAG+22], query-by-humming [GLC+95; KNK+99], querying for the pose a person is currently making in front of a camera [HCC+15], recording parts of a song one is looking for [Wan03; Wan06], sketching 3D shapes in the air [LLG+15; ZSY+17], querying for hand gestures [APRS+20; PWR+21], querying by voice [GGR+17; ARG22], and sculpting 3D shapes in VR [GJS18; BGS+20].

Continuing research on existing and novel query modalities necessitates generic retrieval models which can combine different modalities, which is covered in the next section, Section 3.5.

3.5 Complex Queries

Given that users may query using different modalities, and that those modalities can be interpreted by a retrieval system using different features and retrieval models, the task of combining these models, and generating a ranking from the results is central to retrieval. In this section, we review the fundamentals of complex queries and result fusion.

The underlying assumption behind fusion schemes should be that the individual result sets to be combined have "high performance, a large overlap of relevant documents, and a small overlap of nonrelevant documents" [VC99]. Three key effects help methods which use fusion [Dia98; VC99]:

The Skimming Effect: As different methods retrieve different multimedia documents, taking the top-ranked elements for each method (i.e., *skimming*) can

benefit both recall and precision. Recall is potentially enhanced through the diversity of relevant documents and precision, assuming highly ranked documents tend to be more relevant than lower ranked ones.

The Chorus Effect: A high ranking by different methods for a single item (i.e., a *chorus* instead of a solo) indicates higher confidence in its relevance than if it is just highly ranked by a single method.

The Dark Horse Effect: Retrieval methods may be unusually accurate (or inaccurate) for some queries or documents. Fusion schemes could make use of this effect by dynamically adjusting weights based on the document or query at hand and overweighting the method assumed to work best.

These effects are at odds with each other — skimming too much reduces the chorus effect, the dark horse effect argues for reliance on individual methods while the chorus effect argues for weighing all methods, making fusion an interesting and complex area of research.

For the purposes of this section, we assume that we have scored lists, that is result lists where the individual elements are also assigned a relevance score or dissimilarity. This enables us to do score-based fusion. If the result lists are only ranked, rank-based fusion can be used [RS03; FKS03], which we do not cover, but all methods for rank-based fusion can be used for scored lists as well assuming there is a tiebreaker mechanism.

One of the fundamental concepts of fusion for similarity queries are so-called Distance-Combining Function (DCF) as defined in Definition 3.8. They allow users to express different semantics when combining multiple queries. As discussed previously, a dissimilarity or distance δ is often easier to compute than relevance, which is why DCF operate on distances.

Definition 3.8 Distance-Combining Function (based on [BMS+01])

A Distance-Combining Function (DCF) $\hat{\delta} : (\mathbb{R}_{\geq 0})^n \rightarrow \mathbb{R}_{\geq 0}$ calculates a single distance from n different distances. We label individual distances of a DCF using $\overline{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$, with $\delta_i \in \overline{\delta}$.

There are many different DCF imaginable, we will list some in the following based on and adopted from [SF94; BMS+01].

max: Sometimes called "fuzzy-and" [BMS+01], the maximum distance may be selected when closeness to all subqueries is considered important, formally:

$$\hat{\delta}_{max}\left(\overline{\delta}\right) \mapsto \max_{i=1}^{n}(\delta_i)$$

min: Sometimes called "fuzzy-or" [BMS+01], the minimum distance may be selected when closeness to one subquery suffices, formally:

$$\hat{\delta}_{min}\left(\overline{\delta}\right) \mapsto \min_{i=1}^{n}(\delta_i)$$

sum:

$$\hat{\delta}_{sum}\left(\overline{\delta}\right) \mapsto \sum_{i=1}^{n}(\delta_i)$$

median:

$$\hat{\delta}_{med}\left(\overline{\delta}\right) \mapsto \text{median}(\overline{\delta})$$

anz: The average of all distances below a threshold ε ignores queries which are considered to be failing and can thus be viewed as a variation of an average of non-zeroes[1], formally:

$$\hat{\delta}_{anz}\left(\overline{\delta}, \varepsilon\right) \mapsto \left.\frac{\sum_{i=1}^{n}\tilde{\delta}_i}{\left\|\tilde{\delta}\right\|} \;\right|\; \tilde{\delta} = \left\{\delta_i \in \overline{\delta} \;\middle|\; \delta_i \leq \varepsilon\right\}$$

mnz: This fusion scheme rewards items retrieved by multiple methods[2], formally:

$$\hat{\delta}_{mnz}\left(\overline{\delta}, \varepsilon\right) \mapsto \left.\frac{\sum_{i=1}^{n}\tilde{\delta}_i}{\left(\left\|\tilde{\delta}\right\|\right)^2} \;\right|\; \tilde{\delta} = \left\{\delta_i \in \overline{\delta} \;\middle|\; \delta_i \leq \varepsilon\right\}$$

Linear Combination: The last method combines distances according to user-provided weights. It is called *linear combination* in [VC99], *weighted score-based late fusion* in [DM10; Ros18] or *weighted average* in [BMS+01]. Given user-provided weights $\overline{w} = (w_1, w_2, \ldots, w_n)$ per distance ($\|\overline{w}\| = \left\|\overline{\delta}\right\|$), the weighted average is then defined as follows:

$$\hat{\delta}_{lc}\left(\overline{\delta}, \overline{w}\right) \mapsto \frac{\sum_{i=1}^{n} w_i \cdot \delta_i}{\sum_{i=1}^{n} w_i}$$

[1] This is slightly adopted from [SF94], where they define "CombANZ" to be the average of non-zero similarity values because we are combining distances and not similarity values. It is reasonable to assume that distances above a threshold ε would have a similarity value of zero.

[2] This is also adopted from "CombMNZ" in [SF94] as we are comparing distances and not similarities. In the original case, the sum of similarities is multiplied by the number of nonzero similarities. In our case, we also want to reward items retrieved by multiple methods and therefore additionally divide again by the number of nonzero similarities.

We additionally discuss different perspectives on fusion in Literature Discussion 3.4.

Literature Discussion 3.4 Fusion

Early fusion is commonly referred to as fusing modalities in the feature space while late fusion aggregates those features (which are often unimodal) in order to achieve better retrieval or classification performance [SWS05; AHES+10; PG17] for example by learning weights [TFG+09; LLC+15].
Another application is machine learning in the combination of audio and visual modalities [KLP13; EML+18; NYA+21] or to train text-image embeddings [LXY+19; RKH+21] which would be considered early fusion, as the feature representation is learned considering multiple modalities.
Connections can also be drawn to "unsupervised fusion and ensembles of classifiers in supervised learning [Die00; PP07; ZM12]" [KC18].
Starting with [SF94], there is a significant body of work for new models and evaluations of different fusion and query evaluation schemes for complex queries [Fag99; FLN01; BMS+01; WC02; FKS03; FKM+06; WCB06; BCO+07].

In this chapter, we have reviewed the fundamentals of multimodal multimedia retrieval, going from the query formulation step to different retrieval models, a holistic perspective on retrieval systems, different modalities for queries and fundamentals of more complex queries. This leads us to our own data, query and retrieval model which enable different abstraction levels for queries, relevance and presentation in Chapter 4, and will also serve as a basis for Chapter 5, where we present vitrivr, a multimedia retrieval system which serves as a proof-of-concept for our presented model and is used for the evaluation in Chapter 6.

But you wouldn't clap yet. Because making something disappear isn't enough — you have to bring it back.

— The Prestige

4

Temporal Multimodal Multimedia Retrieval

Following the foundations of multimedia retrieval, we will turn to the conceptual model for temporal multimodal multimedia retrieval, which is at the core of this thesis. Although our motivating scenario and evaluation is focused on visual data with a temporal progression (video and visual lifelog data), both the content of this chapter and the implementation in vitrivr is not specific to visual data and also applicable to other domains such as audio. This is why we talk about *multimedia* in this chapter even though the title of this thesis is focused on video and lifelog data. The examples will be linked to the scenarios from Chapter 2, and the notation and concepts build upon Chapter 3.

This chapter partially builds upon previous work on the conceptual foundations of multimodal multimedia retrieval in the vitrivr system [Gia18; Ros18], and contains content from peer-reviewed journal, conference, and workshop publications which were (co-)authored [RGH+19; SPG+20; HSS+20; HPP+20; HPG+20; HGI+21; HAG+22; HSS23], which are directly referenced where appropriate. Literature discussion blocks often go beyond the content of the publications referenced based on correspondence with the original authors to ensure a correct description and mapping to our model.

We start with our data model in Section 4.1, and then go to the information needs which are within the scope of our model by showing our query model in Section 4.2. Afterwards, we follow the title in reverse order by first discussing complex multimodal queries, which consider different input modalities (e.g., sketch, text) and combinations thereof, in Section 4.3. The chapter ends with our execution model for temporal queries, which enable users to formulate temporal context and constraints for queries in Section 4.4.

4.1 Data Model

The overarching goal of the conceptual data model is to be generic enough such that it supports different kinds of media, such as video, images, audio and 3D models. This is not only important from a theoretical perspective, but also for the implementation work described in the next chapter, where we describe our work on vitrivr, a multimedia retrieval system which supports all those media types.

In our data model, we start with a multimedia *object* $o \in O$ which is divided into *segments* $s \in S$ by a *segmentation function* $\text{SEG} : O \rightarrow 2^S$. For notation purposes, we write the power set, that is the set of all subsets, of a set S as 2^S in this thesis.

Different segmentation functions are imaginable, in the context of our work we consider segmentation functions which map an object onto a linear space with temporal progression from one segment to the next[1]. Other segmentation functions such as partitioning a 3D model into semantically meaningful components, or multiple segmentation functions per object are imaginable, but not supported in the system presented in this thesis as discussed in Chapter 2.

In this section, we first introduce relevant definitions for our multimedia data model and then briefly discuss metadata. This serves as the basis for the sections in which our retrieval model is defined.

Multimedia Data

We define objects and segments in Definitions 4.1 and 4.2, a scored segment in Definition 4.3, segmentation functions which map an object to segments in Definition 4.4, and then *retrieval features* used for retrieval in Definition 4.5.

Definition 4.1 Object

A multimedia object $o \in O := \langle \texttt{oid}, \texttt{type}, \texttt{path} \rangle$ has an identifier `oid`, is of a `type`, and tracks a reference `path` to the actual location where its content is stored.

Using `path` instead of the actual data enables theoretical objects where the entire content is contained in the segments. In such cases, content-resolution

[1]This does not necessarily prevent multimedia objects which do not have such a progression from being segmented. It just means that retrieval functionality which depends on this progression will not work. This is fine, as there is not really a sensible information need based on temporal progression for objects which do not have one.

by segment identifier would need to be possible. On an implementation level, objects may hold type-specific information such as a frame rate for videos using horizontal or vertical partitioning.

Definition 4.2 Segment

A segment $s \in \mathcal{S} := \langle \texttt{sid}, \texttt{oid}, \texttt{seq}, \texttt{start}, \texttt{end} \rangle$ has an identifier `sid`, a reference to its object `oid`, the index which tracks temporal progression within the object `seq` and a `start` and `end`.

As objects are segmented with a temporal progression, `seq` references the index of a segment as determined by a segmentation function. `start` and `end` can be frame numbers for videos, or timestamps for lifelogging. In our formalization, a segment cannot be a part of multiple objects as `seq` is always relative to a specific object segmentation and `start` and `end` are often relative to a specific object. For retrieval, we assign relevance scores $\tau \in [0,1]$ to elements, with 1 indicating perfect relevance. For clarity, we define a *scored segment* in Definition 4.3.

Definition 4.3 Scored Segment

A scored segment $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$ consists of a segment s and a corresponding relevance score $\tau \in [0,1]$.

Definition 4.4 Segmentation Function

A segmentation function $\text{SEG} : \mathcal{O} \rightarrow 2^{\mathcal{S}}$ takes as input an object $o \in \mathcal{O}$ and returns a list of segments for said object. The sequence number of a generated segment is equivalent to its index in the list, thus indicating its temporal index within the specified segmentation.

We provide an example for a segmentation function in Example 4.1.

Example 4.1 Segmentation for Video Retrieval

In video retrieval, one sensible segmentation approach is *shot* segmentation. A shot is a sequence of frames "from a single camera made without interruption" [Skl93]. Thus, a conversation between two people with a back-and-forth in perspectives will result in several shots as this is a visual and not

semantic unit of segmentation. The semantic unit of segmentation would be a scene [HA99].
Specifically, the approach used in vitrivr is based on [KGU10] and compares color histograms of succeeding frames to detect shot boundaries [Ros18].

We show an example of how the data model could be used in a lifelog retrieval context for lifeloggers (that is, people who record their lives as introduced in Section 2.2) in Example 4.2.

Example 4.2 Data Model for Lifelogging Wearables [HSS23]

For wearables which are configured to take pictures at specified intervals, the data used for retrieval is individual images with an associated capture timestamp. These can be used as the atomic unit of retrieval, and are thus considered as the segments in that scenario [RGH+19].

In this example, we group them by day, and thus each day is an object which has as its segments, the images taken on said day.
Assuming all images from the 23rd of September 2016 are stored in the same directory, the object for the 23rd of September would thus be $\langle 160923, \text{DAY}, \text{lifelog}/160923 \rangle$ and the 1224th image from that day taken at 20:06 would be $\langle 160923\_1224, 160923, 1224, 120600, 120601 \rangle$.

Similarly to what we defined in the previous chapter based on [BR11; Gia18], a retrieval feature $\mathscr{f}$ is responsible for mapping both query and segment into a common space where relevance scores are calculated. Our atomic query unit is called query term $\mathtt{qt}$.

Definition 4.5 Retrieval Feature

A retrieval feature $\mathscr{f} := \langle \mathscr{f}_{\mathtt{qt}}, \mathscr{f}_s, \mathscr{f}_r \rangle$ is defined by three functions: a query transformation function $\mathscr{f}_{\mathtt{qt}} : QT \to \mathcal{F}$, a segment transformation function $\mathscr{f}_s : \mathcal{S} \to \mathcal{F}$, and a retrieval function $\mathscr{f}_r : QT \to 2^{\hat{\mathcal{S}}}$.

Example 4.3 Text Embedding Retrieval Feature for Video Retrieval

The filmmaker from our motivating example could search for the lion using the text "Lion in a desert". vitrivr implements a co-embedding retrieval feature [SGH+22], whose inner workings are shown in Figure 4.1. Both the text input and the frames from the shot get embedded in the co-embedding space, where the distance between the two vectors gets computed.

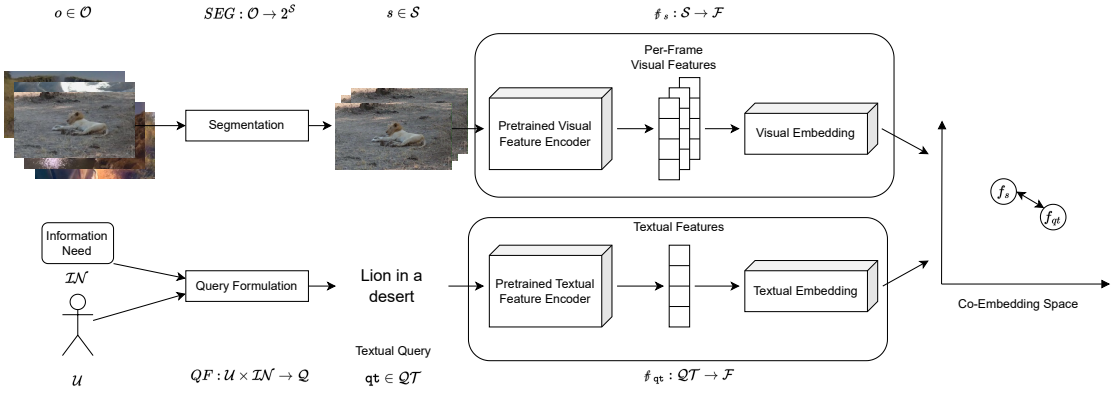


Figure 4.1 Illustration of a retrieval feature with text embedding. Adopted from [SGH+22]

In this case, we use a linear correspondence function to transform the vector distance to a relevance score.

The first two functions have as their output a feature f, $\mathscr{f}_{\mathtt{qt}}$ transforms a query term $\mathtt{qt}$ and $\mathscr{f}_s$ a segment s. Those features are used to evaluate relevance through $\mathrm{REL}(f_{\mathtt{qt}}, f_s)$, where 0 indicates no relevance and 1 indicates perfect relevance.

For notation purposes and conceptual clarity, this is summarized in $\mathscr{f}_r$,

where a single scored list $r_{\mathscr{f}} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n) \in 2^{\hat{\mathcal{S}}}$ is generated given a query term `qt`. Each individual element of this list is a scored segment $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$, whose relevance score is determined by evaluating the relevance of the segment $s \in \mathcal{S}$ for the given query term. For an individual segment s, $\hat{s} = \left\langle s, \mathrm{REL}(\mathscr{f}_{\mathtt{qt}}(\mathtt{qt}), \mathscr{f}_s(s)) \right\rangle$.

Metadata

The metadata model looks the same for objects and segments and was already described in [HSS23]. Literature often views all extracted features as metadata, and based on this we can differentiate metadata between two types of metadata in our model from a conceptual perspective: metadata that is used for Boolean retrieval, and metadata that is used for content-based retrieval such as OCR data. From an implementation perspective, we can separate two different types: metadata that can be extracted from the given multimedia data such as frame rate, and externally provided metadata such as archival annotations in a cultural heritage context.

Metadata for Boolean Retrieval: To support information needs which contain a component where binary relevance evaluation is possible, we use simple *metadata tuples* as defined in Definition 4.6.

Definition 4.6 Metadata Tuple

Given the identifier `id` of either a segment or an object, a metadata tuple `m` is defined as follows:

$$\mathtt{m} := \langle \mathtt{mid}, \mathtt{id}, \mathtt{domain}, \mathtt{key}, \mathtt{value} \rangle$$

with `mid` being a unique identifier per metadata tuple.

Domains might include *exif*, *technical* for aspect ratio or resolution, or *provided* to indicate external sources. As we assign an artificial primary key, multiple metadata tuples per object or segment are possible.

Metadata for Content-Based Retrieval: Multimedia data in real-world scenarios often comes with provided information from third-party sources such as textual descriptions. In these cases, the retrieval features defined in Definition 4.5 can shed the need for a segment transformation function $\mathscr{f}_s$ and instead only

provide a query transformation function $f_{\texttt{qt}}$. Otherwise, this kind of metadata can be considered identical to extracted features.

4.2 Query Model

In this section, we build the query model incrementally by starting with our atomic query unit, define complex similarity queries with specific examples, and then define temporal similarity queries.

4.2.1 Query Term

At the core of the query model lies a *query term* $\texttt{qt} \in QT := \left\langle \texttt{data}, \overline{f} \right\rangle$, which contains information about a single modality, and the retrieval features used. We define qt in Definition 4.7 and provide an example in Example 4.4.

Definition 4.7 Query Term

A query term $\texttt{qt} \in QT := \left\langle \texttt{data}, \overline{f} \right\rangle$ is a representation of a user's information need for a specific modality (e.g., free text, Boolean predicates, visual sketch). The actual content depends on the modality and is captured in data, and $\overline{f}$ contains a list of retrieval features that the user deems sensible for the content of the query term.

Example 4.4 Text Query Term

Marion is looking for images that contain a tree next to a river. They therefore formulate a query with a single modality: the text "tree next to a river". Internally, the retrieval system has two kinds of retrieval features that work with text: a text-embedding retrieval feature such as W2V++ [LXY+19], and a traditional text retrieval feature that searches in textual descriptions generated by a commercial API using Lucene[a]. Assuming the retrieval features are called *embedding* and *description* and default parameters are used, $\texttt{qt}_t = \langle$"tree next to a river", (embedding, description)$\rangle$

[a] https://lucene.apache.org

A possible extension would be to allow users to specify explicitly how the results from the features are combined, but we assume that if multiple features are evaluated for the same query term, a simple linear combination should suf-

fice. Otherwise, a user could simply evaluate the features in two different query terms and specify a complex similarity query, which we will discuss in the next subsection.

Configurable parameters such as the number of results to be returned k, or which correspondence function to be used are included in the parameter `data` in our model.

Building upon that, we extend our retrieval model in the next section with more complex queries.

4.2.2 Complex Similarity Queries

Having defined our atomic query unit, we now turn to more complex information needs that still target a single segment. Our query model relies upon fundamental work in the domain of complex queries [Fag99; BMS+01][2]. Specifying the combinations in advance allows for improved query planning and execution models for the retrieval engine and database.

Additionally, as the retrieval features we use in our model are tasked with generating ranked lists with a relevance score, we use a slightly adapted version of DCF [BMS+01] which we call Similarity-Combining Function (SCF)[3]. We define SCF in Definition 4.8, build our definition for *complex similarity queries* on that in Definition 4.9 and provide an example in Example 4.6.

Definition 4.8 Similarity-Combining Function

A Similarity-Combining Function (SCF) $\hat{\varrho} : ([0,1])^n \rightarrow [0,1]$ calculates a single relevance score $\tau \in [0,1]$ from n different relevance scores. The individual scores to be combined by a SCF are $\overline{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$, with $\tau_i \in \overline{\tau}$.
$\hat{\varrho}$ is monotonic for all arguments: $\hat{\varrho}\Big(\tau_1, \tau_2, \ldots, \tau_n\Big) \leq \hat{\varrho}\Big(\tau'_1, \tau'_2, \ldots, \tau'_n\Big)$ if $\tau_i \leq \tau'_i$ for all i.

$\hat{\varrho}$ can be chained and nested. Consider Example 4.6, where the final $\hat{\varrho}$ combines a min operator with a linear combination: $\hat{\varrho}_e = \hat{\varrho}_{min}\left(\hat{\varrho}_{lc}(\tau_t, \tau_p), \tau_d\right)$.

[2]Which has also informed some work on predecessors of the vitrivr system [Spr14; Gia18].

[3]The space of potential SCF is large. We have covered different fusion functions in the previous chapter, with some adopted from foundational work on combining similarities [SF94]. The implementation chapter will cover the implemented functionality in vitrivr, which slightly deviates from the model for engineering reasons.

Definition 4.9 Complex Similarity Query

Given a list of individual query terms $\overline{\mathtt{qt}}$, we define a complex similarity query $\mathtt{csq} := \left\langle \overline{\mathtt{qt}}, \hat{\varrho} \right\rangle$ with $\left\| \overline{\mathtt{qt}} \right\| = n$, and $\hat{\varrho} : ([0,1])^n \rightarrow [0,1]$ as defined above describing the desired similarity-combining functions to merge the different result sets from individual query terms.

In this definition, the query terms are comparable to "atomic queries" in [Fag99] and "reference objects" in [BMS+01].

Having defined complex similarity queries, we will now provide two examples. The first is a simple weighted multimodal similarity query in the context of video retrieval in Example 4.5 and the second is a more complex similarity query in the context of lifelog retrieval Example 4.6.

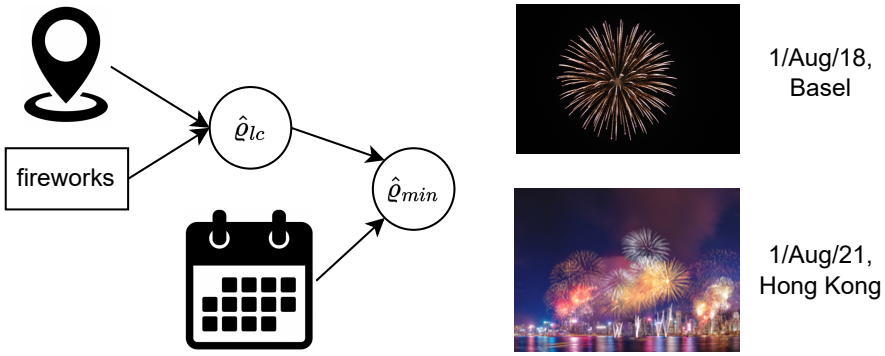
In the first example, we consider a weighted multimodal similarity query with a list of query terms to considered $\overline{\mathtt{qt}} = (\mathtt{qt}_1, \mathtt{qt}_2, \ldots, \mathtt{qt}_n)$, relevance scores are combined using linear combination $\hat{\varrho}_{lc} : [0,1]^n \times [0,1]^n$ and thus the user provides additionally weights for each query term $\overline{w} = (w_1, w_2, \ldots, w_n)$, $\|\overline{w}\| = \left\|\overline{\mathtt{qt}}\right\|$.

Example 4.5 Weighted Multimodal Similarity Query

Andrea is looking for pictures of the time they visited their red vacation house, which sits on top of a hill. They therefore formulate a query with two modalities: a textual one "A red house on top of a hill, surrounded by trees." and a sketch sk serialized to Base64-representation sk_{b64} with equal importance and thus $\overline{w}$=(1,1). Assuming there are multiple sketch-features such as *edge* and *localcolor*, $\mathtt{qt}_t = \left\langle \text{"A red house on top of a hill, surrounded by trees."}, (\text{embedding}) \right\rangle$ and $\mathtt{qt}_s = \left\langle sk_{b64}, (\text{edge}, \text{localcolor}) \right\rangle$ the full query is $\mathtt{csq} = \langle (\mathtt{qt}_t, \mathtt{qt}_s), \hat{\varrho}_{lc}(\overline{\tau}, (1,1)) \rangle)$ meaning the results are fused using linear combination as discussed in the previous chapter.

The example shows how the implementation model used in [RGT+16; Gas17], which introduced first iterations of the vitrivr system, can be easily mapped onto our retrieval model.

Example 4.6 Complex Similarity Query - Lifelog



A lifelog query looking for images from the Swiss national day where fireworks are visible would result in two queries: *date=01/08* and a content-based query for "fireworks". Thus $\mathtt{qt}_b = \langle \text{date=01/08}, (\text{metadata}) \rangle$, $\mathtt{qt}_t = \langle \text{"fireworks"}, (\text{embedding}) \rangle$. Since the date is a hard constraint, $\hat{\varrho} = \min(\tau_1, \tau_2)$ as the query for the date will return binary relevance scores.
If we are additionally most interested in images in proximity of the "Mittlere Brücke" and would add a proximity query $\mathtt{qt}_p = \langle [47.56, 7.59], (\text{map}) \rangle$, the SCF becomes slightly more complicated. In this example, we go for a linear combination of the two content-based features and keep the date as a hard constraint and thus the final $\hat{\varrho}$ is as follows:
$\hat{\varrho} = \min(\tau_1, \ \hat{\varrho}_{lc}(\tau_2, \tau_3))$.
In this case, an image of fireworks from a trip to Hong Kong over the 1st of August would still be retrieved, albeit with a lower score.

4.2.3 Temporal Similarity Queries

Having defined a query model that is sufficient to address complex single-segment information needs, we now introduce our query model for information needs with temporal components such as those defined in our motivating scenarios in Chapter 2. Definition 4.10 defines a *temporal similarity query* `tsq`.

Definition 4.10 Temporal Similarity Query

A temporal similarity query $\texttt{tsq} := \langle \overline{\texttt{csq}}, \overline{\phi}, \omega, \overline{\texttt{qt}_{\texttt{tsq}}}, \hat{\varrho}_{\texttt{tsq}} \rangle$, is defined as a list of subqueries $\overline{\texttt{csq}}$ specifying individual components, an optional list of user-specified distances between those components $\overline{\phi}$ ($\left\|\overline{\phi}\right\| = \|\overline{\texttt{csq}}\| - 1$), an optional maximum length of a result ω and an optional list of query-level constraints $\overline{\texttt{qt}_{\texttt{tsq}}} = \left(\texttt{qt}_{\texttt{tsq}_1}, \texttt{qt}_{\texttt{tsq}_2}, \ldots, \texttt{qt}_{\texttt{tsq}_q}\right)$ together with a SCF $\hat{\varrho} : [0,1]^{q+1} \rightarrow [0,1]$, which specifies how the query-level constraints are to be merged with the results of the subqueries.

ϕ can be either a real number or a semantic expression of a temporal distance such as "immediately afterwards" or "shortly after" which can be used by the execution engine and interpreted differently based on the scenario. We do not restrict the semantic space of user-specified distances ϕ,ω as it is scenario-dependent. The query execution section will discuss how a subset of these are used in the individual fusion algorithms.

It is important to note that although this model cleanly builds upon our model for complex similarity queries, other retrieval models for subqueries could also be used as long as a list of scored segments is produced per subquery (i.e., $r_{\texttt{csq}} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$). This flexibility is important for other approaches from literature which might have different approaches for queries targeting a single segment but still wish to allow users to specify temporal context.

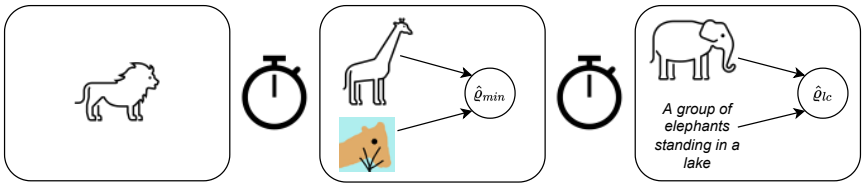
One design alternative to the query model is that we could allow specifying the desired duration of an individual component. Our model allows this indirectly through the distances between the subqueries, but the semantics are slightly different.

While the model is built with the assumption that the subqueries follow each other in a linear sequence, it can also be used if the ordering does not matter. Specifying *does not matter* as a semantic distance between subqueries can be used to provide the information that algorithms not requiring a strictly matching temporal order can be used or are preferred. While this might not be the most common scenario, it might have its uses especially for memory aid, where the exact order of a sequence in the past is not remembered anymore.

We give two examples for temporal similarity queries, one for video retrieval in Example 4.7 and one for lifelog retrieval in Example 4.8.

Example 4.7 Temporal Similarity Query for Video Retrieval

Considering the example discussed in Chapter 2 and the availability of a concept-detector, sketch features and a textual embedding, a query might look as follows:

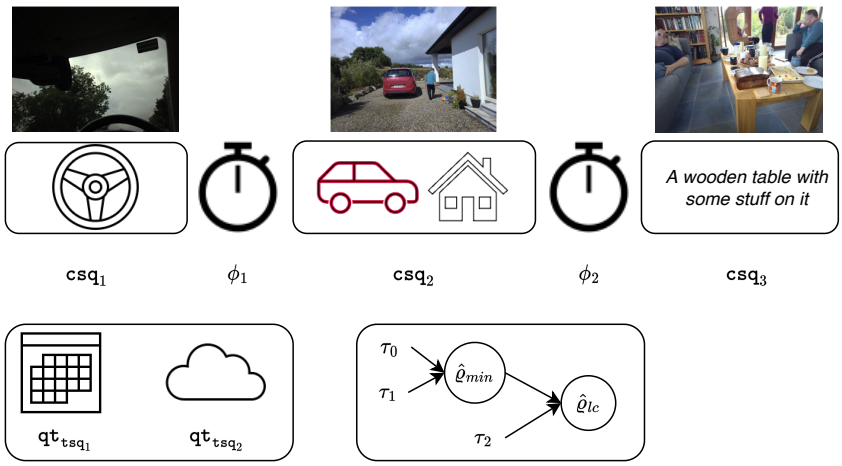


Here, the three subqueries include a concept-based query for a lion, a combination of sketch and concept-based query for the giraffe and a combination of concept search and textual embedding for the elephants. Whereas for the sketch, a minimal combination makes sense as we are specifically interested for sketch matches where a giraffe was detected, for the elephants, a linear combination is selected. Two distances specifying ϕ_1, ϕ_2 = "shortly afterwards" are also provided.
While we do not provide query-level constraints in this example, simple examples might include that a song was playing through all sequences or technical details such as aspect ratio.
Formally:
$\mathtt{csq}_1 = \langle(\langle\text{"lion"}, (\text{tag})\rangle), ()\rangle$
$\mathtt{csq}_2 = \langle(\langle\text{"giraffe"}, (\text{tag})\rangle, \langle sk_g, (\text{sketch})\rangle), \hat{\varrho}_{min}\rangle$
$\mathtt{csq}_3 = \langle(\langle\text{"elephant"}, (\text{tag})\rangle, \langle\text{"A group [...]"}, (\text{embedding})\rangle), \hat{\varrho}_{lc}\rangle$

and $\mathtt{tsq} = \langle(\mathtt{csq}_1, \mathtt{csq}_2, \mathtt{csq}_3), (\phi_1, \phi_2), -, -, -\rangle$

Example 4.8 Temporal Similarity Query for Lifelog Retrieval

This example is based on a task from LSC 2019[a] [GSH+19]. A significant amount of information is available. The lifelogger was driving for an hour until arriving at a friends house, where a red car was parked. Afterwards, the lifelogger spent time inside with their friends around a wooden table. It was a cloudy day on a weekend.



Putting all this together, we get a more complex query than in the previous example, where the individual query terms specify driving, a red car and a house, and then a description of the inside of the house. Additionally, we have query-level constraints such as the information that it was cloudy and the possible days of the week. The combination function requires a match for the date using $\hat{\varrho}_{min}$ but allows misses for the cloud constraint as there might not be clouds visible inside the house using $\hat{\varrho}_{lc}$. Thus, given τ_0 as the score of a result before the query-level constraints, $\hat{\varrho}_{\texttt{tsq}} = \hat{\varrho}_{lc}\,(\tau_2, \hat{\varrho}_{min}\,(\tau_0, \tau_1))$

[a] Specifically, Task 26.

4.3 Query Execution for Multimodal Queries

Having defined our data and query model, we will first discuss our execution model for complex multimodal queries in this section before moving to the execution model for temporal queries in the next section.

4.3.1 Retrieval Features

The task of an individual retrieval feature is to generate a scored list of the relevant elements with respect to a given query. There are different types of similarity queries imaginable, such as returning the k most similar items, all elements above a given relevance threshold ε or even the k most dissimilar items. Additionally, retrieval features might speed up retrieval using index structures relevant to the feature representation.

As previously introduced, a retrieval feature $\mathscr{f}$ is tasked with generating a scored list $r_{\mathscr{f}} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$, with the individual element being a scored seg-

ment $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$ when provided with a query term `qt`. The retrieval function $\ell_r : QT \to 2^{\hat{\mathcal{S}}}$ could also be expressed in pseudo-sql as shown in Listing 4.1, where `feature_transform` indicates transforming the query term as previously defined using $\ell_{\texttt{qt}}$ and `score` indicates evaluating per-row relevance of a segment.

```
SELECT id,
        score(
                feature_transform(query),
                segment_features
        ) AS score
FROM feature_table
LIMIT 1000
ORDER BY score DESC
```

Listing 4.1: Pseudo-SQL for retrieval on a single retrieval feature, retrieving the top 1000 relevant results. Taken from [HSS+20]

This is sufficient for the purposes of our model. We have shown different retrieval models and features in the previous chapter, and will give a brief overview over the ones implemented in vitrivr in Chapter 5.

4.3.2 Complex Multimodal Queries

As argued previously, the space for potential SCF that could be specified is large, and we will focus in this section on selected SCF that are useful in practice.

SCF can generally be evaluated by the application layer (in our case, the retrieval engine) after the individual query terms have been executed independently. In most cases, there are performance gains to be had by pushing them down to the database layer.

We will first show SCF where late fusion is easily possible, and then afterwards discuss examples where it can lead to significant performance gains to have a different execution model.

The first group of SCF all operate under the assumption that the individual query terms have been executed independently and in parallel[4].

This means that when evaluating the SCF $\hat{\varrho} : ([0,1])^n \to [0,1]$, we have the intermediate results $r_{\texttt{sqi}} = \left(r_{\texttt{qt}_1}, r_{\texttt{qt}_2}, \ldots, r_{\texttt{qt}_u}\right)$ available similar to the previous section and therefore for notation purposes in this subsection, $\hat{\varrho} : ([0,1])^u \to [0,1]$.

[4]This allows systems to show preliminary results early, which has interesting implications for the database layer [Gia18].

This allows us to support different SCF which are useful in different contexts. We will list some of them here, based on the DCF discussed in the previous chapter [SF94; BMS+01]. Preliminary versions of this work were published in [HSS+20; HPP+20], and negative relevance feedback has benefited from work supervised during this dissertation project [Pas20].

Linear Combination: $\hat{\varrho}_{lc}$ requires a vector of weights $\overline{w}_{lc}, \|\overline{w}_{lc}\| = u$ as an additional argument and is then defined as:

$$\hat{\varrho}_{lc}\left(\overline{\tau}, \overline{w}_{lc}\right) \mapsto \frac{\sum_{i=1}^{u} \overline{w}_{lc_i} \cdot \tau_i}{\sum_{i=1}^{u} \overline{w}_{lc_i}}$$

For context, this is the approach that was used in earlier versions of vitrivr [RGT+16; Ros18], and remains a simple, yet effective configuration of the user interaction today. We provide an example in Example 4.9.

Example 4.9 Linear Combination for Multimodal Queries

Alex is searching for multimedia objects containing mountains. Therefore, they formulate a query containing two modalities: a sketch of a mountain and a concept search for *mountain*. Both modalities are equally important to them. This is internally translated to two query terms, a base64-encoded sketch and a query term which simply has the tag *mountain*.
Thus, $\overline{\mathtt{qt}} = (\mathtt{qt}_{sketch}, \mathtt{qt}_{tag})$, $\overline{w} = (1,1)$, $\hat{\varrho} = \hat{\varrho}_{lc}(\overline{w})$, and $\mathtt{csq} := \left\langle \overline{\mathtt{qt}}, \hat{\varrho} \right\rangle$. After query execution, this results in two scored segments $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$ per term. $r_{\mathtt{qt}_{sketch}} = (\langle s_1, 0.8\rangle, \langle s_3, 0.4\rangle)$, $r_{\mathtt{qt}_{tag}} = (\langle s_1, 1\rangle, \langle s_2, 1\rangle)$, and $r_{\mathtt{sqi}} = (r_{\mathtt{qt}_{sketch}}, r_{\mathtt{qt}_{tag}})$.
Applying linear combination results in the following score per segment:

$$\hat{s}_1.\tau = \hat{\varrho}_{lc}(\overline{\tau}_{s_1}, \overline{w}_{sq}) = \frac{1 \cdot 0.8 + 1 \cdot 1}{2} = 0.9$$
$$\hat{s}_2.\tau = \hat{\varrho}_{lc}(\overline{\tau}_{s_2}, \overline{w}_{sq}) = \frac{1 \cdot 0 + 1 \cdot 1}{2} = 0.5$$
$$\hat{s}_3.\tau = \hat{\varrho}_{lc}(\overline{\tau}_{s_3}, \overline{w}_{sq}) = \frac{1 \cdot 0.4 + 1 \cdot 0}{2} = 0.2$$

and so, $r_{\mathtt{csq}} = (\hat{s}_1, \hat{s}_2, \hat{s}_3) = (\langle s_1, 0.9\rangle, \langle s_2, 0.5\rangle, \langle s_3, 0.2\rangle)$.
Implementations may choose to pass intermediate results $r_{\mathtt{sqi}}$ to the frontend so Alex could change the weighting of their modalities without executing another query.

min: Sensible especially for the combination of Boolean relevance scores as in a lifelog scenario where metadata is plentiful, $\hat{\varrho}_{min}$ requires closeness to all query terms, formally:

$$\hat{\varrho}_{min}(\overline{\tau}) \mapsto \min_{i=1}^{u}(\tau_i)$$

This is basically the AND operator in fuzzy logic. As an example, consider the combination of a query term which looks for a specific date and a textual description of an image.

max: Taking the maximum relevance score of a single query term can be sensible, but it requires the assumption that the relevance scoring function of the used retrieval features are closely aligned. This means that a relevance score of 0.8 for a sketch-feature indicates the same relevance as a relevance score of 0.8 for an OCR retrieval feature. However, if two retrieval features use the same underlying method (e.g., when combining two different OCR methods which both use Lucene), it can be sensible. This is basically the OR operator in fuzzy logic.

$$\hat{\varrho}_{max}(\overline{\tau}) \mapsto \max_{i=1}^{u}(\tau_i)$$

Absolute Negative Relevance Feedback: In case a user wishes to use content-based late filtering (e.g., by removing all video segments that contain wedding imagery[5]), this can also be done in a late filtering SCF[6]. This is sensible when there is a threshold applied to the retrieval features that return undesirable segments, either by limiting the number of segments (knn) or a fixed relevance threshold ($\varepsilon nn; \varepsilon \in [0,1]$). Formally, assuming an arbitrary number of query terms which describe undesirable content, their indexes m with $\forall m_i \in [1,u]$, and a SCF $\hat{\varrho}_r : [0,1]^{u-\|m\|} \rightarrow [0,1]$ which combines the remaining similarities:

$$\hat{\varrho}_{nr}(\overline{\tau}, m, \hat{\varrho}_r) \mapsto \begin{cases} 0 & \sum_{i=1}^{\|m\|} \tau_{(m_i)} > 0 \\ \hat{\varrho}_r(\{\tau_i | i \notin m\}) & \sum_{i=1}^{\|m\|} \tau_{(m_i)} = 0 \end{cases}$$

This filters out all segments that match the negative query terms. We provide an example in Example 4.10. Alternatively, one might consider having

[5]For example, multiple participants in user studies over the years have noted the proportionally large amount of wedding footage in V3C1 [BRS+19].

[6]This is the implementation methodology used in [HPP+20; Pas20].

an extended variant which explicitly specifies how the negative relevance feedback scores should be combined.

Example 4.10 Negative Relevance Feedback in Video Retrieval

Ollie is searching for videos containing people in green dresses, but knows their collection contains a significant amount of wedding imagery, and they are specifically looking for non-wedding examples. Therefore, they formulate a query containing two content-based text terms: $\texttt{qt}_1 = \langle$"people in green dresses", embedding$\rangle$, $\texttt{qt}_2 = \langle$"people at a wedding", embedding$\rangle$. The indexes of the query terms describing undesirable content are $m = \{2\}$. As there is only one query term specifying relevant content, $\hat{\varrho}_r$ is irrelevant and we use $\hat{\varrho}_{max}$. The query specified is $\texttt{csq} = \langle(\texttt{qt}_1, \texttt{qt}_2), \hat{\varrho}_{nr}(2, \hat{\varrho}_{max})\rangle$.
For the first query term we get two scored segments, $\hat{s}_1 = \langle s_1, 0.8\rangle, \hat{s}_2 = \langle s_2, 0.5\rangle$ and for the second query term $\hat{s}_3 = \langle s_1, 0.3\rangle$. Accordingly, we have two set of relevance scores, $\overline{\tau}_1 = (0.8, 0.3), \overline{\tau}_2 = (0.5, 0)$.
This results in the following scores per segment:

$$\hat{\varrho}_{nr}(\overline{\tau}_1, m, \hat{\varrho}_r) = 0 \qquad \sum_{i=1}^{1} \overline{\tau}_{1_{(m_i)}} = \overline{\tau}_{1_2} = 0.3$$

$$\hat{\varrho}_{nr}(\overline{\tau}_2, m, \hat{\varrho}_r) \mapsto \hat{\varrho}_{max}(\overline{\tau}_{2_1}) = 0.5 \qquad \sum_{i=1}^{1} \overline{\tau}_{(m_i)} = \overline{\tau}_{2_2} = 0$$

and thus, $r_{\texttt{csq}} = (\hat{s}_1, \hat{s}_2) = (\langle s_1, 0\rangle, \langle s_2, 0.5\rangle)$. This is in line with Ollie's expectations as the first segment was a content match for the wedding query term.

For the next operator, it makes sense to involve the database layer more closely due to their semantics.

Staged Queries: Considering Example 4.9, a more sensible approach might be to evaluate similarity to the sketch on all segments that have been classified as mountains, instead of executing the queries independently. Similarly, for the $\hat{\varrho}_{min}$ operator introduced previously, a common use case for lifelog retrieval is to have a content-based query with a Boolean filter.

In both cases, if a user is interested in the k most similar results, execution order matters. Searching for the k most relevant segments for a query term

and filtering for those which match a Boolean filter is almost guaranteed to return less than k items. On the other hand, Boolean filters often would return more than k items, and ranking those by similarity is not necessarily sensible.

To guarantee being able to return k items in late fusion, the database layer would have to return the relevance score for all items in a collection, which makes the usage of index structures unfeasible and has significant performance drawbacks.

Thus, we introduce the $\hat{\varrho}_{k1}$ operator, which guarantees that k elements are returned that are relevant for the filtering relevance score[7]. The semantics of the operator are not the same as for $\hat{\varrho}_{min}$, as it only operates on two relevance scores[8]. The combination of the two resulting similarity scores can be specified with $\hat{\varrho}_2 : ([0,1])^2 \to [0,1]$

$$\hat{\varrho}_{k1}(\tau_1, \tau_2, \hat{\varrho}_2) \mapsto \begin{cases} 0 & \tau_1 = 0 \\ \hat{\varrho}_2(\tau_1, \tau_2) & \tau_1 > 0 \end{cases}$$

The reason we only require τ_1 to be 0 is that the second part might be a color sketch with a relevance of zero, but the user would still want to see the result. This operator can be chained indefinitely, although the usefulness of this scenario is questionable. A previous version of this operator and its implementation is described in [HSS+20], where we described it using the term *Staged Queries*, where for example a Boolean filter would be stage 0 and the content-based query term stage 1.

For this operator, there are different execution plans imaginable: the whole query could be sent to the database as-is, or we first query all relevant items for the first operator, and then send the relevant ids along with the second similarity query as a filter. We show pseudo-SQL for the first variant in Listing 4.2 and for the second variant in Listing 4.3.

Sophisticated execution models for complex similarity queries have long been a source of research interest [Fag99; BMS+01], however looking at state-of-the-art systems for video and lifelog retrieval [LVM+21; HGB+22; GJS+22] they are rarely implemented in practice. In contrast, the execution model presented in

[7]Only elements with a relevance score $\tau > 0$ are returned, therefore there can be edge cases where fewer than k elements are returned.

[8]Recall from Section 4.2.2 that SCF can be chained, and thus this does not limit the number of query terms.

```
SELECT id,
        score(
                feature_transform(query),
                segment_features
        ) AS score
FROM feature_table
WHERE id IN(
        SELECT id
        FROM other_feature
        LIMIT 1000
        ORDER BY
        score(
                feature_transform(other_query),
                segment_features
        )
        DESC
)
LIMIT 1000
ORDER BY score DESC
```

Listing 4.2: Pseudo-SQL for retrieval on a single retrieval feature, retrieving the top 1000 relevant results limited to ids deemed relevant by a subquery

```
SELECT id,
        score(
                feature_transform(query),
                segment_features
        ) AS score
FROM feature_table
WHERE id IN(id_1,id_2,...id_k)
LIMIT 1000
ORDER BY score DESC
```

Listing 4.3: Pseudo-SQL for retrieval on a single retrieval feature, retrieving the top 1000 relevant results limited to ids obtained by a previous process. Adopted from [HSS+20]

this section cleanly separates database and application layer which significantly simplifies both implementation and model. Similar to what we will argue in the next section however, advances in modeling database support for multimedia retrieval [Gas23] and query planning in multi-model databases [Vog22] open interesting avenues for research that considers tighter integration of the retrieval layers, which we leave for future work.

4.4 Query Execution for Temporal Queries

In this section, we will present our retrieval and execution model for temporal queries. It contains concepts and implementation work partially done in

the context of VBS [SPG+20; HGI+21; HAG+22], LSC [HPG+20], and supervised theses [Gst21; Ben22a]. A preliminary version of this work was published in [HSS+20].

We will begin by defining the problem in Section 4.4.1, then discuss our retrieval model in Section 4.4.2 from which different fusion algorithms are derived in Section 4.4.3.

4.4.1 Problem Definition

Whereas previously, a *segment* was the atomic unit of retrieval and therefore a scored segment the unit in which results were returned $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$, expanding the desired unit requires us to consider segment-spanning *sequences* as our result unit. We define a sequence ς in Definition 4.11 and its scored equivalent, a *scored sequence* $\hat{\varsigma}$ in Definition 4.12.

Definition 4.11 Sequence

A sequence $\varsigma \in \mathfrak{S} := \langle \texttt{oid}, \texttt{start}, \texttt{end} \rangle$ consists of an object identifier `oid`, and a reference to its `start` and `end` inside said object.

Since sequences are constructed on the fly based on segments but may be only a partial segment, or span multiple segments, a sequence only carries about its start and end.

Definition 4.12 Scored Sequence

Similarly to a scored segment $\hat{s} \in \hat{\mathcal{S}} := \langle s, \tau \rangle$, a scored sequence $\hat{\varsigma} \in \hat{\mathfrak{S}} := \langle \varsigma, \tau \rangle$, $\tau \in [0, 1]$ consists of information about a sequence and its score.

The problem a temporal fusion algorithm needs to solve is described in Definition 4.13 and builds on our query model. A brief overview of approaches by other systems is provided in Literature Discussion 4.1.

Definition 4.13 Temporal Fusion Problem

Given a temporal query $\texttt{tsq} := \langle \overline{\texttt{csq}}, \overline{\phi}, \omega, \overline{\texttt{qt}_{\texttt{tsq}}}, \hat{\varrho}_{\texttt{tsq}} \rangle$, a retrieval system is tasked with retrieving relevant *sequences* ς and assigning them a relevance score τ, thus producing a ranked list of scored sequences $r_{\texttt{tsq}} = (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_x)$.

Literature Discussion 4.1 Temporal Search Perspectives of Other Systems

Taking a look at other retrieval systems participating in video retrieval benchmarks, no system offers query-level constraints, with most research prototypes in 2022 offering temporal queries that also follow a late fusion approach similar to ours [HDN+22; ABC+22; AMG+22; KSJ+22; HSJ+22; AGG22; TNN+22]. The execution model differs, as does the level of detail provided in the papers, but all the cited systems first execute the subqueries independently, and then perform a re-ranking to show sequences that match the query specification. The unit of result presentation differs, with some showing segment pairs matching the query sequence [HDN+22; AGG22], others constructing matching sequences with some level of aggregation or enhancement [ABC+22; HSJ+22], and others simply using an entire video as the result unit [KSJ+22].
As an outlier, [LMS+22] directly evaluates the full query. This is among other reasons possible because each subquery is precisely a textual description of a segment, from which a vector is generated and then the generated vectors for each subquery are used in a linear scan over all stored vectors with a sliding window.
In the context of lifelog retrieval, some systems have no temporal query capabilities, but implement a form of lifelog summarization or event clustering [NLN+22; HTN+22; ARG22].

As discussed, there are different execution models imaginable, particularly when it comes to the handling of query-level constraints. We will discuss a few here, and then afterwards reason why we have chosen to investigate a late fusion option in this thesis.

Append Query Terms: One option is to append all query constraints to each subquery, and to extend the SCF of the subqueries by the specified SCF of the temporal query. While this executes each query-level constraint provided in $\overline{\texttt{qt}_{\texttt{tsq}}}$ per subquery, it significantly simplifies the execution model. This option is compatible with both a late fusion of the subquery results, and an approach where the logic is pushed down to the database layer.

As a concrete example, consider the query-level constraint of a weekday in Lifelog Retrieval. Appending this Boolean filter to each subquery is trivially done, but has performance drawbacks as it has to be applied for each subquery.

Execute Constraints Only Once: Executing separate queries for the query-level

constraints avoids repeat execution, but forces a late fusion approach where results are fused only at the end.

In the example of a weekday query-level constraint, this would mean merging the scores of the query-level constraint with the scores of the subqueries.

Push Logic to Database: By either pushing the entire logic down to the database layer or rewriting the query even further, one could take advantage of database planning and caching capabilities. This is however not supported in most current multimedia retrieval databases [Gia18; GRH+20].

We assume in the remainder of this section that whichever approach was chosen, the scores of the query-level constraint have been fused into subquery results and are therefore agnostic to the approach chosen. At its core, the argument we make in our model for temporal queries is for late fusion. Therefore, all algorithms will be presented with results from the individual subqueries and are tasked with generating a relevant and ranked list of sequences.

There is an argument to be made to view the entire space of query possibilities as one query and then treat query execution as a classical database query planning problem. In this approach, the database layer might create more efficient plans or design approaches that exploit that the full query is available. We leave this approach for future work for both conceptual and practical reasons. On the conceptual side, this would require significant additional work in terms of modeling the interplay between query specification, query transformation, the data to be queried and the execution model for complex similarity search on the database side. As argued previously, recent advances in modeling database support for multimedia retrieval [Gas23] and query planning in multi-model databases [Vog22] make this a potentially feasible approach for future work. On the practical side, our late fusion approach significantly simplifies the interaction between application and database layer.

4.4.2 Execution Model

As discussed, we consider in this thesis temporal fusion as a *late fusion* problem and thus assume that the subqueries are executed independently, and no matter which option was chosen, the scores of the query-level constraints have already been considered. Thus, we define our approach to the problem in Definition 4.14.

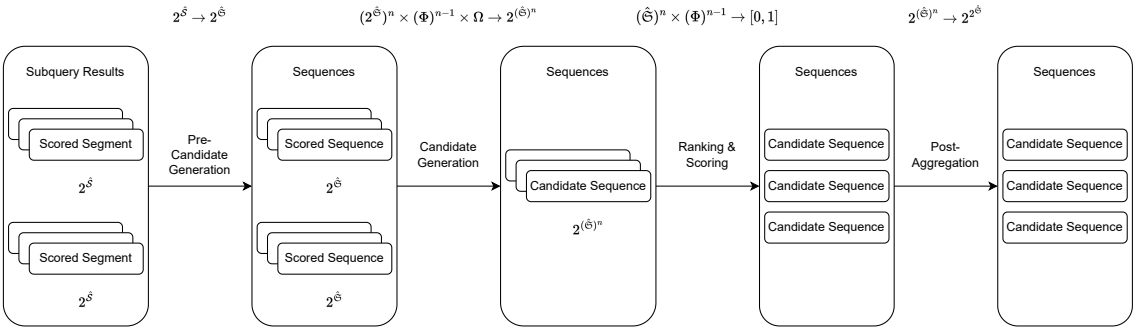


Figure 4.2 Visual overview of late fusion for temporal scoring

Definition 4.14 Temporal Late Fusion

Given a list of results per subquery, $\overline{r_{\texttt{csq}}} = \left(r_{\texttt{csq}_1}, r_{\texttt{csq}_2}, \ldots, r_{\texttt{csq}_n}\right)$ with $n = \|\overline{\texttt{csq}}\|$, an optional list of user-specified distances $\overline{\phi}$ ($\left\|\overline{\phi}\right\| = n - 1$), and an optional maximum distance ω, a temporal fusion algorithm is tasked with generating, scoring and therefore ranking *sequences* ς and assigning them a relevance score τ, thus producing a ranked list of scored sequences $r_{\texttt{tsq}} = (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_x)$.

We identify different stages in our execution model where different approaches are imaginable, and this subsection is structured accordingly. We show a visual representation of this process in Figure 4.2. As is usually the case in retrieval approaches, units of retrieval must be *retrieved* or *generated*, *scored* or *ranked*, and then *presented*. In the figure, this corresponds to pre-candidate and candidate generation, ranking & scoring and then post aggregation.

In a first step, algorithms must choose on which basis they will generate our unit of retrieval, the *sequence* ς. Most algorithms we present do so by mapping the individual result lists to sequences, for example by merging temporally close segments from a single result list. This assumes that the underlying system has over-segmented a video relative to the desired concept. Alternatively, algorithms may also not aggregate or assume under-segmentation and partition elements from the result lists.

Having selected their approach, algorithms must generate *candidate sequences*, that is sequences that are potentially relevant to the given query. These candidate sequences must then be scored and ranked, where there are different approaches, e.g., in the handling of a user-provided distance between two subqueries. Finally, there is a post-aggregation step, where the resulting sequences are transformed into the *unit of presentation*, for example, overlapping candidate sequences in a given object may be merged or removed from consideration.

Pre-Candidate Generation

As the result unit of our model for temporal queries are sequences, at some point we must transform the segments from the subquery results to sequences. This is rather trivial for individual sequences, as a segment s already possesses all relevant attributes to construct a sequence $\varsigma \in \mathfrak{S} := \langle \texttt{oid}, \texttt{start}, \texttt{end} \rangle$.

However, algorithms can choose to aggregate temporally close segments from the same subquery. We define this step in Definition 4.15, show an example of this aggregation in Example 4.11 and then discuss approaches used by other systems in Literature Discussion 4.2. As we assume all subqueries $\overline{\texttt{csq}}$ target the same object, in the pre-candidate generation phase all scored segments per object need to be merged into scored sequences for said object.

Definition 4.15 Pre-Candidate Generation Phase

Given a list of results per subquery, $\overline{r_{\texttt{csq}}} = \left(r_{\texttt{csq}_1}, r_{\texttt{csq}_2}, \ldots, r_{\texttt{csq}_n}\right)$ with $n = \|\overline{\texttt{csq}}\|$, the pre-candidate generation phase generates a list of scored sequences $\overline{\varsigma}_{o_i}$ per subquery i and object o. It can thus be viewed as applying $2^{\hat{\mathcal{S}}} \rightarrow 2^{\hat{\mathfrak{S}}}$ to the list of scored sequences per subquery and object.

Semantically, including the optional list of user-specified distances $\overline{\phi}$, and optional maximum distance ω makes little sense, as this step is simply concerned with generating sequences and not yet generating or scoring results. For notation purposes in the remainder of this chapter, we will only refer to $\overline{\varsigma}_i$ for the list of scored sequences for a given subquery, this always assumes the results for a specific object unless explicitly noted.

As an example for such an algorithm, we have implemented a fixed-threshold aggregation, which aggregates two segments if they are closer together than a scenario-specific distance. The algorithm is illustrated in Example 4.11.

Example 4.11 Pre-Aggregation for Video Retrieval

Considering our example of the lion and the giraffe, we might receive the following results for two subqueries, which are both sequential segments:

Figure 4.3 Result segments for two subqueries looking for a lion and a giraffe

It is intuitively sensible to not consider the two segments showing the giraffe as separate sequences, but as one sequence which is relevant for the given subquery.
All segments are from the same object (v_7119). To simplify the example, we ignore the relevance scores per segment. This means that for $\texttt{csq}_1$ we have the following segments:
$s_1 = \langle v\_7119\_34, v\_7119, 34, 96, 101\rangle$
$s_2 = \langle v\_7119\_35, v\_7119, 35, 101, 104\rangle$

For $\texttt{csq}_2$:
$s_3 = \langle v\_7119\_36, v\_7119, 36, 104, 108\rangle$
$s_4 = \langle v\_7119\_37, v\_7119, 37, 108, 113\rangle$

When merging all segments within a fixed threshold, we get two sequences, namely for $\texttt{csq}_1$, $\varsigma_1 = \langle v\_7119, 96, 101\rangle$ and for $\texttt{csq}_2$, $\varsigma_2 = \langle v\_7119, 104, 113\rangle$. The scores of the segments can be merged in an implementation-specific manner (e.g., by taking the maximum relevance score).
For lifelog retrieval, this aggregation is even more relevant, as we might consider, for instance, a drive taking an hour with just pictures of a steering wheel and the view looking out of the car.

Literature Discussion 4.2 Pre-Candidate Generation Steps in other Systems

Some systems perform no pre-candidate generation steps [HDN+22; KSJ+22; HSJ+22; AMG+22; AGG22] and simply use the result segments from subqueries for their temporal fusion model. This corresponds to skipping the pre-candidate generation stage in our model. Looking at our example, it would

mean generating two sequences per subquery and keeping their scores.
Others assume under-segmentation of their result or presentation units [ABC+22] and thus generate sequences which may be smaller than their result units.
In [ABC+22], sequences are generated by applying a fixed length sliding window (7 seconds in VBS 2023) for each subquery and taking as sequence score the maximum score of all segments within that timeframe. This is a different algorithm than the fixed-length merge algorithm we have implemented for the evaluation, but the input and output of the pre-candidate generation stage is the same. They start with a list of scored segments, and generate a list of scored sequences. It corresponds to the first output option of our definition, as the scored sequences are generated per subquery.
Looking at our example, this would result for $\texttt{csq}_1$ in two sequences: one from 96 seconds to 103 seconds, and one from 103 seconds to 110 seconds, each scored with the maximum relevance score of the two segments.

Candidate Generation

Having generated a list of scored sequences per object and subquery, the next step is to generate potentially relevant sequences for the given full query. We call such sequences *candidate sequences* $\tilde{\varsigma}$ defined in Definition 4.16, and there are many ways to construct sequences in a video. We will discuss assigning a score to those candidates afterwards.

Definition 4.16 Candidate Sequence

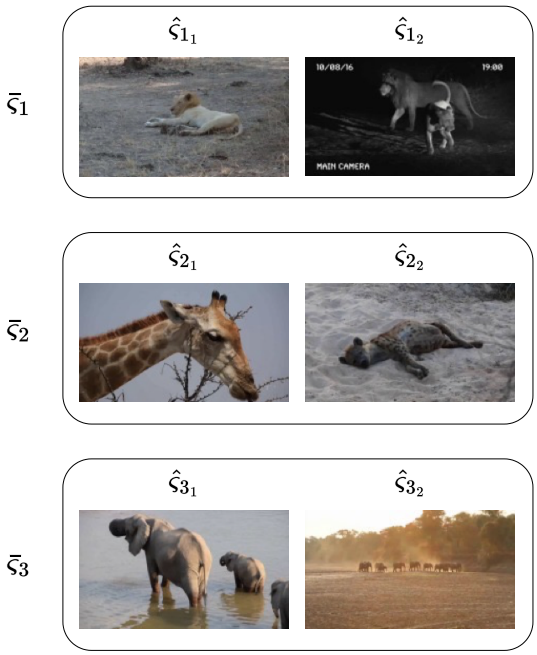
A candidate sequence $\tilde{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$ is a list of scored sequences.

The core idea behind a candidate sequence is that the individual sequences are relevant to a specific subquery. Candidate Generation can thus be viewed as a function that takes the scored sequences per subquery, optional user-specified distances, and an optional maximum length and produces a set of candidate sequences. We define the candidate generation step in Definition 4.17, and provide an example in Example 4.12.

Definition 4.17 Candidate Generation

The candidate generation step $\left(2^{\hat{\mathfrak{S}}}\right)^n \times (\Phi)^{n-1} \times \Omega \rightarrow 2^{(\hat{\mathfrak{S}})^n}$ takes the scored sequences $2^{\hat{\mathfrak{S}}}$ for all subqueries, the user-specified distances $(\phi_1, \phi_2, \ldots, \phi_{n-1})$ and the maximum length $\omega \in \Omega$, and produces a set of candidate sequences $2^{(\hat{\mathfrak{S}})^n}$.

Example 4.12 Candidate Generation in Video Retrieval



Considering again our motivating example for video retrieval, we have results which have been transformed and potentially aggregated into sequences from three different subquery results, $\overline{\varsigma}_1, \overline{\varsigma}_2$, and $\overline{\varsigma}_3$. We must now consider which subquery-spanning sequences we consider as candidates. If there are only two subqueries, one meaningful approach is to consider either temporally close sequences from the next subquery or the next sequences without considering their distance. Once there are more than two subqueries, the question becomes more complex. If we allow subquery misses, we would also consider a candidate sequence that consists only of the lion in $\hat{\varsigma}_{1_1}$ and the elephants in $\hat{\varsigma}_{3_1}$ or $\hat{\varsigma}_{3_2}$. Regardless of whether we allow for subquery misses, at each subquery result, we must consider the question of how many sequences we keep in mind for future options, as the search space grows exponentially. Consider the second subquery. Starting with the lion in $\hat{\varsigma}_{1_1}$, both results for subquery 2, $\hat{\varsigma}_{2_1}$ and $\hat{\varsigma}_{2_2}$, could be considered. If for both of those, all options from the third subquery are also explored, it is easy to see how the potential space of candidate sequences grows large. We call the parameter of how many options are considered for further exploration c in the following.

When it comes to candidate generation, our approach considers the subqueries to be equal and thus potentially allows for retrieval misses in a sub-

query. A significant number of other approaches found in literature assume only two subqueries [HDN+22; ABC+22; HSJ+22; AMG+22] or fix one query as the main query and allow a subquery before and after the main query [AGG22; TNN+22]. In those cases, subquery misses are either punished heavily, result in a sequence not being shown, or are only tolerable for the lesser queries specifying temporal context for a main query. Our approach remains advantageous when using modern machine learning methods, which are still susceptible to noise or misclassifications and is especially important for longer sequences.

The fundamental mechanism we propose for candidate sequence generation is an iterative approach with a configurable threshold which trades off potential quality for speed. For each sequence in a subquery and assuming subquery results are already grouped per object, we construct the best candidate sequence as shown in Algorithm 4.1.

To determine the best candidate for a sequence, we show pseudocode in Algorithm 4.2 which operates on the transformed subquery results generated by the pre-candidate generation phase from Definition 4.15, an optional list of user specified distances $\overline{\phi}$, and an optional maximum distance ω.

For clarity, functions inside the algorithms are noted not through their domains and codomains, but by the input and output variables, which is why $\mapsto$ is used instead of $\rightarrow$ in the following.

Algorithm 4.1 CandidateGeneration(`rcsq` $(\overline{\varsigma})^n$, `dists` $(\phi)^{n-1}$, ω)

Require: BestCandidate : $\hat{\varsigma} \times (\overline{\varsigma})^n \times (\phi)^{n-1} \times \omega \mapsto \tilde{\varsigma}$
Require: `rcsq` is filtered to only contain sequences from a single object
Require: $n \geq 1$
1: $cl \leftarrow ()$ ▷ candidate list
2: **for** $i \leftarrow 1$ to n **do** ▷ subquery index
3: **for all** $\hat{\varsigma} \in \texttt{rcsq}_i$ **do** ▷ best candidate sequence for each sequence in subquery
4: $\tilde{\varsigma} \leftarrow$ BestCandidate($\hat{\varsigma}$, `rcsq`, `dists`, ω, i)
5: $cl \leftarrow cl$ ++ $\tilde{\varsigma}$ ▷ add candidate to list
6: **end for**
7: **end for**

Handling the case where a subquery has no possible sequences can be done in an algorithm-specific manner using the expand function. The simplest version is to simply move the end of the sequence by mapping the user-specified distance to a number.

Evaluating the score of a candidate sequence $\tilde{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$ is also algorithm-specific using the score function, which we will further discuss later.

Generating candidates (genCandidates in the pseudo-code) given an existing sequence and an optional user-provided distance can be done in different

Algorithm 4.2 BestCandidate(`seq` $\hat{\varsigma}$**,** `rcsq` $(\overline{\varsigma})^n$**,** `dists` $(\phi)^{n-1}$**,** ω**, i)**

Require: EXPAND : $(\hat{\varsigma})^n \times \phi \mapsto \hat{\varsigma}$ ▷ there are no results for a subquery
Require: SCORE : $(\hat{\varsigma})^n \times (\phi)^{n-1} \mapsto [0,1]$ ▷ score a candidate sequence
Require: GENCANDIDATES : $\hat{\varsigma} \times \phi \times \overline{\varsigma} \mapsto \overline{\varsigma}$ ▷ generate candidates from a subquery result
Require: ++ operator concatenates a list an element
Ensure: $1 \leq i \leq N$

```
1:  if i == N then
2:      return seq                          ▷ last subquery; no following sequences
3:  end if
4:  best ← null
5:  sequences ← (seq)                       ▷ initial candidate sequence consists of start
6:  while true do
7:      _s ← sequences.POP()                ▷ fetch candidate sequence to check
8:      for j ← (i+1) to N do
9:          if rcsq_i == null then          ▷ check for subquery misses
10:             _s ← EXPAND(_s, φ_j)
11:             continue
12:         end if
13:         c : ς̄ ← GENCANDIDATES(_s, φ_i, rcsq_i)   ▷ generate |c| candidates from next subquery
14:         for x ← 1 to |c| do
15:             _cs ← _s ++ c_x             ▷ new candidate
16:             if SCORE(_cs, dists) > SCORE(best, dists) then
17:                 best ← _cs
18:             end if
19:             if j == N then
20:                 continue                ▷ final subquery; no more candidates to be generated
21:             end if
22:             sequences.PUSH(_cs)
23:         end for
24:     end for
25: end while
```

ways, a select few of which we will briefly discuss here and in Literature Discussion 4.3, where we show how other approaches can be mapped to our model.

Strict Time-Based Filtering: Given a user-provided distance (of semantic nature or simply a number), one approach is to return the best $|c|$ candidates where the distance between two potential scored sequences is below that threshold $\left(\mathrm{D}_{\varsigma} : \mathfrak{S} \times \mathfrak{S} \to \mathbb{R}\right) \leq \phi$. This assumes that a comparison $\leq$ is defined between the distance, which is in $\mathbb{R}$ and the user-specified distance ϕ. This means in practice mapping different ϕ in a scenario-specific manner.

Probabilistic Time-Based Filtering: This approach rewards sequences that match a user-specified distance while not completely ignoring those that do not exactly match that distance.

Index-Based Filtering: When no user-specified distance is provided, or it cannot easily be mapped to a number, one approach is to find all scored sequences in the subquery results after the startpoint, and then simply con-

sidering the first n results. The intuition here would be that we are simply interested in a sequence and not necessarily in the distance between the elements.

Literature Discussion 4.3 Candidate Generation Steps of Other Systems

While not all approaches to candidate generation can be mapped to our model, we will briefly discuss both those that can be and some that cannot.

[HDN+22] brute-force all possible combinations, which means in our terminology generating $\prod_{i=1}^{\|\overline{\mathtt{csq}}\|} \left\|r_{\mathtt{csq}_i}\right\|$ possible candidates and specifically for the genCandidates candidate function simply returning all sequences.

[ABC+22] follow a similar approach to our model, with the candidate-generation criteria being strict time-based filtering and considering only the best match per subquery ($|c| = 1$).

[HSJ+22] also follow a similar approach to our model, they use index-based filtering, looking ahead three sequences at VBS 2022. More formally, this means that if for subquery 2 $\overline{\varsigma}_{2f}$ all items would be after the start sequence, only the first three items of that list would be considered. From those, the best would be returned ($|c| = 1$)

[AMG+22] only consider "the first valid ordered tuple of each video" to calculate the score, which in our model would mean using index-based filtering with a threshold of 1 (as opposed to 3 in [HSJ+22]), only considering one candidate ($|c| = 1$), and not only generating candidates once per video. In our example, this means only keeping the first valid ordered sequence, and not generating any other candidates, therefore no candidate sequence would contain a different lion than the first one, even if the sequence would be scored higher in the end.

[KSJ+22] operate on a video-level, which can be considered as generating a single candidate sequence per video. This sequence is constructed by applying set-operators to the results of the subqueries, namely one of $\{\cup, \cap, \oplus, \setminus\}$, that is union, intersection, minus, or set difference.

[AGG22] also allow a user-specified distance, similar to our model and then consider the highest-scoring sequence ($|c| = 1$). In their model, there is a single main query and two context queries for before and after, and therefore candidate sequences are only generated for the result of a single subquery. This contrasts with our model, where a candidate sequence which matches the first and last subquery, but not the middle one, would be considered.

Candidate Sequence Scoring

Algorithms need a way to evaluate the relevance of a candidate sequence $\tilde{\varsigma}$ for the specified query. This requires generating a relevance score τ for candidate sequences based on the scores of the sequences, and the match between the user-specified distance and the distance between the sequences.

Given the number of subqueries $\|\overline{\texttt{csq}}\|$ and a candidate sequence, meaning one or zero matching sequences per subquery, a scoring function is tasked with scoring the candidate sequence. This can again be considered a problem of combining similarities and we thus reuse the SCF notation. Formally, we define the problem in Definition 4.18 and map other approaches found in literature to our model in Literature Discussion 4.4.

Definition 4.18 Scoring Candidate Sequences

Given a candidate sequence, that is a list of scored sequences $\tilde{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$ and the list of user-specified distances $\overline{\phi}$, we define $\hat{\varrho}_{scs} : \left(\hat{\mathfrak{S}}\right)^n \times (\Phi)^{n-1} \rightarrow [0,1]$, which generates a single relevance score for the given candidate sequence. The first parameter is the candidate sequence to be scored $\tilde{\varsigma}$ and the last one the user-specified distances $\overline{\phi}$.

The reason we do not include the maximum distance ω is because we assume it has been considered in the candidate generation phase. Some approaches in literature normalize the relevance scores of sequences by the maximum score of the results for a subquery [ABC+22], we assume this has been done in the pre-candidate generation phase and therefore we do not include the full results per subquery.

What is useful about this definition is that it is reminiscent of and closely related to the problem discussed in Chapter 3, where we discussed DCF, which grapple with the question of late fusion for results from multiple methods or systems, and the late fusion approaches in Section 4.3.2. This means similar approaches are imaginable: average, median, sum, max, linear combinations, schemes which reward items retrieved by multiple methods such as mnz [SF94], but also negative combinations (e.g., in a lifelog context, instances where the lifelogger left their house but did *not* get in the car).

None of the approaches from literature consider a scenario where adherence to a user-specified distance is not binary. We close this gap in the next section, where we introduce the notion of a *reward function*, which rewards sequences

that match the user-specified distance while not applying a binary threshold.

Literature Discussion 4.4 Candidate Scoring in Other Systems

[ABC+22; HSJ+22; AGG22] sum relevance scores of the subquery results. In our model $\hat{\varrho}_{\texttt{tsq}} = \sum_{i=1}^{\|\overline{\texttt{csq}}\|} \tilde{\varsigma}_i.\tau$.
[AMG+22] apply the inverse exponential function to the sum of differences. In our model (requiring $\|\overline{\texttt{csq}}\| \geq 2$):

$$\hat{\varrho}_{\texttt{tsq}} = \frac{1}{\exp^{\sum_{i=2}^{\|\overline{\texttt{csq}}\|}(\tilde{\varsigma}_i.\tau - \tilde{\varsigma}_{i-1}.\tau)}}$$

[KSJ+22] perform rank-based fusion instead of score-based fusion, and use the average rank of all sequences belonging to a candidate. In our model, this would mean caching the absolute rank of each segment pre-transformation, skipping the transformation step and then scoring candidate sequences based on the absolute rank of the result of each subquery within said subquery.

Transformation To Result Unit: Post-Aggregation

Having generated different sequences with their respective relevance scores, we may now further aggregate these results for result presentation in a post-aggregation step. The abstraction for result presentation can have many forms, in our model we assume that they are also some form of sequence, and thus this step can be seen as further aggregating candidate sequences of length n, or more formally $2^{(\hat{\mathfrak{S}})^n} \rightarrow 2^{2^{\hat{\mathfrak{S}}}}$.

The simplest example is a case where a sequence from subquery 2 $\hat{\varsigma}_{2_a}$ is the best match for two sequences from subquery 1: $\hat{\varsigma}_{1_a}$, $\hat{\varsigma}_{1_b}$. Our results then include $\big((\hat{\varsigma}_{1_a}, \hat{\varsigma}_{2_a})(\hat{\varsigma}_{1_b}, \hat{\varsigma}_{2_a})\big)$ One simple solution is to implement a fixed-length aggregation similar to what has been discussed in the pre-candidate generation step, where two candidate sequences are merged if they are temporally close. Alternatively, at this stage event aggregation algorithms such as the ones found in lifelog literature [ARG22; NLN+22] may be considered, or summarization techniques could be used.

The abstraction level that was chosen for sequences is also relevant when it comes to questions like thumbnail selection and fetching additional information may be required.

We discuss approaches found in literature in Literature Discussion 4.5.

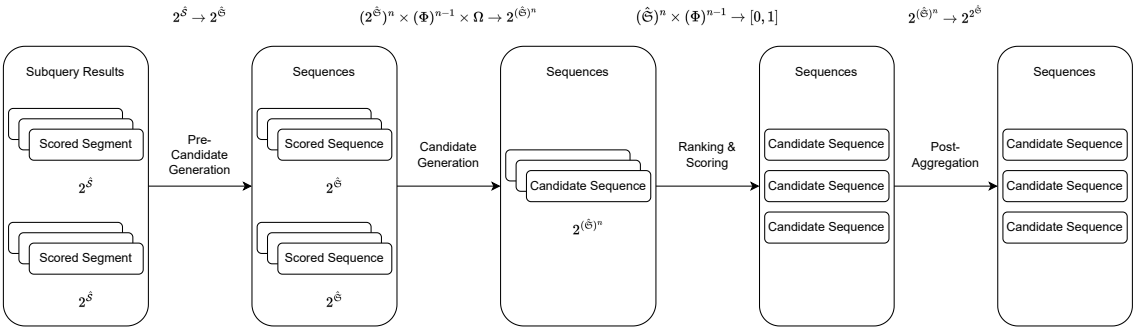


Figure 4.4 Visual overview of late fusion for temporal scoring (Recap from Figure 4.2).

Literature Discussion 4.5 Result Transformation of other Systems

Some systems do not apply a transformation or enhancement [HDN+22; KSJ+22; AGG22]. In some systems, no deduplication steps to candidate sequences are applied [HDN+22; AGG22] and in others it is not necessary because of the model used [KSJ+22].
[ABC+22] avoid showing overlapping content by selecting the highest-scoring sequence if there is overlap.
[HSJ+22] follow a slightly more complicated approach, where only one sequence per best candidate is shown. In terms of our model, this means that if sequence ς_{2_c} is the best match in subquery 2 for both ς_{1_a} and ς_{1_b} from subquery 1, only the highest-scoring combination is kept.
[AMG+22] only generate one candidate sequence per video, and do not perform any other transformations to result units.

4.4.3 Algorithms

Having formulated the temporal query problem, and our theoretical model and approach, we now discuss the temporal fusion algorithms we can potentially derive. To recap, we show the process again in Figure 4.4.

Simple Temporal Scoring

First, we bring together the model into a single modular algorithm. In the first phase, individual segment results are mapped to candidate sequences. This can be turned off to emulate approaches used in earlier work done in this dissertation project on vitrivr [SPG+20; HSS+20; HAG+22] and more recent approaches from literature [HSJ+22].

By default, we have implemented a fixed-length aggregation algorithm where two sequences are merged if their distance is below a scenario-specific threshold. Additionally, we have implemented the pre-candidate generation step described in the VISIONE algorithm [ABC+22] to enable a comparison.

In the second phase, the candidate generation algorithm that is described previously is implemented. The algorithm uses as a stop-criterion by default a strict cutoff after the user-specified distance. For semantic distances ϕ, if simply *afterwards* is specified, all sequences from the following subquery are considered and otherwise the semantic distance ϕ is mapped to a fixed time in a scenario-specific manner. The number of candidates to be considered and the stop criteria itself are configurable, which enables a comparison to prior work published during dissertation project [SPG+20; HSS+20], where the number of candidates was unlimited (no stop criteria), and implementing other approaches from literature such as [ABC+22] with a fixed-length stop criteria. Our version of [HSJ+22] implements this phase separately, as their result transformation step requires caching information acquired during this step.

Scoring is fully configurable, with the default implementation choosing the best-scoring candidates during the candidate generation phase and the average score over a candidate sequence in the candidate scoring phase.

For the result transformation phase, we again implement by default a fixed-length aggregation algorithm, which merges result sequences with a scenario-specific overlap. As in the pre-candidate generation phase, this can be turned off to enable a comparison to prior work and approaches from literature.

Distribution Algorithms

The next family of algorithm relies on the idea that temporal sequences that match the user-specified distances should be rewarded. To this end, all of these algorithms define a *reward function*, which output a score multiplier for sequences inside this temporal sequence based on their adherence to a user-specified distance.

A reward function takes as arguments the current end of a temporal sequence ς_1, a sequence to be considered as a followup ς_2, and a user-specified distance $\phi \in \Phi$ between them and calculates a scoring multiplier. We define the reward function in Definition 4.19.

Definition 4.19 Reward Function

A reward function $\text{REW} : \mathfrak{S} \times \mathfrak{S} \times \Phi \to [0,1]$ determines a score multiplier depending on the adherence of the distance between two sequences $\varsigma_1, \varsigma_2 \in \mathfrak{S}$ to the user-specified distance $\phi \in \Phi$. The function should be monotonically increasing until the maximum, and then monotonically decreasing.
It should have a maximum value of 1, and that maximum value should be reached when the distance between the two sequences exactly matches the user-specified distance.
More formally, $\max(\text{REW}) = 1$, $\text{REW}(\varsigma_1, \varsigma_2, \phi) = 1 \mid \text{D}_\varsigma(\varsigma_1, \varsigma_2) = \phi$, and given two sequences which are temporally ordered $\varsigma_2 \to \varsigma_3$ ($\text{D}_\varsigma(\varsigma_2, \varsigma_3) \geq 0$):

$$\text{REW}(\varsigma_1, \varsigma_2, \phi) - \text{REW}(\varsigma_1, \varsigma_3, \phi) \begin{cases} \geq 0 & \varsigma_1 \to \varsigma_2 \\ \leq 0 & \varsigma_3 \to \varsigma_1 \end{cases}$$

for all functions $\text{D}_\varsigma : \mathfrak{S} \times \mathfrak{S} \to \mathbb{R}$ which determine the distance between two sequences.

We do not require the reward function to be symmetric, which is why for the order $\varsigma_2 \to \varsigma_1 \to \varsigma_3$, no requirements are defined.

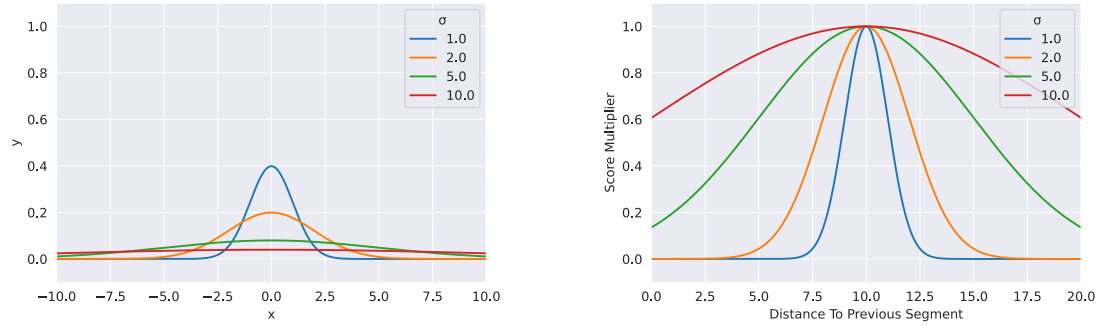
This is due to the fact that for a user-specified distance of 20 seconds, it may be reasonable to treat sequences which are 5 seconds apart differently than those 35 seconds apart.

$\hat{\varrho}_{\texttt{tsq}}$ could already include a handling of the distances. Assuming it does not, we can incorporate the reward function through in different ways, for example through a multiplication, which means given the candidate sequence $\tilde{\varsigma} := (\hat{\varsigma}_1, \hat{\varsigma}_2, \ldots, \hat{\varsigma}_n)$ we calculate the relevance score $\tau_{\tilde{\varsigma}}$ as follows:

$$\tau_{\tilde{\varsigma}} = \left(\prod_{i=1}^{\|\overline{\texttt{csq}}\|-1} \text{REW}\left(\hat{\varsigma}_i, \hat{\varsigma}_{i+1}, \phi_i\right) \right) \cdot \hat{\varrho}_{\texttt{tsq}}\left(\tilde{\varsigma}, \overline{\phi}\right)$$

In this section, we introduce three different algorithms for the reward function based on commonly used distributions: Normal Distribution Algorithm (NDA), Log Normal Decay Algorithm (LNA), and Exponential Decay Algorithm (EDA). These are all based on distributions and serve as an example of reward functions, but do not exhaustively cover all possibilities.

Normal Distribution Algorithm (NDA) One of the most widely used distributions is the normal distribution [Gau23], which has as parameters the mean μ and variance σ, and is shown in Equation (4.1) and visualized in Figure 4.5a.



(a) Normal distribution with varying σ

(b) REW_{nda} with $\mu = \phi = 10s$

Figure 4.5 The normal distribution and the resulting reward function with different parameters

$$\text{ND}(x, \sigma, \mu) \mapsto \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{4.1}$$

As the normal distribution is monotonically increasing until its one and only maxima and monotonically decreasing afterward, we only need to adjust the y-axis to fulfill the requirement that $\max(\text{REW}) = 1$ and choose the x-axis in a sensible way.

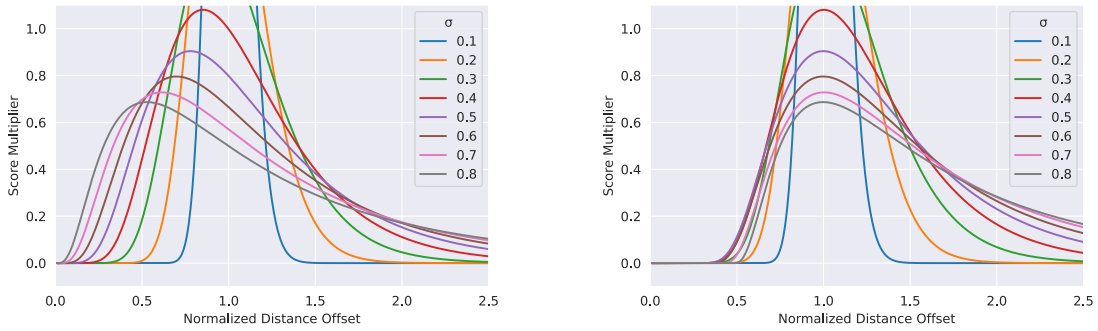
We normalize the y-axis by dividing it through $\max(\text{ND})$, which leads to the NDA reward function shown in Equation (4.2).

$$\text{REW}_{nda}(x, \sigma, \mu) \mapsto \frac{\text{ND}(x, \sigma, \mu)}{\max(\text{ND})} \mapsto \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)}{\max(\text{ND})} \tag{4.2}$$

After normalizing the y-axis, the next consideration is the choice of x-axis. We could choose the distance to the user-specified distance, that is $x = \left(\text{D}_\varsigma(\varsigma_1, \varsigma_2) - \phi\right)$ with $\mu = 0$, or we can choose the distance between the two sequences $\text{D}_\varsigma : \mathfrak{S} \times \mathfrak{S} \to \mathbb{R}$ as the x-axis with $\mu = \phi$. This choice does not matter a great deal, as in both cases the distribution is symmetrical. We choose the second option and visualize the resulting reward function in Figure 4.5b.

However, in both cases the x-axis is directly correlated to the distance between two sequences. This means that depending on the scenario, a different σ might be needed[9].

[9] Alternatively, one might consider normalizing the x-axis relative to a fixed value to the dataset or to the user-specified distance.



(a) Varying σ, original pdf

(b) Varying σ, shifted x-axis based on maximum

Figure 4.6 Probability density function of a lognormal distribution without and with shifted x-axis

Log Normal Decay Algorithm (LNA) Another option to model a reward function is a lognormal distribution[10]. Such distributions are used in fields as diverse as meteorology [BL15] and neuroscience [BM14].

Equation (4.3) shows the probability density function of a lognormal distribution of a random variable x, which requires as parameters mean μ and variance σ of $\ln(x)$.

$$\text{PDF}_{lnd}(x, \sigma, \mu) \mapsto \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right) \tag{4.3}$$

In our work, we build on this distribution for the LNA algorithm. For the x-axis, we consider the ratio between the distance of the two sequences:

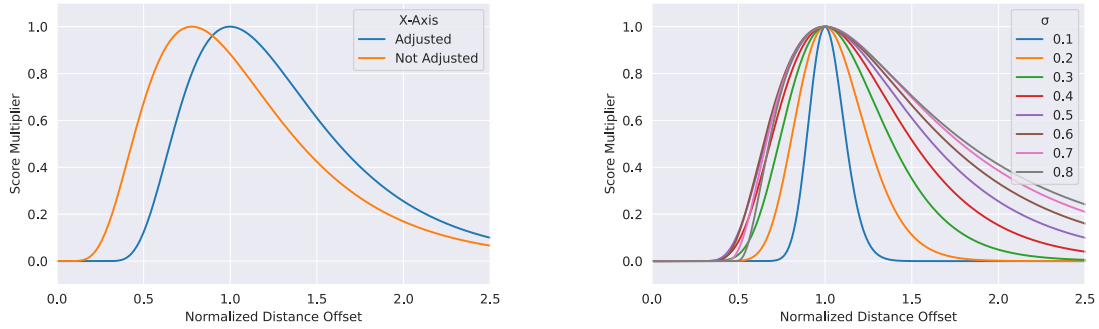
$$x_{lna} = \left\| \frac{\mathrm{D}_{\varsigma}(\varsigma_1, \varsigma_2)}{\phi} \right\|$$

We thus reward sequences that fit the specified distance while still considering those that are not a perfect match. This means that our optimal case is $x_{lna} = 1$ and since $\ln(1) = 0$, $\mu = 0$. We visualize the probability density function in Figure 4.6a.

As clearly visible in Figure 4.6a, depending on the chosen σ, the maximum reward will not always be at $x_{lna} = 1$. We thus shift the x-axis relative to where PDF_{lnd} has its maximum with the given σ, μ. Given $x_{max} = \{x | \text{PDF}_{lnd}(x, \sigma, \mu) \geq \text{PDF}_{lnd}(u, \sigma, \mu), \forall u \in \mathbb{R}_{\geq 0}\}$, the shift is defined in Equation (4.4).

$$\hat{x} = \max(0.01, x_{lna} - (1 - x_{max})) \tag{4.4}$$

[10]Which has no commonly agreed upon origin story, see [Gad45]. Galton [Gal79] is often credited with the fundamental idea.



(a) LNA reward function without and with x-axis offset

(b) LNA reward function with varying σ

Figure 4.7 Visualizations of the LNA reward function

The motivation for the shift to ensure that $\max(\text{REW}_{lna}) = \text{REW}_{lna}(1)$ is given in Example 4.13, and the resulting function is visualized in Figure 4.6b. As the x-axis is proportional to the user-specified distance, the chosen parameters do not necessarily need to vary based on the application.

To achieve $\max(\text{REW}_{lna}) = 1$, we normalize the y-axis similar to the approach for NDA. The difference between an adjusted and non-adjusted x-axis after normalizing the y-axis is shown in Figure 4.7a

Example 4.13 X-axis offset for LNA reward function

To illustrate why an x-axis offset is needed, consider the simple example of $\sigma = 0.5$, $\mu = 0$. As the local maxima is at $x = \frac{1}{\sqrt[4]{e}}$, we compute both y without adjustments $y = \text{PDF}_{lnd}(x, \sigma, \mu)$ and with adjustments $y_s = \text{PDF}_{lnd}(\hat{x}, \sigma, \mu) \mapsto \text{PDF}_{lnd}\left(\max\left(0.01, x - \left(1 - \frac{1}{\sqrt[4]{e}}\right)\right), \sigma, \mu\right)$:

x	y	y_s
0.6	0.78	0.31
$\frac{1}{\sqrt[4]{e}}$	0.90	0.72
1	0.79	0.90
1.4	0.45	0.64

This results in the desired property $f(1) = \max(f(x))$. We visualize the difference in reward functions with and without an offset when fixing $\sigma = 0.5, \mu = 0$ and additionally normalizing the y-axis in Figure 4.7a.

Given that $\hat{x}$ indicates a modified x_{lna} as defined above, the reward function

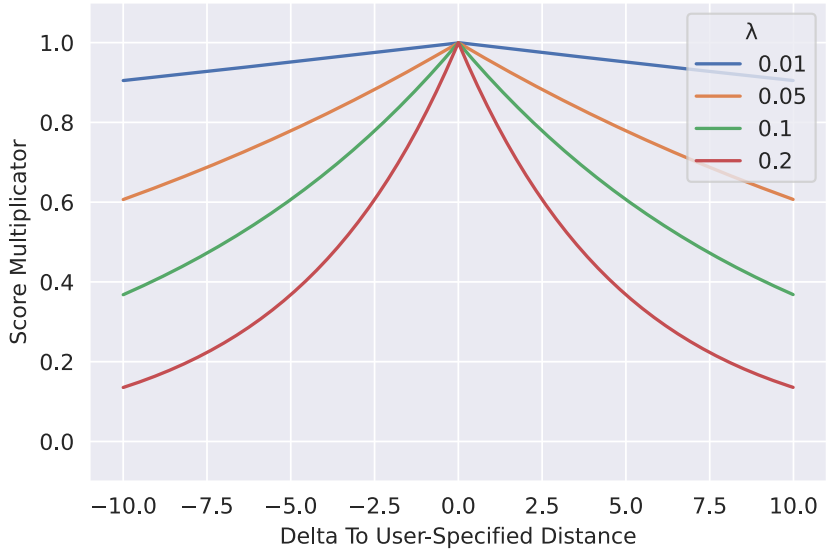


Figure 4.8 EDA reward function with different λ

for LNA is shown in Equation (4.5).

$$\text{REW}_{lna}(x_{lna}, \sigma, \mu) \mapsto \frac{\frac{1}{\hat{x}\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(\hat{x})-\mu)^2}{2\sigma^2}\right)}{\max(\text{PDF}_{lnd})} \tag{4.5}$$

We show the LNA reward function with different σ and fixing $\mu = 0$ in Figure 4.7b.

Exponential Decay Algorithm (EDA) Another option to model a reward function is an exponential decay distribution. Such distributions are used in different contexts such as modeling tail distributions of mobile devices [KLBV10]. The key idea which makes the distribution useful in our context is that we want the reward function to decrease proportionally the further it strays from the ideal distance. Exponential decay functions are characterised by two parameters: N_0 which is the value of the function at $x = 0$, and an exponential decay constant λ. We show the exponential decay function used for EDA in Equation (4.6).

$$\text{ED}(x) \mapsto N_0 e^{-\lambda x} \tag{4.6}$$

For our scenario, we need to make a slight adjustment. The trivial choice is to set $N_0 = 1$, as we wish to have a perfect reward for an exactly matching distance. Given a decay constant $\lambda > 0$, we define our reward function REW_{eda} as follows in Equation (4.7). Additionally, we invert λ depending on the distance of the candidate sequence being smaller or larger than the user-specified distance, to make the decay exponential in both directions.

$$\mathrm{REW}_{eda}(x, \lambda) \mapsto \begin{cases} e^{\lambda x} & x > \phi \\ e^{-\lambda x} & x < \phi \\ 1 & x = \phi \end{cases} \tag{4.7}$$

We show how the reward function changes for different values of λ in Figure 4.8.

To make the function fit different scenarios[11] we can either normalize the distance to the user-specified distance or vary λ. The considerations are similar to the ones discussed for previous distributions, for EDA it is convenient to vary λ dependent on the scenario.

[11] In the context of video retrieval, 10 seconds is a reasonable distance for a user to specify, while in lifelog retrieval, distances are more commonly specified in hours.

For the most part, things never get built the way they were drawn

— Maya Lin

5

vitrivr: A Multimodal Multimedia Retrieval System

The conceptual model for multimedia retrieval presented in this thesis is backed by an implementation in vitrivr, an open-source full-stack multimedia retrieval system. In this chapter, we present vitrivr with a focus on the conceptual system design. We give an overview of the architecture and describe the two components at the center of this thesis, the retrieval engine and the user interface. In doing so, we cover the user journey with extraction, query formulation, and result presentation.

Before doing so, we will briefly discuss and delineate the contributions made to the vitrivr system which are also referenced and described in this chapter. Software development is collaborative work and vitrivr is no exception[1]. vitrivr was originally introduced in [RGT+16] and has grown out of the IMOTION stack [RGS+15; RGH+16; RGT+17], and parts of it have been the subject of previous [Gia18; Ros18] and concurrent [Gas23] dissertations.

This chapter contains partial content from, and summarizes implementation contributions made in (co-)authored peer-reviewed publications [RPG+19; RGH+19; GRH+20; HSS+20; HPP+20; SPG+20; HPG+20; HGI+21; HGP+21; SGH+21a; SGH+21b; HAG+22; HRS+22; SGH+22; HSS23] and [**rossettoDeepLearningBasedCo** HGG+23]. A complete list of publications of the vitrivr project is available online[2]. vitrivr received contributions from numerous Bachelor and Master

[1] While the full commit log of all components is openly available, distinguishing conceptual and implementation work, which is sometimes also done collaboratively via pair programming or designing, is rarely a sensible enterprise. vitrivr nowadays squashes pull requests into one commit, which can lead to misleading statistics. The repositories have also moved between platforms, and the commit history has not always been transferred.

[2] `https://dbis.dmi.unibas.ch/research/projects/vitrivr-project`

projects, some of which supervised during this dissertation [Pas20; Nem20; Gst21; Ill21; Pop21; Ben22a; Pet22], and some the author's own [Hel16; Hel18]. Additionally, vitrivr has demonstrated its feasibility and attractiveness as an open-source research system in different ways, for example through its participation at Google Summer Of Code (GSOC) in 2016, 2018 and 2021[3]. In past years, state-of-the-art retrieval methods such as CLIP [RKH+21] were integrated into vitrivr, and novel retrieval methods (e.g., for OCR [TRB22][4]) or textual embeddings for video retrieval [SGH+22] were developed in vitrivr.

These implementation contributions enable vitrivr to be used in various contexts outside video and lifelog retrieval, such as cultural heritage [SRS18; RSS+21; PSS+22] and retrieval for speech transcription [SLT+21]. It is now also used in two large-scale interdisciplinary projects in the context of Virtual Reality (VR)/Augmented Reality (AR) and cultural heritage [Wel22; LFF22].

The database layer, Cottontail DB, is only described briefly in this chapter, as it is subject to a separate dissertation [Gas23] and has only received minor contributions in this dissertation project in [GRH+20; GRH+21]. For the purpose of this thesis, it is relevant to note that Cottontail DB supports all relevant retrieval modes introduced in this thesis and used by the retrieval engine. In contrast to its predecessor ADAM$_{pro}$, Cottontail DB is a single-node database, which places limitations on retrieval efficiency as the amount of data grows, but means it is a better fit for most evaluation scenarios in contemporary retrieval research.

This chapter starts with an architecture overview in 5.1, then covers the retrieval engine and user interface in Sections 5.2 and 5.3.

5.1 System Architecture

vitrivr follows the traditional three-tier architecture of information systems [Sch18] similar to what was introduced in Section 3.3, which separates three areas of concern: data management, application logic, and presentation layer. This separation means components are easily replaceable and can be used individually in research applications. We show an architecture overview in Figure 5.1, which shows the different components vitrivr consists of:

Database Engine: vitrivr requires support for various retrieval models, such as vector space retrieval, text retrieval, and traditional Boolean retrieval, as described in Section 3.2. This is delegated to a dedicated storage layer. The

[3]The author was one of the mentors and org admins for vitrivr 2018 and 2021.

[4]HyText was developed for vitrivr, see `https://github.com/vitrivr/cineast/pull/212`.

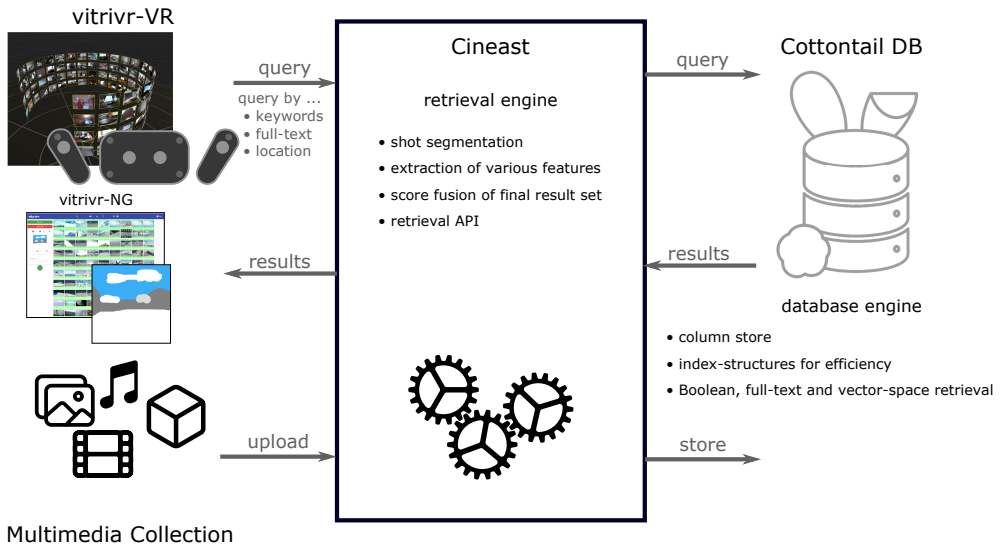


Figure 5.1 Architecture of the vitrivr ecosystem, with the shared components of the retrieval engine and the database. Two different frontends are shown, the traditional desktop UI, and a VR one. Taken from [HSS23]

retrieval features described in Chapter 4 rely on the database engine to perform retrieval using the appropriate retrieval model.

Retrieval Engine: The retrieval engine is responsible for feature extraction from document collections. It receives and orchestrates queries and processes the results. It is thus responsible for handling and processing multimodal and temporal queries as described in Sections 4.3 and 4.4.

User Interface: To formulate the kinds of queries described in Section 4.2 and browse results, vitrivr offers a desktop-based user interface. Other projects have built upon the retrieval and database engine and created user interfaces for retrieval in mobile devices [SRS18] or VR [SGH+22].

Individual components can be easily replaced, such as in the transition of the database layer from ADAM$_{pro}$ [GS16; Gia18] to Cottontail DB [GRH+20; GRH+21; Gas23]. They can also be used individually in different applications which only need one component of the system. Examples of this include medical applications for Magnetic Resonance Fingerprinting (MRF) [Gas23] where only the database layer is used, different lifelog retrieval systems [RBA+20; RBG+21] that re-use the frontend with different backends, mobile applications that use both retrieval engine and the database [SRS18], or VR museum applications [PSS+22].

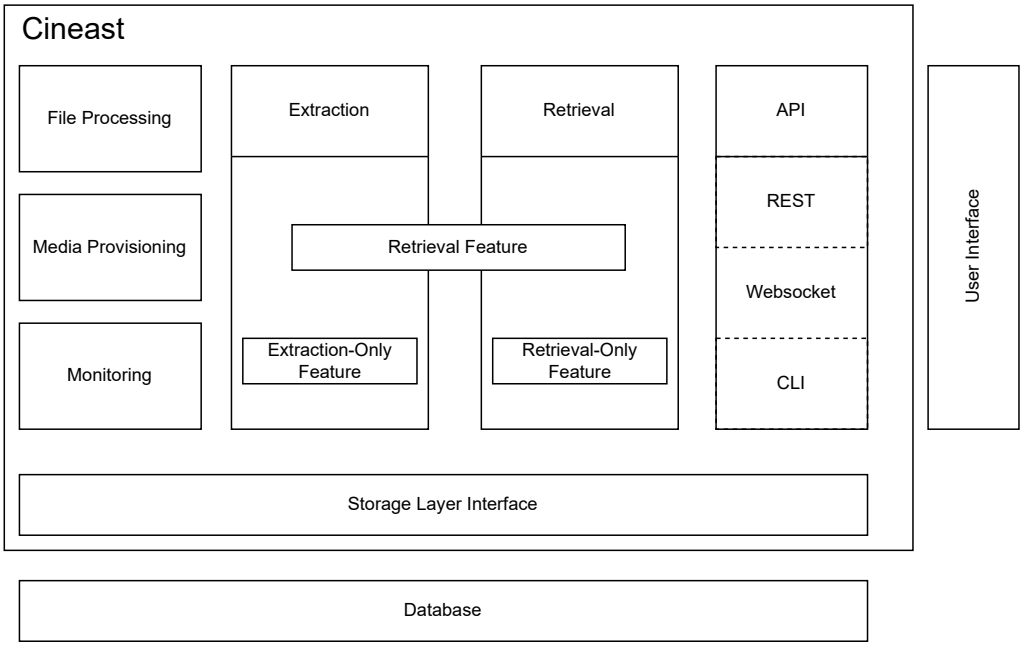


Figure 5.2 Cineast Architecture. Based on [Ros18; HGG+23]

5.2 Retrieval Engine

The retrieval engine of vitrivr is called Cineast and was originally introduced in [RGS14]. It offers various important functionalities such as media segmentation and extraction, and a full-fledged retrieval API. Significant contributions were made to all parts of the system, including features, retrieval model, API, extraction, and performance improvements. In this section, we will give an architectural view of the functionality of Cineast to contextualize those contributions and show how the presented concepts are implemented. For a comprehensive and detailed overview of its core functionality for video retrieval, such as the features used in sketch retrieval, we refer to [Ros18].

Figure 5.2 shows an architectural overview of Cineast and how it fits into the bigger picture. We see two major modules: extraction and retrieval, which correspond to the offline and online parts described in Chapter 3. Some classes are only used during extraction (e.g., thumbnail generation), others only during retrieval (e.g., retrieval features querying externally generated features), and some are used during both. There is a REST and Websocket API used by different frontends, and a CLI. The database layer is abstracted to enable switching between different implementations such as ADAM$_{pro}$ [GS16], Cottontail DB [GRH+20] or a JSON implementation for testing. Additionally, Cineast needs to decode and segment multimedia files, serve multimedia and thumbnails during runtime and offers monitoring capabilities during long-running extraction tasks and for

productive use.

Following the structure of Chapters 3 and 4, we first cover the extraction module and then the retrieval module.

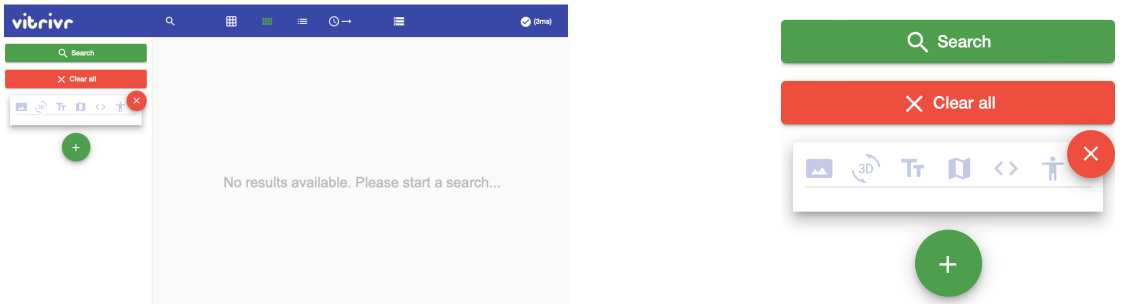
Information about multimedia objects $o \in O$ to be extracted is provided either via API or via CLI (e.g., when running a distributed extraction for large multimedia collections). The objects are then distributed to an appropriate handler (videos, images, 3D models, and lifelog collections need to be handled differently), which segments them as defined in the previous chapter ($\text{SEG} : O \to 2^{\mathcal{S}}$). Each segment $s \in \mathcal{S}$ is then processed by specified classes, either extraction-only classes (such as metadata extractors or thumbnail generators) or feature classes which are also used during retrieval ($\mathscr{f}_s : \mathcal{S} \to \mathcal{F}$). Afterwards, the generated feature representations $f \in \mathcal{F}$ are stored in the database layer for future retrieval purposes. These representations can not only be vectors as in previous work [Ros18], but also text or other meaningful and useful representations for retrieval. The extraction API enables Cineast to be integrated into broader multimedia analytic pipelines, for example when collecting and analyzing social media data for political sciences [Pet22].

In the retrieval module, Cineast uses mainly late fusion of retrieval results. As discussed in the previous chapter, each query term is delegated to the specified retrieval features, where a list of scored segments is returned ($\mathscr{f}_r : QT \to 2^{\hat{\mathcal{S}}}$). For efficiency reasons, only the segment id is fetched. Associated metadata and segment information is only retrieved at the end of the fusion process. Afterwards, the implemented SCF are applied, and temporal scoring is performed as discussed in the previous chapter. Generally speaking, each retrieval feature is evaluated independently and in parallel, but for some queries, it is desirable to have an execution order. An example of this would be the $\hat{\varrho}_{k1}$ operator introduced in the previous chapter, which we also call staged queries in [HSS+20]. An overview of the current features used in Cineast with a specific focus on Lifelog retrieval can be found in [HSS23]. The current implementation of temporal scoring in the main branch is described in [HAG+22] and is based on the EDA algorithm described in Chapter 4.

Cineast is fully open-source[5] and written in Java. It uses a plethora of libraries for processing and analyzing multimedia. To list a few, TensorFlow [AAB+15] for retrieval features that use machine learning, BoofCV [Abe16] and OpenCV [Bra00] for video and image processing is done, and JavaEWAH[6] [LKA10; KL16] for binary vectors in the context of near-duplicate detec-

[5] `https://github.com/vitrivr/cineast`

[6] `https://github.com/lemire/javaewah`



(a) Entry point

(b) Toggles for different query modalities

Figure 5.3 Overall entry point and toggles for different modalities

tion [Hel18]. The API and CLI use gRPC[7], Javalin[8], and Airline[9], and monitoring is implemented using Prometheus[10] and Grafana[11].

5.3 User Interface

Fundamentally, a retrieval user interface should address the three basic requirements of query specification, inspection of results, and query reformulation [BR11]. vitrivr-ng is the desktop-based user interface for vitrivr and the first iteration was introduced in [Gas17; GRS19a; GRS19b]. It is implemented in Typescript and uses the Angular Framework[12]. Query formulation happens on the left side of the screen, and results are displayed in the center. Different result views can be toggled in the header. Significant contributions were made to all parts of the frontend, both conceptually and in the implementation. This section draws on the conceptual and implementation contributions mentioned at the beginning of this chapter, and specifically contains a significant amount of content from [HSS23], where we systematically introduce and compare the interfaces for vitrivr-ng and vitrivr-VR in the context of lifelog retrieval. For the sake of readability, overlap with that paper is not quoted explicitly.

5.3.1 Query Formulation

Figure 5.3 shows how the user interface looks when the user encounters it and a closeup of the empty query formulation view. In vitrivr-ng, all modalities can

[7] https://grpc.io
[8] https://javalin.io
[9] https://rvesse.github.io/airline
[10] https://prometheus.io
[11] https://grafana.com
[12] https://angular.io

be toggled. New subqueries can be added by clicking on the green plus button, which allows specifying temporal context. Users express their information needs by formulating queries using the different modalities available, which usually translate to queries involving one or multiple of the aforementioned features. We briefly list the modalities supported by vitrivr, and then show what the user interface looks like for a selection of those.

Sketch: Hand-drawn sketches, retrieval is performed based on color and shape features [RGS14; Ros18]

Aural: Audio samples or recorded input, retrieval is performed based on audio features [Gas17; GRS19b]

Pose Queries: Queries specifying the pose of one or more people [Par21; HAG+22]

3D: 3D model similarity search based on 3D model descriptors [Gas17; GRS19b; BGS+20]

Semantic Sketch: Sketching of the spatial distribution of different high-level concepts such as „mountain“ or „sky“ [RGS19]

Fulltext: Text input used for search in textual information (e.g., ASR, OCR), or textual embeddings [RPG+19]

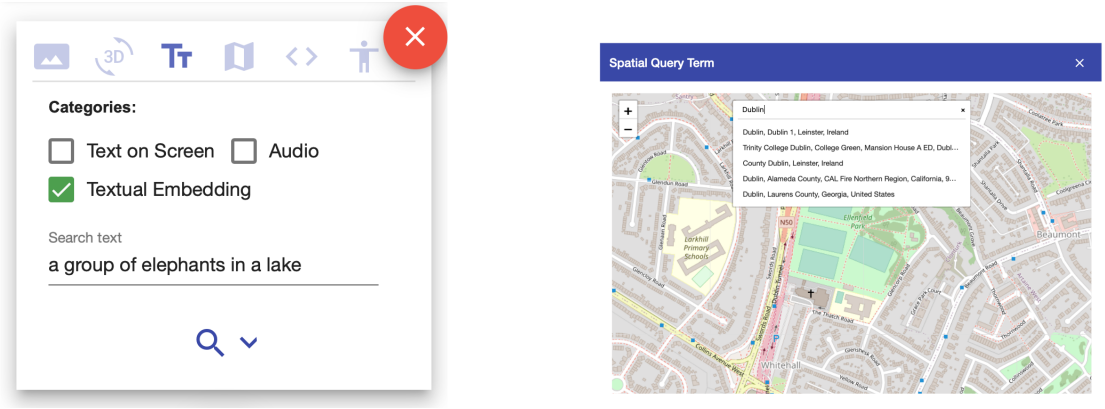
Tags: Queries for specific tags assigned by an object classifier [RPG+19; RGH+19; SPG+20].

Boolean: Classic metadata retrieval [HPG+20]

Maps: Location-based retrieval used in applications such as cultural heritage [BS16] or lifelogging [HGP+21]

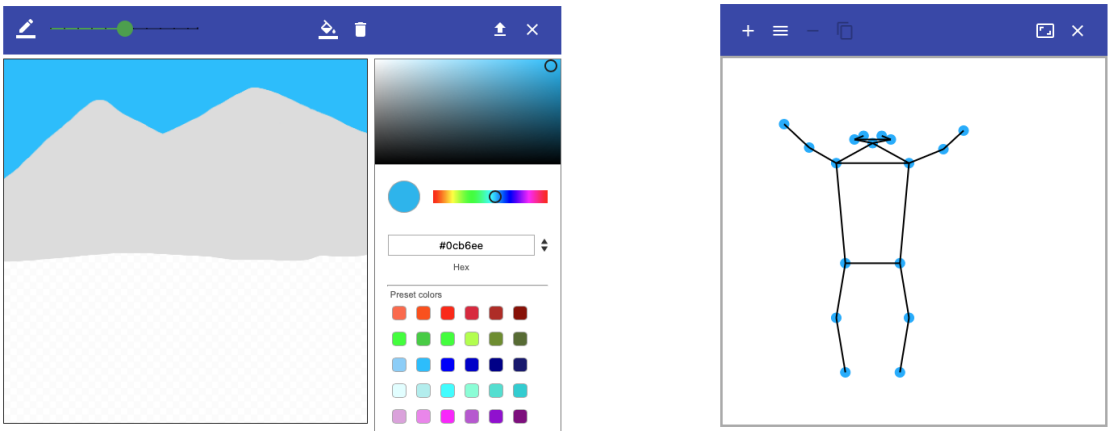
We show the interfaces for textual and geospatial queries in Figure 5.4. Textual queries for OCR, ASR, or textual embeddings are specified with a traditional text input field, where boxes can be checked depending on the desired features. For information needs with a spatial context, vitrivr-ng supports the simple use case of putting a pin on the map and searching for segments in proximity. This is implemented using Leaflet[13] and OpenStreetMap[14], and the leaflet-geosearch package[15] is used for location lookup independent of the dataset (i.e., searching

[13] `https://leafletjs.com`
[14] `https://www.openstreetmap.org`
[15] `https://github.com/smeijer/leaflet-geosearch`



(a) Textual queries (b) Geospatial queries

Figure 5.4 Textual and geospatial query formulation view in vitrivr-ng

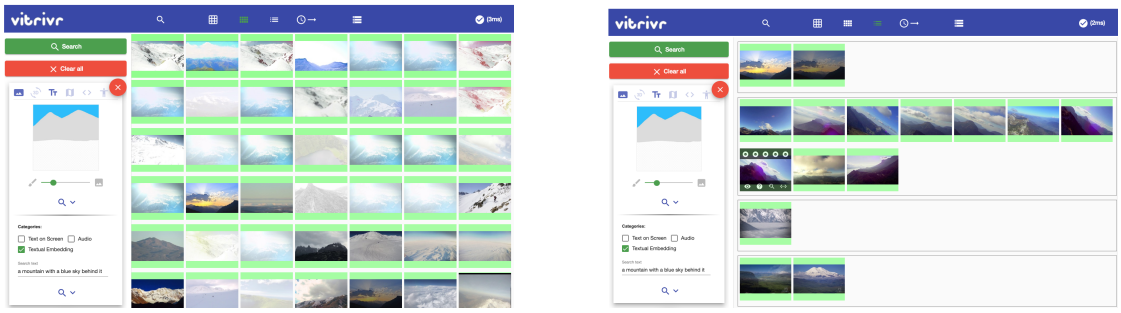


(a) Sketch queries (b) Pose queries

Figure 5.5 Sketch and pose query formulation views in vitrivr-ng

for „Dublin“). The user interface for geospatial queries is partially based on experiences gained during a student project supervised during this dissertation project [Pop21].

The interfaces for the sketch and pose modality are shown in Figure 5.5. Sketch queries were the motivation for the original Cineast system [RGS14], and the current query formulation view features a palette of frequently used colors and a size-adjustable pencil to draw. Pose queries are the most recent addition to the vitrivr system [HAG+22]. The user interface allows specifying multiple independent people and has individually adjustable keypoints. It does not yet allow specifying occlusion or rotation, which is subject to research in pose-based retrieval [Par21].



(a) Ranked segment view (b) Object view

Figure 5.6 Result displays where segments are either ranked individually, or grouped by object

5.3.2 Result Presentation and Browsing

Displaying results means either showing the full document or a compact representation, sometimes called *surrogate* (e.g., a thumbnail of a video shot) [BR11]. For result presentation of videos, we use shot-based thumbnails as widely recommended in literature [GWG+03; BN07] and also used in other state-of-the-art retrieval systems [HGB+22].

vitrivr-ng has three important result presentation views: a ranked segment view, a view that groups segments based on their object, and a view that considers specified temporal context. The first two are shown in Figure 5.6, and the temporal context view is shown in Figure 5.7. We describe the different result views in the following and show examples.

In all the views, the background of the thumbnail is colored according to the score of the element shown, with a dark green indicating a relevance score of 1 and a white background indicating a relevance score of 0. Hovering over the thumbnails reveals additional information, such as relevance feedback functionality, and the possibility to inspect metadata and associated features. Clicking on a thumbnail opens the associated segment or object data in a sensible manner. Examples include jumping to the corresponding point in the video, opening a 3D viewer for 3D objects, and a IIIF[16] viewer for images served from a IIIF server, which is frequently used in the context of cultural heritage.

The default view orders individual segments (e.g., shots in the context of video retrieval, individual images in the context of lifelog retrieval) by their score, with the segment with the highest relevance score shown in the top left.

In the object view, all segments belonging to an object are shown together. The score is either calculated through max- or averagepooling of individual seg-

[16] `https://iiif.io`

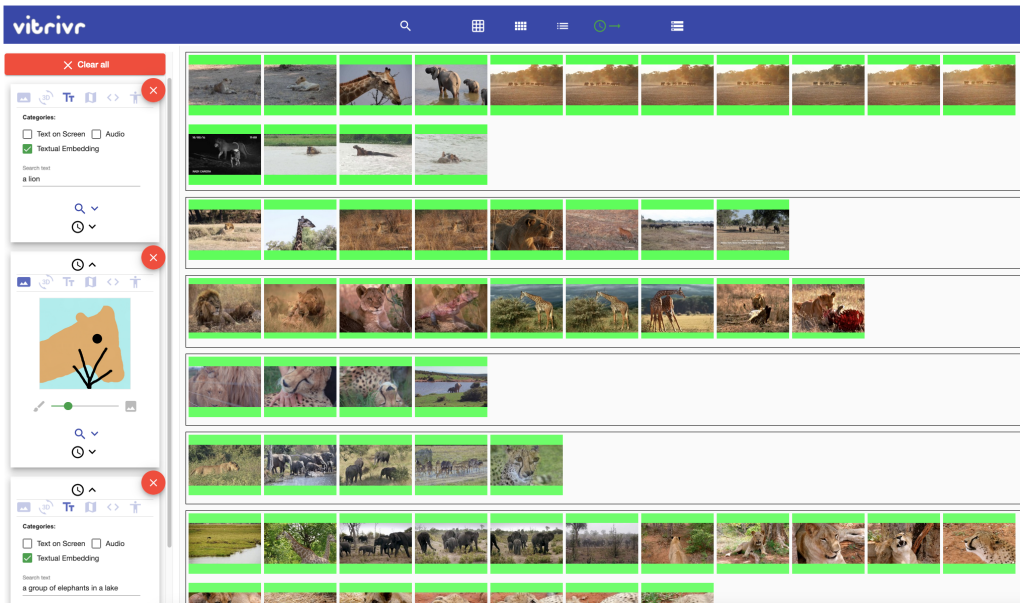


Figure 5.7 Result view for the temporal query example used in Chapters 2 and 4. Query formulation view adjusted for ease of understanding

ment scores. One scenario where the object view is useful is when many relevant segments are from a small set of videos which can from quick visual inspection be judged as irrelevant.

The temporal context result view is similar to the object view but shows the sequences calculated by the algorithms shown in Chapter 4 and sorts those sequences by their score. This offers additional motivation for the post-aggregation step because sequences with overlapping content would mean significant additional effort for the user when browsing results with little gain.

When browsing results, users can additionally filter them without reformulating the query based on available metadata [HPG+20] (e.g., by specifying the day of the week or location).

Nicht alles, was zählt, kann gezählt werden, und nicht alles, was gezählt werden kann, zählt

— Albert Einstein

6

Evaluation

In this chapter, we evaluate our contributions and show the feasibility of the conceptual model and implementation. In addition to a traditional system-centered evaluation, we include results from user-centered evaluations. These focus on a more holistic comparison of retrieval systems, and consider interactions with real users from query formulation to result presentation, inspection, and browsing.

In the user-centered evaluation, we analyze results from four years of interactive retrieval evaluation campaigns for video and lifelog retrieval, with a perspective both on our contributions and a broader view of the format and learnings.

In the system-centered evaluation, we compare retrieval quality and speed of the different algorithms introduced in this thesis and discuss their reliance on underlying retrieval functionality by comparing two features for text embedding.

Both parts of the evaluation are centered around the motivating scenarios described in Chapter 2, where users know the item they are searching for with their information need.

We first show results from user-centered evaluations, then the system-centered evaluation follows, with a systematic walkthrough of requirements, methods, and results, and then the chapter closes with a summary and discussion of the evaluation.

Figure 6.1 Keyframes from a Visual Known Item Search (V-KIS) task. A sequence of 25 seconds was shown. The example is from VBS 2021 [HGB+22]

6.1 User-Centered Evaluation: Interactive Evaluation Campaigns

In this section, we first motivate interactive evaluation campaigns[1], and afterwards describe two such campaigns, VBS and LSC, and the results and learnings of our participations from 2019–2022. For VBS, there are publications that analyze the results of the campaign [RGL+21; LVM+21; HGB+22], while for LSC, one is available for the 2018 [GSJ+19] and 2021 iteration [TND+23], but none for 2019–2020.

To contextualize the contributions of this section, the vitrivr system has participated successfully to VBS and LSC multiple times during this dissertation project (VBS: [RPG+19; SPG+20; HGI+21; HAG+22; SGH+23], LSC: [RGH+19; HPG+20; HGP+21; HRS+22]), achieving the highest score in 2019 and 2021 at VBS and 2019 at LSC. In addition to the yearly iterations of the interactive evaluation campaigns, the format has also been used for the evaluation of student projects supervised during this dissertation [Pas20; Ill21] and in a large-scale comparison [RGH+21] of vitrivr and SOMHunter [KVM+20]. These contributions have not only informed developments of vitrivr, but also work on evaluation tooling by others [RGS+21; SGB+22], and evaluation methodology [LBB+22]. This section will draw on co-authored work, which attempts to provide a systematic categorization for user-centric comparative multimedia search evaluations [LBB+22] and significantly on the analysis of VBS 2021 [HGB+22][2].

After analyzing and discussing results from the 2019–2022 participations at VBS and LSC, we will discuss insights and recommendations based on the experience gained.

[1] Sometimes also called benchmarking campaigns.

[2] Contribution statement: lead author with responsibility for structure, coordination, supervising and determining analysis methods (except Sections 4.3.3, 4.3.4, and Figure 4), significant contributions to writing all sections (except Sections 1, 4.3.3 and 4.3.4).

Table 6.1 Textual Known Item Search (T-KIS) task t-2 from VBS 2021 with its descriptions, which get more detailed over time. After 120 seconds, the full description is revealed, the task duration is 420 seconds [HGB+22]

Time	Text
0s	A hand opening and closing a window of a mountain hut.
60s	A hand opening and closing a window of a mountain hut. There are snow covered mountains outside.
120s	A hand opening and closing a window of a mountain hut. There are snow covered mountains outside. The weather is sunny, the shadow of the hut is visible in the snow.

6.1.1 On Interactive Retrieval Evaluations

Since different multimedia retrieval systems will have significant differences in not only retrieval models and functionality but also in their user interaction approaches, a fair comparison of different systems is challenging. Additionally, as prominently mentioned by [BR11, p. 131], "To evaluate an IR system is to measure how well the system meets [information needs]. This is troublesome, given that [the] same result might be interpreted differently by distinct users". One approach to tackle this problem is benchmarking campaigns, in which different systems are compared against one another in controlled environments[3]. Benchmarking campaigns have also been motivated in literature: "It is desirable to have a forum or gathering at regular intervals for discussing different approaches, as well as their respective performance and shortcomings using the evaluation strategy" [DJL+08].

Two key challenges in these benchmarking campaigns are that participant motivation is essential when evaluating interfaces [BCB+05; Spo02], and that they should "adequately reflect user interest and satisfaction" [DJL+08]. Examples of these evaluation campaigns for information retrieval include TREC [ABC+21b], CLEF [SCIG+21], NTCIR [GJH+19], ImageCLEF [IMP+19] and MediaEval [CHL+20].

Two examples of interactive benchmarking campaigns are VBS for video retrieval and LSC for lifelog retrieval, and their format is very similar. Search tasks are defined on a dataset, and users with different experience levels (i.e., novices

[3]Which variables are controlled depend on the campaign. Some restrictions may include used hardware, time of day during which the evaluation is done, or available preparation time.

and experts) solve the tasks simultaneously in a real-time setting. Each participating team brings its own system, and thus the entire system ranging from UI to retrieval features, retrieval efficiency, and engineering is evaluated. Both benchmarking campaigns we discuss in this evaluation have KIS tasks. In Figure 6.1, we provide an example of a Visual Known Item Search (V-KIS) task, and Table 6.1 shows an example of a Textual Known Item Search (T-KIS) task.

For T-KIS tasks, a textual description of the desired scene is gradually revealed. The textual description is inherently an incomplete representation of the original item and thus models a realistic setting with limited recollection. However, there can be ambiguity, especially when considering cultural and language barriers in understanding the provided description. This ambiguity is a somewhat problematic limitation, especially given that not all participants and users are native speakers of English.

For V-KIS tasks, a unique video clip of approximately 20 seconds is shown to all users, and they have to use their retrieval systems to find the clip in the dataset. These tasks are unique to VBS, as in the visual lifelog setting, such an information need is deemed not very interesting. Even though they are a staple of VBS, they do not necessarily model a realistic scenario, as the presentation of the video is not obfuscated in any way which would model human perception and memory [RBB21].

LSC experimented in 2022 for the first time with Q&A tasks, where the scenario is that a person has an information need specific to their memory (e.g., when did I last use my hammer?), and an item from the collection has to be submitted which contains the correct answer.

As the scoring function for KIS tasks in VBS and LSC is the same, we will briefly recap it here [HGB+22]: Given a linearly decreasing function f_{ts} based on search time, the time of correct submission t and the wrong submissions w, the score for a given KIS task is as follows:

$$f_{kis}(t, w) \mapsto \max(0, 50 + 50 \cdot f_{ts}(t) - 10 \cdot \|w\|)$$

f_{kis} thus awards at least 50 points for a correct submission if no wrong submission was made and penalizes each wrong submission with a malus of 10 points.

In the following, we focus on comparing expert users, as only VBS 2019 out of the eight analyzed benchmarking campaigns had a novice session. Generally speaking, the literature suggests a significant performance difference between novices and experts [HC04], which has been replicated at VBS 19 [RGL+21].

6.1.2 Video Browser Showdown (VBS)

VBS is usually collocated with the International Conference on Multimedia Modeling (MMM), started in 2012, and has since then been a yearly fixture at the conference[4]. VBS 2019–21 used the V3C1 [BRS+19], and VBS 2022 additionally the V3C2 [RSB21] dataset, which are shards of the V3C [RSA+19] dataset consisting of videos scraped from Vimeo. It has three task types: V-KIS and T-KIS, as described previously, and Ad-Hoc Video Search (AVS), where multiple correct submissions can be made.

The scoring functions for those tasks have changed over the years. Fundamentally, the intent for KIS tasks is to reward quickly finding the correct item while punishing wrong submissions. In AVS tasks, the goal is to reward both precision and recall. We will recap the scoring function used in 2021 and 2022[5] for AVS tasks: Given correct submissions c and incorrect submissions i of a team, all correct submissions of all teams for a task p and a quantization function QUANT which merges temporally close correct shots into ranges[6], the scoring function for AVS tasks is as follows:

$$f_{avs}\,(c,i,p) \mapsto \frac{100 \cdot \|c\|}{\|c\| + \frac{\|i\|}{2}} \cdot \frac{\|\textsc{quant}(c)\|}{\|\textsc{quant}(p)\|}$$

We show a tabular overview of the results of VBS from 2019–2022 in Table 6.2. For reading clarity, we have sometimes named systems by the same research group with similar approaches consistently over the years.

Immediately noticeable is the increase in the number of participants. Some participants have been present over multiple years, such as VISIONE [ABC+22], VIRET [LKS+19b; LBS+21], and VIREO [NWN+20]. The placement of teams also varies significantly over the years, which indicates that the field progresses fast, and the adaption and invention of new features is essential to keep up with state-of-the-art systems. The analysis papers of the 2020 and 2021 iterations showed that enabling users to express temporal context is a crucial feature of top performing systems [LVM+21; HGB+22]. When specifically considering vitrivr, the success in 2019 can be mainly attributed to the inclusion of various deep learning features for OCR, ASR, and concept re-

[4]The analysis of VBS 2021 was led during this dissertation project, and this subsection draws significantly on it [HGB+22]. For the sake of readability, we will not use quotation marks explicitly when quoting from our own journal paper in this subsection.

[5]For 2023, a new scoring function is used that only awards points for one correct submission per video.

[6]"since VBS 2018, ranges are fixed static non-overlapping segments of 180s duration" [Sch21b], in 2021 the ranges were dynamic and based on shot segmentation.

Table 6.2 VBS result overview from 2019–2022

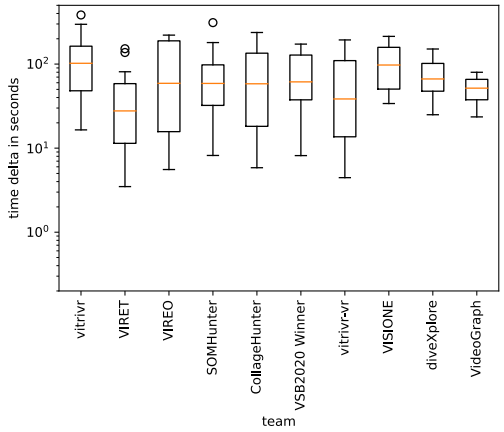
2019	2020	2021	2022
vitrivr	SOMHunter	**vitrivr**	HTW
VIRET	VIRET	VIRET	VIRET
VIREO	**vitrivr**	VIREO	VISIONE
VISIONE	VIREO	SOMHunter	IVIST
ITEC	Exquisitor	HTW	AVSEEKER
-	IVIST	VIRET	HCMUS-FIRST
-	AAU	VERGE	VideoFall
-	ITEC	VBS20 Winner	VERGE
-	VERGE	vitrivr-VR	**vitrivr**
-	VNU	Exquisitor	VNUHCM
-	-	VISIONE	VIREO
-	-	diveXplore	UIT
-	-	VideoGraph	vitrivr-VR
-	-	noshot	diveXplore
-	-	IVIST	Exquisitor
-	-	EOLAS	ViRMA

trieval [**rossettoDeepLearningBasedConcept2019notes**; RPG+19; RGL+21]. At VBS 2020, the novel textual embedding of the top two performing systems proved crucial to their success [LVM+21], with the 2020 iteration of vitrivr not yet integrating such an embedding and thus placing third. The new textual embedding specifically developed for vitrivr and vitrivr-VR in 2021 [SGH+21a], together with a new and improved temporal search functionality [HSS+20; Gst21; HAG+22], and a strong AVS performance, resulted in vitrivr achieving first place again [HGB+22]. The midfield performance of vitrivr in 2022 can most probably be attributed to the fact that most top performing systems used a version of CLIP [RKH+21], which vitrivr only integrated later that year, and operator performance[7].

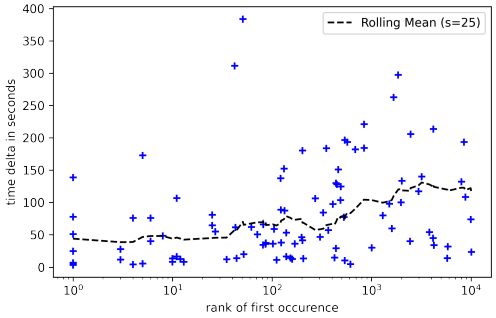
One advantage of the format is that it enables an analysis of the result logs. In addition to the submissions, most teams logged the result sets of their queries, either storing the logs locally or sending them directly to the competition server. In [HGB+22], we have shown what insights can be drawn from analysing logs and submissions and will show a few interesting highlights here.

One interesting question is how long it took operators to find an item once it was present in a result set. This is dependent on the system (i.e., how good the

[7] A preliminary analysis of result log data shows a significant number of browsing failures for vitrivr, indicating that while the retrieval model worked, the operators (including the author of this thesis) missed the correct item during browsing.



(a) Time deltas of teams between first appearance of correct item in result logs and submission



(b) Relation between rank and submission delta

Figure 6.2 Relation between the rank of first occurrence of a shot in the result logs and time delta to correct submission at VBS 21. As expected, time delta increases with rank, with variance increasing as well [HGB+22]

browsing capabilities of a system are), and the operator, since some operators prefer to browse a result set exhaustively, while others prefer to reformulate and execute new queries. Figure 6.2a shows the elapsed time between the first appearance of the correct shot in the result set and submission time of the correct item. Note that it is possible that a correct item was found through the video and not the shot. To visualize the dependency between the rank of a found item and the time until correct submission, we show in Figure 6.2b each correct submission as a datapoint with the rank it was found at first, and the time it took until correct submission. Overall, the figure shows that, as expected, the time between the first appearance and a correct submission increases with the rank. However, the figure also demonstrates that variance increases as well, indicating that operator differences are indeed occurring. While some operators might have browsed for a long time, others reformulated their query or found the correct item through the correct video. The two plots show that while there is a relation between the rank at which a correct item is found, and the time it is submitted, the effect is relatively weak and operator efficiency is crucial with browsing misses (that is, the correct item is visible at a low rank but not submitted) relatively common.

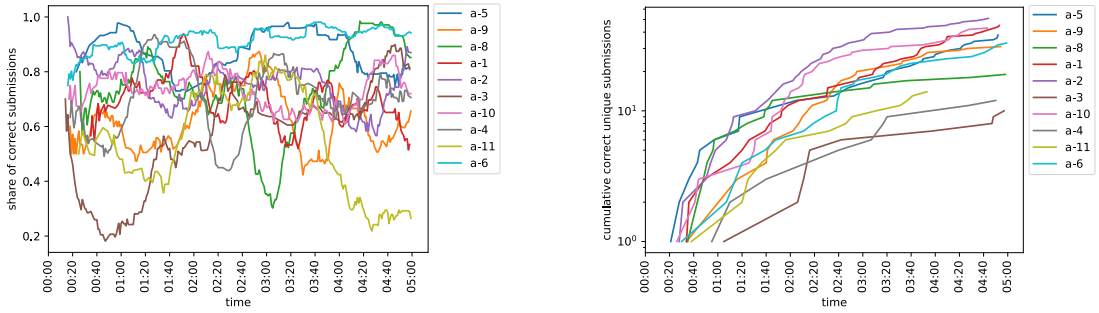
In the 2019 and 2020 iterations of VBS, there was no analysis of AVS tasks due to technical issues [Sch19; RGL+21; LVM+21]. At VBS 2021, the new evaluation server [RGS+21] improved testing by teams before the competition, which helped improve data quality. This also meant that we could analyze research

Table 6.3 List of AVS tasks for VBS 2021 with their description, ordered by appearance order in the competition (a-5 was solved first, a-6 last)

Task ID	Task Description
a-5	Find shots of a person holding or waving a flag.
a-9	Find shots of at least one person drinking beer.
a-8	Find shots inside an airplane, showing at least one passenger.
a-1	Find outdoor shots of two women walking and talking to each other.
a-2	Find shots of people having their hair done.
a-3	Find shots of a person skiing, with his/her own skis in the picture.
a-10	Find shots of two adult men hugging each other.
a-4	Find shots of kids playing football (soccer).
a-11	Find shots of people skiing, shot with the camera looking into the sun (backlit shot, possibly with lens flare).
a-6	Find underwater shots of one or more fish.

questions around AVS tasks, for which both retrieval and judgement is done interactively at VBS. Table 6.3 shows all AVS tasks of VBS 2021 and their description in the order in which they were solved in the competition. All plots going forward include the task identifiers.

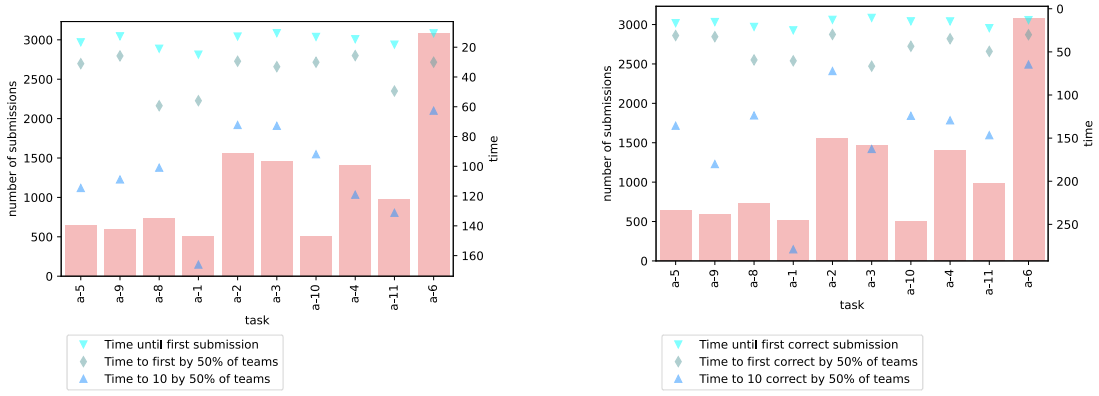
One area of interest is how the assessed correctness of submissions changes during the time allocated to a task. The hypothesis being that at the start of a task, there is some ambiguity between the task description and judge and operator understanding of the description, which is improved as teams see in real time thumbnails correct or incorrect submissions. Figure 6.3a shows the ratio of submissions judged as correct over time. What stands out is that there were two tasks with a large degree of difference in task understanding, a-3 (person skiing with their own skis in the picture) and a-11 (person skiing, camera looking into the sun). For a-3, the difference (the task intention was for point-of-view shots) was clarified with a comment from a judge, however the ratio remains low since



(a) Share of AVS submissions judged as correct over time during an AVS task

(b) Cumulative unique correct video submissions over time during an AVS task

Figure 6.3 AVS submission and judgment statistics [HGB+22]



(a) All submissions

(b) Submissions judged as correct

Figure 6.4 Selected AVS metrics per task, looking at all and only correct submissions. Higher y-axis values indicate that for a given task, it is easier to find results [HGB+22]

not all teams followed the discussion. For a-11 the different understandings persisted. Overall, no clear trend emerges. Some tasks exhibit consistently high agreement (e.g., a-6 looking for fish underwater, and a-5 person with a flag), while most tasks have a high variance during the task. Figure 6.3b shows that the number of unique correct videos that are found per task continues to increase even toward the end of the task, showing that even at the end of the time limit, new videos matching the description are still being found. This indicates that given a longer task duration, the number of unique correct submissions would probably still increase, as long as relevant segments exist in the collection.

Another interesting question is what differences, if any, there are between AVS tasks. For some tasks, looking at a thumbnail is sufficient (e.g., underwater shot of fish), while for tasks describing an action, the video needs to be inspected (e.g., shots of two women walking and talking). Additionally, some tasks might have a wide range of acceptable results, while others are quite narrow. Fig-

ure 6.4a, with all submissions, and Figure 6.4b, with only correct submissions, show the difference between the AVS tasks in terms of selected metrics: the number of overall submissions (shown as bars), time until first (correct) submission, time to first (correct) submission by half the teams, and time until first ten (correct) submissions by half the teams. The y-axis indicating the time, on the right, has been inverted so that higher y-axis values indicate that a task is easier for all metrics. On the x-axis, tasks are ordered by their appearance in the competition. Looking at these three graphs, the data indicates that there are relevant differences between the AVS tasks. For example, in the task a-1, it took almost five minutes for half of the teams to find 10 submissions which were judged as correct.

This analysis led improvements in evaluation methodology like a pre-event briefing of judges at VBS 2022 and 2023, and the methods are re-used for the 2022 analysis [LAB+23]. In particular, there is a focus on ensuring that both T-KIS and AVS tasks are clearly formulated, and ambiguity is reduced. Additionally, edge cases for judgment calls are discussed beforehand to reduce significant differences between the judges.

In general, the evaluation challenges noted in [Fer17; RGH+21] are visible in the analysis of VBS over the years [LVM+21; RGL+21; HGB+22]. Performance of human operators varies greatly, with the format not necessarily collecting enough data to make robust statements about the comparison of systems. Almost all systems rely on "countless opaque parameters and configurations" [RGH+21], and many systems are meta-evaluated before the competition to be improved, making "key observation and motivation for a specific configuration of the system irreproducible" [RGH+21].

The challenge that systems are meta-evaluated before the competition also partially applies to vitrivr. Although the contributions made in the course of VBS were aimed at general purpose video retrieval, some functionality has been primarily motivated by or geared toward an interactive evaluation setting. To name a few, functionality for cooperative retrieval and quick submission from thumbnails for AVS tasks [**rossettoDeepLearningBasedConcept2019notes**], specific UI views [RPG+19], and the move toward a new database system with improved single-node performance [SPG+20; GRH+20].

Other contributions made in the context of VBS have been motivated more broadly by the journey toward a general purpose multimedia retrieval system and not necessarily geared toward a competitive setting, such as the inclusion of novel deep learning methods [RPG+19], tighter integration of retrieval mod-

Table 6.4 LSC result overview from 2019–2022

2019	2020	2021	2022
vitrivr	MySceal	MySceal	MySceal
VIRET	SOMHunter	SOMHunter	LIFESEEKER
HCMUS-FIRST	**vitrivr**	LIFESEEKER	Memento
LIFESEEKER	VIRET	Voxento	HCMUS-FIRST
THUIR	Exquisitor	VIRET	Voxento
Exquisitor	diveXplore	Memento	**vitrivr**
ITEC	ViRLE	HCMUS-FIRST	diveXplore
NTU	LIFESEEKER	NTU	vitrivr-VR
LENS	HCMUS-FIRST	diveXplore	MEMORIA
-	VideoGraph	LifeMon	-
-	THUIR	**vitrivr**	-
-	NTU	vitrivr-VR	-
-	BIDAL-HCMUS	Exquisitor	-
-	DCU Vox	XQC	-
-	-	PhotoCube	-
-	-	ViRMA	-
-	-	VideoGraph	-

els [SPG+20], exploring explainability [HGI+21], or novel query methods like pose-based queries [HAG+22].

6.1.3 Lifelog Search Challenge (LSC)

The Lifelog Search Challenge (LSC) is "modelled on the successful Video Browser Showdown (VBS)" [GSJ+18] and has as the underlying dataset multimodal data captured by a single lifelogger over the years. The exact dataset which is used for LSC has differed but is consistently "a multimodal lifelog dataset gathered by one active wearer (lifelogger)" [GJS+22]. It has over the years included various metadata such as location, music listening history, and biometric data. Anonymized images taken by a wearable camera are central to the dataset and the challenge, as the queries often describe visual context. LSC is collocated with the ICMR conference yearly and has been the Grand Challenge of ICMR 2022. For task types, LSC includes T-KIS and AVS tasks similar to VBS and has experimented with a Q&A task in 2022 [HRS+22].

For a direct comparison, the main difference between LSC and VBS lies in the task types, with LSC having no V-KIS tasks and a Q&A task in 2022, and the dataset. Otherwise, the format and scoring function is the same.

We show a tabular overview of the results of LSC from 2019–2022 in Table 6.4.

For reading clarity, we have sometimes named systems by the same research group with similar approaches consistently and matched them with the VBS names.

Comparing the result overview with the one from VBS, we can see that the top-performing teams (especially in 2021 and 2022) are more similar, indicating a larger performance gap. We attribute this to the specialization involved with the top performing systems (e.g., MySceal [TNN+22], LIFESEEKER [NLN+22], Memento [AGG22]), which are built specifically for lifelog retrieval. Including specialized features like spatial aggregation of metadata, sometimes with handcrafted mappings, requires significant engineering effort, and the specialization also means operators are intimately familiar with the datasets and the setting. Due to the COVID-19 pandemic, LSC 2020–2022 was run in hybrid/online mode, meaning novice sessions were not included, and thus the performance of the one allowed operator is crucial for high scores, and further benefits expert systems.

To contextualize the results of vitrivr, the deep learning functionality integrated for VBS 2019 also proved essential in achieving the highest score at LSC 2019, and in 2020 a specialized Lifelog System and SOMHunter with an improved textual embedding proved the benefit of specialization and textual embeddings. The results of LSC 2021 showed that the embedding used by vitrivr was not ideal for lifelog retrieval, and systems incorporating CLIP or specializing in Lifelog Retrieval (e.g., MySceal, LifeSeeker) achieved a higher score [TND+23]. Even though CLIP was added for the 2022 iteration, systems specialized in Lifelog retrieval (and also optimized specifically for LSC) still achieved higher scores than vitrivr, indicating that the general-purpose approach in the implementation has its limit and that future work on result presentation and summarization in the context of lifelog retrieval and beyond may be a worthwhile endeavor both on a conceptual and system level.

LSC has acted as a driver for vitrivr to further move toward a general purpose retrieval system, with additions for Boolean retrieval in both retrieval model and UI, and late filtering functionality for metadata [RGH+19], or experimentation with image stabilization and addition of geo-spatial queries [HGP+21].

6.1.4 Four Years of Interactive Retrieval Evaluation Campaigns

To wrap up this section, we will discuss subjective experiences and impressions, sometimes backed by data, gained during four years of interactive retrieval evaluation campaigns and provide a critical look toward these evaluation

campaigns. As discussed previously, the format of synchronously comparing retrieval systems at a conference (e.g., VBS, LSC) or in a dedicated setting [RGH+21] has significant advantages. These settings have been described as "equitable" [LBB+22], which is true to some degree, yet there are challenges which also have been highlighted by reviewers and in (co-)authored works [RGH+21; HGB+22; HSS23] which we wish to discuss briefly.

Clarity about Goals & Methods: In its current form, the focus and methods of the analysis papers is determined after the competition. While this has some benefits, best practice in some fields of natural science include preregistration of analysis goals and methods [NED+18].

One example of this is the question of interaction and result logs. While there are significant challenges with normalizing a variety of user interfaces and conceptual approaches to a format which can be analyzed, these are not unsolvable problems. Since both format and extent of required logging is often communicated only very close to the competition and not enforced through automated testing, analysis papers only have access to logs from a subset of teams, which require a significant post-hoc normalization effort. Additionally, it means results are subject to publication bias [Sut09].

While there is a balance to be struck between barrier to entry for new teams and extensive data collection from participants, the current format and organization could benefit from a clearer communication of the goals of these campaigns and the methods which are used to achieve these goals.

Barrier to Entry: Looking at the results of previous years, the top teams often come from the same research groups. On one hand, this makes sense as building a performant and user-friendly system is not a trivial endeavor, and teams benefit from work done and experience gained. On the other hand, this means individual aspects or improvements are harder to identify as relevant, and new participants require a significant effort to become competitive.

Efforts have been made to reduce the barrier to entry, such as in the VBS context open-sourcing the data used by vitrivr in 2019 [RPG+19] and extracted data for the dataset used in 2022 [RSB21], and in the LSC context providing output of the Microsoft vision API for the entire dataset. These efforts are commendable, but more work remains to be done. vitrivr has been between 2019–2022 the only fully open-source participant at both competitions, and a limited number of participants have released snapshots of their code in separate publications.

A commitment to open code would aid new participants and existing participants in understanding the precise nature of the approaches used by other researchers, as not every implementation detail can fit into a 6-page ACM double-column paper.

Robustness of Results: As shown in [RGH+21], inter-user performance differences can be significant, even for expert users of the same system. In its current format, both VBS and LSC only allow a limited number of participants (2 resp. 1), which limits the robustness of the results of these evaluation campaigns.

This is related to the fact that they physically take place at a conference, which poses organizational challenges, but advances in tooling [RGS+21] has enabled remote and hybrid participation, which would allow more users per system.

Fully Fair Setting: Multiple areas can be identified where the current setting does not offer a fully fair evaluation. While these areas involve tradeoffs and are not easily solvable, it is nevertheless important to briefly mention them here. First, the textual presentation modality for T-KIS tasks means that participants which do not share a similar cultural background as the person creating the queries can be at a disadvantage as they do not comprehend references or might describe a scene differently. Second, there are commonly differences between expert users in terms of used hardware or time spent practicing with their system.

The Nature of a High Score: Related to the first point, participants and external reviewers have different outlooks on what a high / the highest score at such a competition and benchmarks in general mean. It starts with the basic question of calling the team which has achieved the highest score "winner". On one hand, participants invest significant efforts in their concepts and implementation, and in a competitive setting it is entirely appropriate to term the highest-scoring participant "winner". On the other hand[8], among the many aims of science one could reasonably formulate, none of them include winning against other researchers. If the aim is to further human understanding, science should be a collaborative endeavor and rewarding high-achieving teams incentivizes teams to focus on their own score instead

[8] As noted by multiple reviewers in different contexts.

of advancing the community as a whole[9]. The current practice of giving the lead of the common journal publication to the highest-scoring team also further incentivizes a focus on one's own score.

6.2 System-Centered Evaluation

Turning to a more standard way of evaluating contributions, we will evaluate our model and implementation in this section using a newly created dataset with appropriate metrics, and associated significance tests.

We focus on the scenario described in Chapter 2, which is mapped to a task occurring in benchmarking campaigns [LBB+22], which is that of "retrieving one particular data item which satisfies a specific information need for a user (i.e., a KIS task)" [LBB+22]. The results shown in this evaluation serve two purposes. On one hand, we can make recommendations about algorithm selection and gain insights for our scenario, on the other hand we offer a blueprint for future work aiming to further drive progress in this area. Following [Hul93; BBF+07; BR11] we briefly list requirements from literature for retrieval experiments:

Test Collection: Any test collection should contain the data itself, tasks for the collection and a ground truth containing correct answers[10]. We will describe the dataset used in this evaluation in Section 6.2.1.

Evaluation Measures: The effectiveness of the used system or method needs to be quantified using suitable metrics. We describe the metrics used in our evaluation in Section 6.2.2.

Significance: There should be a statistical methodology which determines whether the differences between the methods are statistically significant. We outline our significance tests in Section 6.2.3, and report results for the metrics, with additional information in the appendix.

Work in this section has benefited from supervised student theses [Gst21; Ben22a; Ben22b] and is performed using a separate evaluation client which emulates the functionality of the retrieval engine and in which model and algo-

[9] As a simple example, consider incentives around sharing data extracted from a paid API such as the Google Vision API. Allowing teams to use paid commercial APIs without requiring them to share such data could be construed as a form of pay-to-"win".

[10] In a KIS scenario, the relevance for each item except the desired one is 0 for any query formulated for a specific task. Outside the KIS scenario, the item might be relevant for the query.

rithms are implemented. We now start with a description of the dataset, metrics and significance method and then show evaluation results.

6.2.1 Dataset

As existing datasets for multimedia retrieval evaluation are focused on a queries without temporal context, we constructed our own evaluation dataset based on the VBS and LSC evaluations. Our argument for a distinct reference collection which is focused on a particular type of information need is also in line with arguments from standard literature [BR11, p. 134]. Following the requirements described for a test collection, we describe our datasets.

Multimedia Data For the multimedia data, we use V3C1 [BRS+19] for the video retrieval evaluation and the dataset from LSC 2020 & 2021 [GLN+20; GJS+21] for the lifelog retrieval evaluation. V3C1 consists of a wide range of videos which were collected from Vimeo[11], and the lifelog dataset consists of four months of multimodal lifelog data including approx. 180'000 images, location logs and biometrics.

Task Data As the datasets are used at benchmarking campaigns, we also have tasks associated with them from the KIS parts of these campaigns. This means that there are tasks which are relevant to the dataset and have been selected and designed by independent actors.

Queries The queries for these tasks have been collected from a variety of users. For the video retrieval scenario, we have asked users to describe video clips of the defined tasks in plain text. Details on the tasks and prompt are available in Appendix B. For the lifelog scenarios, the queries have been created by the author and in a student project [Ben22a]. [CPK+08] argues that "more queries with fewer or noisier judgments is preferable to evaluation over fewer queries with more judgments". We follow their argument and generate new, artificial queries based on the user-provided queries. Specifically, we simulate users leaving out one or more query elements in their descriptions as described in Example 6.1.

[11] `https://vimeo.com`

Example 6.1 Query Generation

Assuming we have a query with three subqueries: "giraffe", "lion", and "elephants" with user-specified distances of 5 and 10 seconds. We generate the following new, additional, queries:

- "giraffe" → "lion" with a distance of 5 seconds
- "giraffe" → "elephants" with a distance of 15 seconds
- "lion" → "elephants" with a distance of 10 seconds

This simulates users leaving out queries, but adds noise as the duration of the subquery which is removed is set to 0. We still think this is preferable following the arguments of [CPK+08].

To give an overview of the datasets, Table 6.5 shows the number of tasks, queries and how many queries there are in total after applying query expansion.

Table 6.5 Tabular overview of task data

Dataset	Tasks	Queries	Expanded Queries
Lifelog	16	17	136
Video	69	143	3904

6.2.2 Metrics

In this subsection, we describe the metrics we use to measure retrieval performance. We use the best rank as a metric for individual queries and hit@k as a summary metric.

Metric I: Best Rank

The best rank of a correct item is chosen because in the KIS scenario, a user is satisfied after the first correct item, Example 6.2 illustrates this metric.

Example 6.2 Best Rank

Assuming we are looking for an element in the sequence between 00:20–00:40 in video 1, and we are provided the following result list:

1. Video 2, 01:30–01:35

2. Video 1, 00:15–00:25 (correct)

3. Video 5, 02:40–03:10

4. Video 1, 00:30–00:40 (correct)

As the results at rank 2 and 4 are correct, the best correct rank is 2.

Metric II: hit@k

hit@k indicates the percentage of correct answers at or below a certain threshold and is defined in Definition 6.1. It is used in evaluations of other retrieval systems and methods [AGG21; AGG22; WCM+19; Mof13; TNN+22; HK92; KKO+92; CWZ+11; SSX+16; SBH+16; YLS+16] and in machine learning evaluations [FCS+13; NMB+14; IM18].

Definition 6.1 hit@k

Given a ranked list containing the results for a query r, $f_c(r)$ which returns the position of the first correct answer in r, and a threshold k, we define a helper function $h(r,k)$. This indicates if for a given result set, the correct item was found at or below the given threshold:

$$h(r,k) = \begin{cases} 1 & \text{if } f_c(r) \leq k \\ 0 & \text{otherwise} \end{cases}$$

Then, given the set of all evaluation queries Q_e and the set of results for all queries $\overline{r} = (r_1, r_2, \ldots, r_n)$, hit@k is defined as follows:

$$\text{hit}(Q_e,k) = \frac{\sum_{i=1}^{\|Q_e\|} h(r_i,k)}{\|Q_e\|}$$

As both of those metrics rely on a fair comparison of ranks, we make two adjustments. Firstly, algorithms get a configurable maximum execution time of 10 seconds. If this is exceeded, execution is terminated. Secondly, to avoid known issues when comparing top-k lists, best ranks are capped at 10'000 and if computation time is exceeded or an item is not found, its value is simply set to 10'000 + 1. We illustrate the need for these adjustments in Example 6.3.

Example 6.3 Creating Boxplots with unequal n

Moving into the world of visualization, we can showcase the need for having equal sample size with categorical boxplots through a simple example. This example is also relevant for other methods which rely on a comparison of median values.
Let us consider two systems A and B which are tasked with retrieving the top k elements for a query. Given four tasks and $k = 10$, the correct item is found at the following ranks:

Task	A	B
1	1	1
2	-	2
3	3	3
4	-	4

System A now has a median best rank of 2, while system B has a median best rank of 2.5 — making system B appear worse in the boxplot, even though it is clear that it is preferable. In the most favorable scenario, A would have found the desired item at ranks $k + 1 = 11$, which would mean A has a best-case median of 7.

6.2.3 Significance

To briefly summarize, significance tests aim to disprove the null hypothesis, which in our case is that there is no difference between two methods. This rejection would imply that there is indeed a difference, with either of the two methods outperforming the other.

In our work, we will use the *paired sign test* [DM46; Sie57], which compares not the magnitude of difference between two methods for a query, but only which method performed better. We discard other tests because they make significant assumptions about the distribution of measurements and differences, as argued by [van79; BBF+07]. As a simple example, if method A has a best rank of 1 for a query, and method B has a best rank of 2, the sign is -1. This would also be the case for 1 vs 500.

The statistical methods we use for significance tests are taken from [BBF+07, p. 357–359]. The implementation of the significance tests has partially been done in the context of a supervised student project [Ben22b].

Specifically, given two methods A and B we compute the sign s_i of each measurement as follows:

$$s_i = \begin{cases} -1 & \text{if A outperforms B on measurement } i \\ 0 & \text{if both methods are equal on measurement } i \\ 1 & \text{if B outperforms A on measurement } i \end{cases}$$

Afterwards, we compute the number of occurrences c_1, c_{-1} where each method outperforms the other. Given a function which checks the sign $f : \{1,-1\} \times \{1,-1\} \rightarrow \{0,1\}$, with $f(s_1, s_2) = 1$ if $s_1 = s_2$ and $f(s_1, s_2) = 0$ otherwise, we compute c_1, c_{-1} as follows:

$$c_k = \sum_{i=1}^{\|Q_e\|} f(k, s_i)$$

Following [BBF+07] and given $k = \min(c_1, c_{-1})$ and $n = c_1 + c_{-1}$ (the number of queries where the results are not equal)[12], we then compute the p-value as follows:

$$p = 2 * \sum_{j=1}^{k} \frac{n!}{j!(n-j)!} 0.5^j (1 - 0.5)^{n-j}$$

Finally, we compare the calculated p-value to a pre-determined significance level. If p is below the significance level, we reject the assumption that there is no difference between the two methods, which means we have a statistically significant difference.

We indicate p-values lower than 0.001 as 0.001 following APA statistic guidelines [Ass22].

6.2.4 Results: Retrieval Quality

Having introduced our dataset, metrics and significance method, we will now first show results for retrieval quality and then afterwards turn to retrieval runtime. We focus on the quality of algorithms and underlying retrieval features. For every metric and visualization, we show results for video and lifelog retrieval.

When comparing algorithms, we compare the following based on the presented concepts in Chapter 4:

[12] [BBF+07] does not discuss ties, the original [DM46] suggests incrementing both c_1 and c_{-1} by half of the ties, common reference works suggest decreasing n by the number of ties [SS99; Spr11].

Do Not Consider Temporal Context: Two algorithms which do not consider temporal context but instead just aggregate scores for a single segment using either average- or maxpooling called Average Segment Scoring Algorithm (AVGSSA) and Maximum Segment Scoring Algorithm (MAXSSA). These served as the basis for previous iterations of vitrivr [Ros18].

Algorithms From Literature: For both VIBRO [HSJ+22] and VISIONE [ABC+22], we have received enough information from the original authors to reconstruct the functionality as discussed in Chapter 4. We use those two algorithms also as a baseline, with the expectation that they perform better in the video scenario than in the lifelog scenario as this is what they were originally designed for.

Our Modular Algorithm: Our baseline algorithm uses for pre-aggregation, candidate generation and post-aggregation the mechanisms described in Chapter 4. There is a strict time cutoff to consider candidate segments. We term this algorithm SIMPLE as it uses the default configuration of our suggested model.

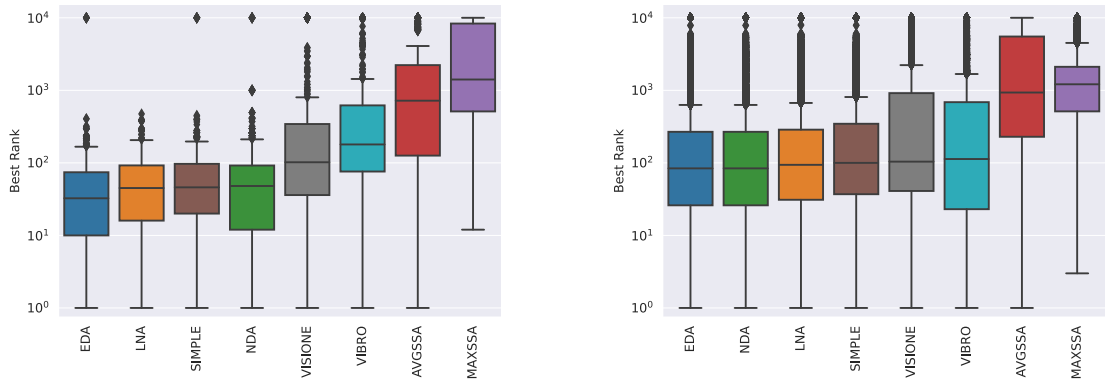
Distribution Algorithms: As introduced in Section 4.4.3, these algorithms reward candidate segments which are a close match to the user-specified distance while not ignoring those who are not a match. These are called NDA, EDA, and LNA and use the distributions discussed previously in Section 4.4.

For the underlying retrieval features, we compare two textual embeddings, CLIP [RKH+21] and the visual text co-embedding which vitrivr and vitrivr-VR have used at previous benchmarking campaigns [SGH+21a; SS22]. We abbreviate them as CLIP and VTE respectively.

Best Rank

Figure 6.5 shows the best rank of a correct item over all tasks per algorithm. As mentioned, all ranks are capped at 10'000, and if the algorithm did not find the correct item or exceeded computation time, its rank is also set to 10'000+ 1. Algorithms are ordered by median best rank. Looking at the results for both video and lifelog retrieval, it is evident that there is a clear benefit for algorithms which consider temporal context. For both datasets, the algorithms which do not consider temporal context perform worst (AVGSSA and MAXSSA).

For the lifelog scenario, the two algorithms from literature designed for video retrieval struggle as expected even though we tried to have a fair parameter map-



(a) Lifelog (b) Video

Figure 6.5 Comparison of the best rank of a desired item over all tasks

Table 6.6 Tabular overview of best rank significance results for all Lifelog queries

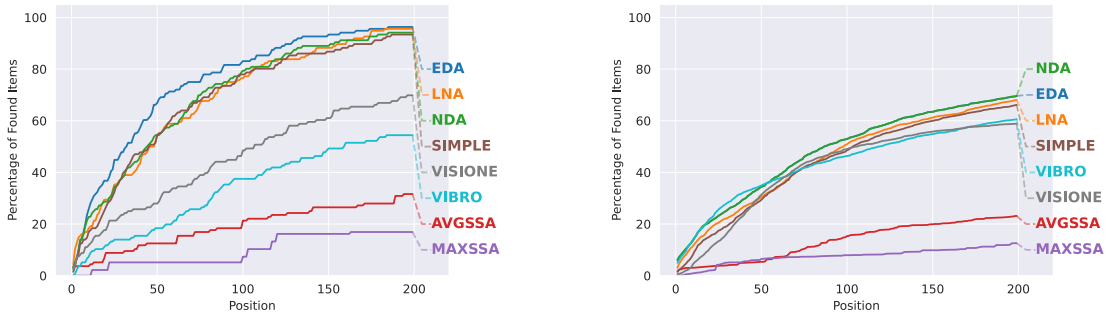
	EDA	LNA	SIMPLE	NDA	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		**0.036**	0.181	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
LNA			0.729	0.543	**0.001**	**0.001**	**0.001**	**0.001**
SIMPLE				0.059	0.085	**0.001**	**0.001**	**0.001**
NDA					**0.016**	**0.001**	**0.001**	**0.001**
VISIONE						**0.004**	**0.001**	**0.001**
VIBRO							**0.001**	**0.001**
AVGSSA								0.347
MAXSSA								

Table 6.7 Tabular overview of best rank significance results for all Video queries

	EDA	NDA	LNA	SIMPLE	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		**0.001**	**0.001**	**0.001**	0.55	**0.001**	**0.001**	**0.001**
NDA			**0.001**	**0.001**	0.55	**0.001**	**0.001**	**0.001**
LNA				**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
SIMPLE					**0.001**	**0.001**	**0.001**	**0.001**
VISIONE						**0.001**	**0.001**	**0.001**
VIBRO							**0.001**	**0.001**
AVGSSA								**0.001**
MAXSSA								

ping to the lifelog scenario. In both scenarios, the exponential decay algorithm EDA performs best. In the video scenario, there is little visible difference in the median of the boxplots between the four algorithms implemented in this thesis and the two from literature, which is why we turn to the significance tests.

We show the results from the significance tests in Tables 6.6 and 6.7, with



(a) Lifelog (b) Video

Figure 6.6 How many of the desired items were found at or below a certain position

statistically significant results ($p < 0.05$) highlighted in bold. Algorithms are ordered by median best rank on both axes. For example, we can see that there is a statistically significant difference between EDA and MAXSSA with $p = 0.001$ for both video and lifelog datasets.

In both scenarios, the differences between the algorithms which do not consider temporal context and the algorithms which do are reported as significant by the sign test. In the video retrieval scenario, there are enough queries to achieve significant differences between all pairings except for the VISIONE algorithm, while in the lifelog scenario, the differences between the SIMPLE and VISIONE algorithms and some differences between our four algorithms are not reported as significant.

This can indicate that more tasks and queries are needed, or that the uncertainty associated with information needs spanning longer periods of time make it difficult to design reward functions which work for all cases.

We show quality with regard to the number of subqueries in Appendix A.1 in Figures A.1 and A.2. Significance results for queries without query expansion can be found in Appendix A.3.1.

Cumulative HIT@k

Figure 6.6 shows hit@k on the y-axis for k up to 200. The reason for choosing a cutoff is the assumption that even expert users will rather re-formulate a query in the interactive setting than browsing an entire result set [HGB+22].

Both figures show that there are diminishing returns when continuing to explore results. It is also evident in both figures that any choice of temporal scoring algorithm has benefits compared to ranking mechanisms which do not consider temporal context. The results with the cutoff applied are relatively

Table 6.8 Tabular overview of hit@k significance results for all Lifelog queries, only looking at results with a best rank below 200

	EDA	LNA	SIMPLE	NDA	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		0.063	0.149	**0.002**	**0.001**	**0.001**	**0.001**	**0.001**
LNA			0.93	0.481	**0.001**	**0.001**	**0.001**	**0.001**
SIMPLE				0.069	0.115	**0.001**	**0.001**	**0.001**
NDA					**0.015**	**0.001**	**0.001**	**0.001**
VISIONE						**0.001**	**0.001**	**0.001**
VIBRO							**0.001**	**0.001**
AVGSSA								**0.002**
MAXSSA								

Table 6.9 Tabular overview of hit@k significance results for all Video queries, only looking at results with a best rank below 200

	EDA	NDA	LNA	SIMPLE	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		**0.005**	**0.001**	**0.001**	**0.001**	0.698	**0.001**	**0.001**
NDA			**0.001**	**0.001**	**0.001**	0.742	**0.001**	**0.001**
LNA				**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
SIMPLE					**0.001**	0.134	**0.001**	**0.001**
VISIONE						**0.021**	**0.001**	**0.001**
VIBRO							**0.001**	**0.001**
AVGSSA								**0.001**
MAXSSA								

consistent with those from the previous plots. It is evident that the gap between the VIBRO/VISIONE implementations and ours is larger in the lifelog context, which makes sense as those were not originally developed for lifelog retrieval and thus, are not adopted for the longer distances between subqueries and different information needs.

The difference between the VIBRO algorithm and our algorithms at very low ranks for video retrieval is interesting and could warrant further investigation. Preliminary analysis from VBS 2022 data indicates that one of the reasons contributing to VIBRO achieving the highest score was the very fast submission of correct results compared to other systems. This is consistent with the results shown here.

For significance results, we follow the same methodology as for the best rank but ignore all queries with a best rank below 200, following the hypothesis that users would ignore those. We show the results in Tables 6.8 and 6.9.

Again, the results are similar to those for comparing the best rank. In the

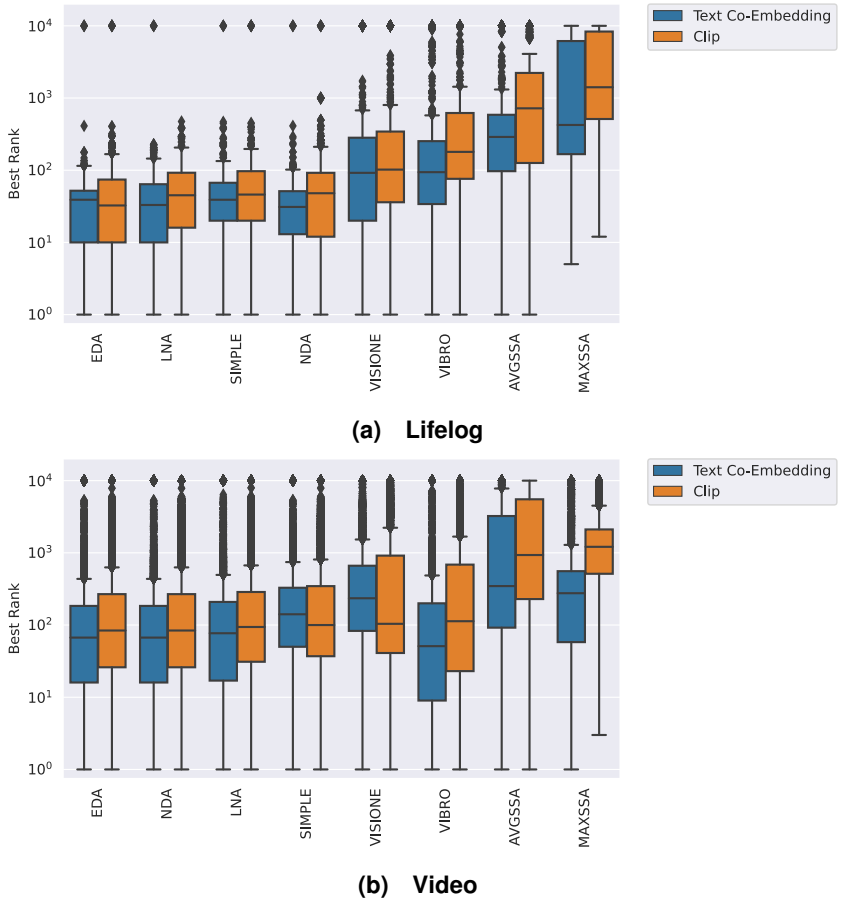


Figure 6.7 Comparing best rank for two different textual embedding features

lifelog scenario, more queries may help getting to more robust results, but a clear difference between our algorithms and those from literature or those not considering temporal context is visible. In the video retrieval scenario, our algorithms have significantly better results than the others with the exception of VIBRO.

Underlying Retrieval Features

We now turn our attention toward the importance of the underlying retrieval functionality for temporal queries. As argued previously, complex queries and fusion are useful when the individual components which are to be combined are of high quality. Figure 6.7 shows a comparison of two different textual embeddings, CLIP [RKH+21] and the text co-embedding which vitrivr and vitrivr-VR have used at VBS and LSC previously [SGH+21a; SS22].

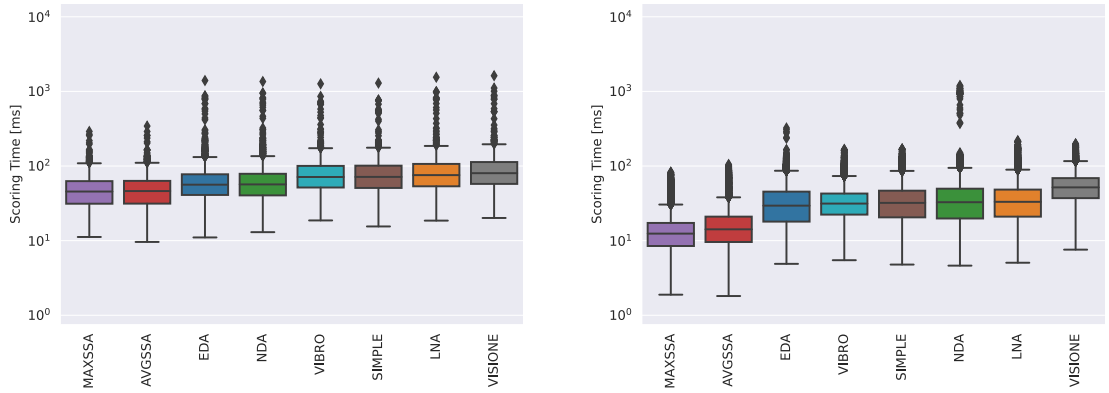
The results show that the visual-text co-embedding vitrivr uses performs better than the CLIP model for most algorithms in both the lifelog and video retrieval scenario. One reason for this discrepancy could be that VTE operates

Table 6.10 Lifelog: significance results when comparing CLIP and VTE per algorithm

Method	EDA	LNA	SIMPLE	NDA	VISIONE	VIBRO	AVGSSA	MAXSSA
CLIP v VTE	0.335	**0.006**	0.258	**0.001**	0.861	**0.001**	**0.001**	**0.001**

Table 6.11 Video: significance results when comparing CLIP and VTE per algorithm

Method	EDA	NDA	LNA	SIMPLE	VISIONE	VIBRO	AVGSSA	MAXSSA
CLIP v VTE	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**



(a) Lifelog **(b) Video**

Figure 6.8 Comparison of the algorithm runtime over all tasks

on the entire shot and temporally aggregates frames, while the implementation of CLIP in vitrivr operates on a single frame. However, given that the difference is not present in all algorithms, no clear recommendation emerges, indicating that depending on the algorithm, a different ranking of the textual query works better. This points toward result fusion of the two features being an attractive option.

We show significance results for a comparison of CLIP and VTE per algorithm in Tables 6.10 and 6.11. These indicate significant results across all algorithms for the video dataset and significant results for most algorithms on the lifelog dataset.

6.2.5 Results: Retrieval Runtime

Having focused of the quality of the algorithms and underlying retrieval functionality, we now turn our attention toward the execution time. Figure 6.8 shows the algorithm runtime over all tasks. We expect the algorithms which score seg-

ments individually (i.e., MAXSSA and AVGSSA) to be significantly faster. In both cases EDA is the fastest of the temporal scoring algorithms and LNA and VISIONE are the slowest. Combined with the quality results, EDA emerges as an attractive choice due to being both fast and accurate. More importantly, the results show that with median scoring times below a second for single-threaded execution on commodity hardware[13], late fusion for temporal scoring does not incur significant execution time costs when compared to the runtime of query transformation and execution time of the actual database queries [Gas23]. We show how runtime changes w.r.t the number of subqueries in Appendix A.2 in Figures A.3 and A.4.

Detailed results for the significance tests are in Appendix A.3.2, and show that the differences for video retrieval are robust and for lifelog retrieval mostly robust.

6.3 Discussion

Offering temporal search capabilities significantly enhances the performance of modern multimedia retrieval systems. While underlying retrieval features matter a great deal, the interplay of retrieval models and user interfaces, and specialization for expert users is of significant importance. Humans describe and remember with temporal context, and both system-centric and user-centric evaluations show the importance and benefits of temporal search. We recommend multimedia systems offer at least the option to specify multiple sequential terms, and combine results in a late fusion step which avoids showing duplicate results. For user-centric benchmarking competitions, we argue for clarity around goals and methods, and a concerted effort to make sure that the format suits those goals and methods.

User-centered evaluations have shown vitrivr to be a competitive system which is able to incorporate novel concepts, methods and ideas over a long timespan. The ability to express temporal information needs and evaluate them efficiently has played a key role in the success of vitrivr at these evaluations. In interactive evaluation campaigns, the incorporation of improved temporal query functionality together with a competitive textual embedding feature has proven key to achieving the highest overall score at VBS 2021 in addition to a strong AVS performance. We also show how result and submission logs can be leveraged to

[13] AMD EPYC 7302P for the video dataset, AMD Ryzen Threadripper 1950X for the lifelog dataset.

gain insights into the nature of AVS tasks and performance of retrieval systems. However, to ensure meaningful, robust and reproducible results, more participants and a clearer research design than currently used would be beneficial, as demonstrated in the analysis of multiple interactive evaluations (co-)authored during this dissertation project [RGH+21; HGB+22; HSS23].

The system-centered evaluation shows the benefits of the presented concepts for efficient video and lifelog retrieval. The model which considers the entire pipeline including sequence generation, candidate generation and scoring, and result generation enables a comparison of different approaches and that enabling users to specify more complex information needs with temporal components increases retrieval performance. It is evident that while there is a benefit to rewarding elements which match the user-specified sequence and distance closely, the best function to do so depends on the scenario and the users. In our evaluation, an algorithm which rewards segments matching user-specified distances through exponential decay modeling was best in terms of retrieval performance, but this might vary for other datasets and scenarios. Content-based retrieval features still dominate retrieval performance overall, which is consistent with expectations about fusion algorithms as discussed in Chapter 3. As so often, there is a tradeoff between speed and quality. In our case, the tradeoff is not severe enough to warrant considering dropping temporal late fusion, but it needs to be evaluated and kept in mind, especially for larger data and result sets.

From an implementation perspective, one limitation of the approach used by vitrivr is that the clear separation of concerns in its components means more integrated and focused approaches are not within scope, and thus vitrivr struggles versus specialized systems in evaluation campaigns such as LSC. The monolith nature of the retrieval engine, which also offers media segmentation and feature extraction, also means vitrivr requires more onboarding effort and collaboration than smaller, more focused systems. However, the improvements made by the years over all systems also showcase the need for a flexible and modular system which can adapt to new research methods and results. Having a stable system with proven stability and usefulness enables productivity gains for both core computer science and interdisciplinary research.

Eragon looked back at him, confused. "I don't understand." "Of course you don't," said Brom impatiently. "That's why I'm teaching you and not the other way around."

— Christopher Paolini, *Eragon [p. 148]*

7 Related Work

In this chapter, we will briefly cover relevant research related to this thesis which has not been mentioned in Chapter 3 or the literature discussions across the previous chapters. Different multimedia retrieval systems are discussed in Section 7.1, a discussion on how contributions in the area can be evaluated is found in Section 7.2, and we make a detour into the field of *Temporal Information Retrieval* in Section 7.3. For broader surveys on multimedia retrieval from the past decades, we happily refer to [SWS+00; DJL+08; HXL+11; Gia18; Gas23]. In addition to the direct references, content is also based on foundational texts in multimedia and information retrieval [BBF+07].

7.1 Multimedia Retrieval Systems

In this section, we give an overview of current trends and developments in systems for multimedia retrieval. For historical context, most early research on systems for information retrieval focused on text and document retrieval [Sch80; Bla88]. Self-contained multimedia retrieval systems started appearing in the 1990s, for example the QBIC system [FSN+95; NBE+93], which enables users to query using sketches or example images and had color, shape and texture features. Other notable examples include QVE [HK92; KKO+92], VRSS [CR95], Photobook [PPS96], Chabot [OS95], PicSOM [LKO02] or MindFinder [CWW+10]. A more recent example of an open-source system is LIRE[1] (Lucene Image REtrieval) [LRH+16], which builds on Apache Lucene [MHG10]. Similarly to Cineast, it supports feature extraction and retrieval with a number of retrieval features. However, it does not have a user interface and is limited to the visual

[1] `https://github.com/dermotte/LIRE`

domain, whereas vitrivr also handles multimedia such as 3D and audio.

Looking at more recent work, there are various other "retrieval systems that support cross-modal searches and multiple query types (e.g., [LXY+19; LSV+20; WNM+21; WN20])" [LBB+22], and for a recent comparison of retrieval systems and their underlying functionality, we look at the top performing participants of VBS 2021 in a tabular form (directly taken from [HGB+22]). The retrieval methods are shown in Table 7.1, and the interaction methods are shown in Table 7.2.

Looking at the tabular comparisons, it is clear that deep learning is also unavoidable in contemporary video retrieval. All top performing systems except HTW have a joint embedding, and HTW added one for its 2022 iteration [HSJ+22]. In terms of temporal functionality, all user interfaces enable browsing temporal context, but functionality and methodology for temporal queries is underdeveloped as shown in the literature discussions of Chapter 4. As similar picture emerges when looking at contemporary systems for lifelog retrieval [TND+23], where most systems feature embeddings and temporal queries but with simple approaches with underdefined methodology.

Additionally, lifelog retrieval has overlap with the field of human memory augmentation for which systems research is also done [KW07; HKB+09; BC17; CGR+20].

7.1.1 Interactive Retrieval

In our work, we have focused on KIS scenarios, where users have an information need which is satisfied by a single answer. As discussed in previous chapters, there are many more user models.

Where earlier work distinguished between information lookup and exploratory search, the latter covering both learning and investigating [Mar06] or differentiated between classic retrieval, dynamic interaction, browsing and recommendation [BN07], current work characterizes needs along an exploration-search axis [ZW14], which starts with browsing and moves over structuring, summarization, finding relevant items, KIS to ranking at the other end of the scale.

For perspectives on how users fulfill those needs, there are different models in literature. The classic notion has four steps: problem identification, information need articulation, query formulation and result evaluation [SE98; BR11].

This has both in theory and practice moved to more interactive or dynamic models [Rob00; BR11]. [SWS+00] defines an interactive query session as a sequence of query spaces, with each interaction of the user yielding a relevance

Table 7.1 Selected *search* approaches used by systems participating at VBS 2021. For each system, a reference to the paper describing the method is given; V3C1 means meta-data provided with the V3C1 dataset [BRS+19]. The ASR data for V3C1 was provided by [RPG+19]. Table from [HGB+22]

system	shot detection	joint embedding	concepts	ASR	OCR	image search	sketch search	fusion of modalities	temporal query	relevance feedback
vitrivr [HGI+21]	V3C1	[SGH+21a]	[RPG+19]	V3C1	[RPG+19]	[RGS14]	[RGS14]	[HSS+20]	[HSS+20]	
VIRET [PKS+21]	[LKS+19a; SL20]	[LSV+20; MKS20; RKH+21]				[LSV+20; MKS20]			[PKS+21]	
VIREO [WNM+21]	V3C1	[WN20]	[WN20]	V3C1	[Smi07]	[WN20]	[NLZ+18]	[WN20]	[WNM+21]	
SOMHunter [VML21]	[LKS+19a; SL20]	[LSV+20; MKS20]				[LSV+20; MKS20]			[LKS+19a]	[CMO+96]
HTW [HSJ+21]	[HSJ+21]					[HSJ+21]	[HSJ+21]	[HSJ+21]	[HSJ+21]	[HSJ+21]
CollageHunter [LBS+21]	[LKS+19a; SL20]	[LSV+20; MKS20]				[LSV+20; MKS20]			[LKS+19a]	[CMO+96]
VERGE [AMG+21]	V3C1	[GM20]	[MMG+18]			[PMM+17; JDS11]		[AMG+21]	[AMG+21]	
SOMHunter 2020 [KVM+20]	[LKS+19a; SL20]	[LSV+20; MKS20]				[LSV+20; MKS20]			[LKS+19a]	[CMO+96]
vitrivr-VR[SGH+21a]	V3C1	[SGH+21a]	[RPG+19]	V3C1	[RPG+19]	[RGS14]		[HSS+20]		
Exquisitor [KJL+21]	V3C1		[XGD+17]	(V3C1)					[KJL+21]	[KJR+20]
VISIONE [ABF+21]	V3C1	[MFE+21]	[AFG+17]			[MFE+21; RAR+19]	[ABC+21a]	[ABC+21a]	[ABF+21]	

Table 7.2 Selected *interaction* approaches used in systems participating at VBS 2021, with the ✓ symbol indicating implementation in a given system. Table from [HGB+22]

	top-k from video filter	temporal context	video preview	video summary	video player	2D map embedding
vitrivr	✓	✓	✓	✓	✓	
VIRET	✓	✓		✓		
VIREO	✓	✓	✓		✓	
SOMHunter	✓	✓		✓		✓
HTW	✓	✓	✓	✓	✓	✓
CollageHunter	✓	✓		✓		✓
VERGE		✓	✓		✓	
SOMHunter 2020	✓	✓		✓		✓
vitrivr-VR	✓	✓	✓	✓	✓	
Exquisitor	✓	✓			✓	
VISIONE		✓		✓	✓	

feedback, and the transition from one element to the next materializing the feedback of the user. To mention a few nature-related models, Bates [Bat89] describes information seeking as "berry-picking", where users require a series of pieces of information (berries) that they find along their ways scattered among the bushes. Pirolli [PC95] use a foraging analogy, where the *information scent* guides humans on their retrieval journey and humans adapt their seeking strategies to improve their searches, similar to the behavior of animals which look for food.

From a systems perspective, in addition to the one mentioned in the previous section, there are other "interactive and user-centric systems, where the query expressing the user's information need is no longer considered predetermined and static, but rather evolves dynamically during a search process [Chr07; WSd+06]" [LBB+22]. One particularly interesting example is the Exquisitor system, in which the entire retrieval model is centered around interactive learning and relevance feedback [KJR+20].

For a more in-depth overview on search models and strategies, we refer to [BR11, p. 22–25], [BN07], and [HGB+22, p. 4–8], and for query modification and relevance feedback to [BR11, Ch. 5].

7.1.2 User Interaction

Somewhat independent of the model of information seeking and information needs is the modality, which can range from traditional devices such as keyboard and mouse to mobile phones and more recently VR headsets or AR glasses.

Each modality requires its own way of expressing information needs, as they have unique capabilities and limitations. It is important to consider the impact of device properties such as screen size, interaction method (mouse, keyboard, touch, controller) or navigation possibilities (2D, 3D, AR, VR) on the user experience, as adjusting the interaction methods is essential for user satisfaction and effectiveness [SC06; SMW+13; KTS+17; MLK+18; DG20]. Looking at interactive learning and retrieval, research is also done on how to make use of mobile phones [CSC+07; BSK+21]. The best solution may also differ based on the user who is using the system, for example children may prefer different interfaces [LRL+10]. This is also one of the reasons why VBS sometimes incorporates novices in the evaluation campaign. This allows a comparison of systems for expert and novice users [LVM+21].

Another key element of user interaction is result visualization, where there are different definitions and categorizations in literature around result visualization. [SWS+00] very broadly a visualization operator which maps the query space into the display space D having a perceived dimension d. d is the inherent dimensionality of the result, which might need to be mapped onto the available dimensionality e.g., 2D for traditional desktop user interfaces, and 3D for VR. Boertjes and Nijholt [BN07] discuss matching presentation and content modality and differentiate between result presentation and visualization, the argument being that visualization uses "techniques [..] to interpret the data and [helps present] the data in a more understandable form" [BN07]. We do not use this separation in the following, as it is not often found in literature. [DJL+08] categorize four different presentation categories: objects can be ordered (by relevance or chronological), clustered (by either metadata or content), arranged hierarchical or some composite of those methods. In terms of contemporary systems with a traditional UI, SOMHunter [KVM+20] use a Self-Organizing Map (SOM) [Koh90], and HTW [HSJ+21] a hierarchical Self-Sorting Map (SSM) [SG14]. vitrivr-VR presents the result set in a sorted grid which is wrapped cylindrically around the user [SGH+21a], and has a in-video browsing mechanism "resembling a file cabinet drawer, which allows quickly riffling through a temporally ordered box containing the segments of a video" [HGB+22]. Another interesting visualization approach in VR is ViRMA [DJ22a; DJ22b], where the data is mapped to a three-dimensional space in which the user navigates.

7.2 Multimedia Retrieval System Evaluation

There are different dimensions along which evaluations can be categorized. We have used one prominent one, that of interactive or user-centric versus system-centric to structure Chapter 6, which is similar to the glass box vs black box categorization [Gro96], where glass box evaluations assess systematically components of a system and black box evaluations evaluate the system as a whole. In the context of our thesis, the interactive evaluation campaigns can be considered black box evaluations and the system-centric evaluation a glass box evaluation.

From a historical perspective, Text Retrieval Conferences (TREC) [VH05] is considered the first collaborative effort to both create test collections and evaluate methods and the format has been broadly adopted [BBF+07, p. 350–353]. VBS and LSC follow a similar approach, with a focus on the interactive aspect of the evaluation but represent only one of many options for a framework of evaluating interactive retrieval systems [Bor03; LBB+22]. There are numerous evaluation campaigns with their respective collections, consider for example [BR11, p. 158–165]. For a recent overview of current multimedia retrieval evaluation campaigns, we refer the reader to [LBB+22, p. 194–197].

Lifelog retrieval evaluations come with their particular challenges, such as relevance judgments being even more subjective than in the traditional retrieval context [GSD14]. The first test collection for lifelog research was released in [GJH+16], and has paved the way for future evaluation campaigns such as LSC, and future NTCIR [GJH+19] and ImageCLEF [DPR+18; DNPR+19; NLZ+20] tasks.

For our work, we have chosen to focus on metrics suited for scenarios in which only the first correct item is of interest. Evaluations where multiple relevant items are considered and relevance judgments are provided might use Precision/Recall-based metrics such as Mean Average Precision (MAP) or E- resp. F-Measure [van79; SBH97], or Discounted Cumulative Gain (DCG) for non-binary relevance judgments.

7.3 Temporal Information Retrieval

There are different definitions of *Temporal Information Retrieval* in literature. [CDJ+14] define temporal information retrieval as "satisfy[ing] search needs by combining the traditional notion of document relevance with temporal relevance". [KBN15] define it as focused on "how user behavior, document content

and scale vary with time". Generally, there is a significant focus on aspects like freshness of retrieved documents, considering the relation between temporal features of a document and the query [KGC11; MTY16] or multi-versioning. This is not the focus of this thesis, as we do not consider the creation date of a document during the retrieval process and leave multi-version aspects for future work. Additionally, and consistent with earlier work in the context of web retrieval [AGB07], the research question of how to best extract temporal information from documents is raised (e.g., by detecting time specifications in a text which are relative such as a weekday and mapping them to an actual date). Early work in this domain relied on explicit specification. A prominent example is TimeML [PCI+03], upon which extraction and normalization of temporal expressions could be built [ASB+11].

On the feature side, there is also work which tries to identify activities over a longer period of time in videos [HGS19]. We view this as tackling a different dimension of the problem of temporal context, as this kind of research helps only with the issue of one singular subquery, but not an arbitrary sequential combination of activities. In the context of textual embeddings, recent work has worked on scenarios where the desired item described by the query is shorter or longer than the result item [DCZ+22]. This is related to the problems tackled in Chapter 4, where we also need to aggregate result items to match the query, however they do not consider multiple subqueries.

L'avenir, tu n'as point à le prévoir
mais à le permettre

— Antoine de Saint-Exupéry,
Citadelle, LVI [p. 167]

8

Conclusion and Outlook

In this final chapter, we summarize the contributions and results in Section 8.1 and describe relevant and interesting future research directions in Section 8.2. In particular, we refer back to our motivating scenario and requirements and how our contributions address those.

8.1 Conclusion

As the starting point of this thesis, we have argued that the growth in variety and volume of multimedia data necessitates research on a range of topics relating to multimedia retrieval. Based on two motivating scenarios in the domains of video and lifelog retrieval, we derived requirements for modern multimedia retrieval systems wishing to address complex information needs of real users in a comprehensive manner. After reviewing the foundations of multimodal multimedia retrieval, we have made several contributions which further move the field of multimedia retrieval in general and video and lifelog retrieval more specifically toward the goal of a general purpose retrieval model for complex information needs, which is backed by a usable implementation and evaluated in a meaningful manner.

In Chapter 4, we introduced a retrieval model for complex information needs with temporal components. The data model generalizes to all kinds of multimedia and the query model is designed to enable efficient retrieval while enabling the expression of both simple and more elaborate queries. The model for multimodal queries is based on the assumption that the traditional separation between application layer and database layer is used, and shows how a variety of information needs for different modalities and underlying retrieval features,

and the corresponding notions of relevance, can be mapped to our model. The model for temporal queries considers late fusion of subqueries, and cleanly separates the different steps taken by our approaches and others in literature. For each of those steps, we show different original approaches and how other implementations found in literature can be mapped to our model. We also present different modular algorithms which can be used and evaluated in different formats. It is to the best of our knowledge the first general purpose retrieval model for queries with a temporal context in multimedia retrieval which is sufficiently formalized and detailed to enable a comparison of different approaches in literature and enables the development and comparison of new algorithms.

In Chapter 5, we presented our contributions to vitrivr, an open-source multimodal multimedia retrieval system. Major contributions have been made to user interface and retrieval engine, and minor contributions to the database layer. vitrivr covers the full scope of a modern multimedia retrieval system and user journey, including feature extraction, query formulation and execution, and result presentation and browsing. During this dissertation project, vitrivr has been used in a variety of contexts and has served as a research platform for different applications. It is now used in two large-scale interdisciplinary research projects, one in the domain of cultural heritage [LFF22], and the other in the domain of VR and AR [Wel22].

Our contributions are evaluated in Chapter 6 through both a user-centric and system-centric lens. In the user-centric evaluation, we show results from interactive benchmarking campaigns from 2019 to 2022, and show that vitrivr is a competitive system in both the domain of video and lifelog retrieval, achieving the highest score three times between 2019 and 2022. New ideas for log analysis alongside contributions in [HGB+22; RGH+21; HSS23] further analysis methodology for interactive retrieval evaluations. We also discuss our experience with these campaigns backed by data and provide recommendations for future interactive retrieval evaluations. In the system-centered evaluation, we evaluate model and implementation in a more traditional manner using a newly created dataset with appropriate metrics and significance tests. Our results show that enabling users to express temporal context is essential when considering information needs which are not only focused on a single element of a collection, and that algorithms which consider user-specified distances perform better than those who do not.

Taken together, we make a strong case for considering temporal context not just on the feature level, but also enabling users to explicitly express it and

consider it in the retrieval model. Our model serves as an important step to fulfilling the user needs and requirements outlined at the beginning of this thesis. Together with concurrent work on database management for multimedia retrieval [Gas23] and multimodel multilanguage databases called Polystores [Vog22], we also lay the groundwork for research on retrieval models which bring together work from the domain of databases, multimedia retrieval, and Human-Computer Interaction (HCI).

8.2 Future Work

There are many different interesting directions for future research in the field of multimedia retrieval, and in this section we discuss potential future work which would tackle important unsolved problems.

Explainable Retrieval: Explaining retrieval outcomes to users is especially relevant given the rise of Artificial Intelligence (AI) methods involved in the process [BADDS+20], and can be considered a multidisciplinary effort [BBB+20]. More broadly, recent regulations in the European Union include a "right to explanation", which includes affects also traditional algorithms [GF17]. Making sure that retrieval results are not only relevant, but also the reason for them being shown is explained to users is relevant for both the general public and expert users, and affects all aspects of a retrieval system, from retrieval model and underlying features to result presentation. Explainability is thus a topic that would affect all contributions of this thesis.

Novel Features for Multimedia Retrieval: The underlying retrieval features responsible for understanding multimedia and making it searchable have evolved significantly in the past years. Breakthroughs in embeddings of visual and textual content [LXY+19; RKH+21] and speech recognition [RKX+22] has led to significant gains in retrieval effectiveness. Research in improving understanding of visual and textual content has significant potential, especially when also considering advances in text generation like GPT [BMR+20], image generation like DALL-E [RPG+21] or Imagen [SCS+22], making work on novel retrieval features still an essential tasks for the future of multimedia retrieval research. Analyzing how novel features and users' changing queries based on their mental model of retrieval functionality affects the kinds of information needs addressed in this thesis is thus a natural continuation of this thesis. Of particular interest could be queries that are even more closely

aligned with natural language descriptions of entire events, as users potentially will get used to machines understanding this kind of query based on their experience with text generation models like GPT.

Multimedia: The model and implementation presented in this thesis still considers multimedia mainly siloed into either video, or images, or audio. As identified in Section 2.3, fully enabling users to target different levels of abstraction in multimedia by considering composite multimedia data is relevant in different contexts. Example document types include pdfs, presentation slides, or patient histories in a medical context, all of which have broad and significant practical applications.

Multimedia Summarization: Often, the content of a multimedia object or a result element can and should be represented in a more concise summary to the user [Sme07]. This has been an active area of research in video [BGS+10; HXL+11], lifelogs [PC11], and audio [LLC19; VG20]. Related, but with a different focus is the research area of *content-based fusion*, which considers content when combining results. The algorithms presented in this thesis only consider the content of the results to be combined through the underlying retrieval functionality, but are content-agnostic as they focus on the scores. Content-based fusion has already received some attention in the literature [KK09; LMR+18; RM19]. This is an interesting avenue in general, but specifically when considering queries which have an inherent temporal context. Making use of this temporal context when creating result summaries could be an interesting direction of future research.

Novel Input Modalities: Research at the intersection between multimedia retrieval and human-computer interaction, especially for different input modalities such as mobile phones [CMS17; BSK+21], AR [PPE+21; RRT22], or VR [Sch21a; SGH+21a; DJ22b] is essential in a world where the predominant way of accessing information is on mobile devices for a significant amount of people. Research in this direction could look particularly at questions of query formulation and result presentation and browsing, two areas where the implementation in this thesis has followed a conventional approach with a desktop interface. The relation between the kinds of information needs users have and the context with which they use the interface should also be kept in mind, as users on a mobile or VR device tend to have different information needs than those in front of a traditional desktop computer.

Interaction Models for Interactive Retrieval Systems: As discussed in this thesis, and in the analysis of benchmarking campaigns [RGH+21; RGL+21], a interaction model for retrieval systems which is general enough to be applicable to different implementations would enable a comparison of information seeking strategies for different retrieval systems. Previous research in this direction [LVM+21] has struggled to capture and compare the different interaction modalities along the user journey in a way which enables meaningful results.

Large-Scale Interactive Retrieval Evaluations: Bringing together research in the HCI domain with user-centric benchmarking campaigns would enable much more robust evaluation results when comparing user interaction modalities of multimedia retrieval systems. During this dissertation project, minor contributions were made toward a larger (n = 15) comparison of two retrieval systems [RGH+21], where analysis and discussion showed that there are interesting insights to be gained by increasing the number of participants and tasks, and controlling the setting more rigidly, and we discuss similar issues in [HSS23]. This has already motivated research on the user interface of vitrivr-VR.

Structured Query Languages for Multimedia Retrieval: One of the fundamental research questions raised in [JWZ+16, p. 299] is "Is a novel multimedia query language needed [...] to fully support multimedia analytics, or is an extension of classic query languages sufficient?". Given the prominence and continued success of SQL, we think that extending SQL with the necessary functionality for multimedia retrieval would be extremely beneficial for modern multimedia retrieval systems for developers and users alike with expected benefits in terms of efficiency, effectiveness and developer productivity. As argued in previous chapters, work done in parallel to and collaboration with this dissertation on multimedia databases and Polystores [Gas23; Vog22] paves the way together with this thesis for a new generation of multimedia retrieval system which would truly bridge the gap between the worlds of multimedia retrieval and databases.

The appendix lies
In the back of the thesis,
Waiting to be found.

Tiny appendix,
Nestled deep in thesis pages,
Key to the whole work.

— OpenAI ChatGPT,
Q: "Suggest a Haiku for the appendix of a PhD thesis"

A

Additional Results

In this chapter, we present additional results from the evaluation. In particular, results about quality are in Appendix A.1, about runtime in Appendix A.2, and significance results in Appendix A.3.

A.1 Retrieval Quality

We show how the quality of results changes with the number of subqueries for the lifelog dataset in Figure A.1 and for the video dataset in Figure A.2.

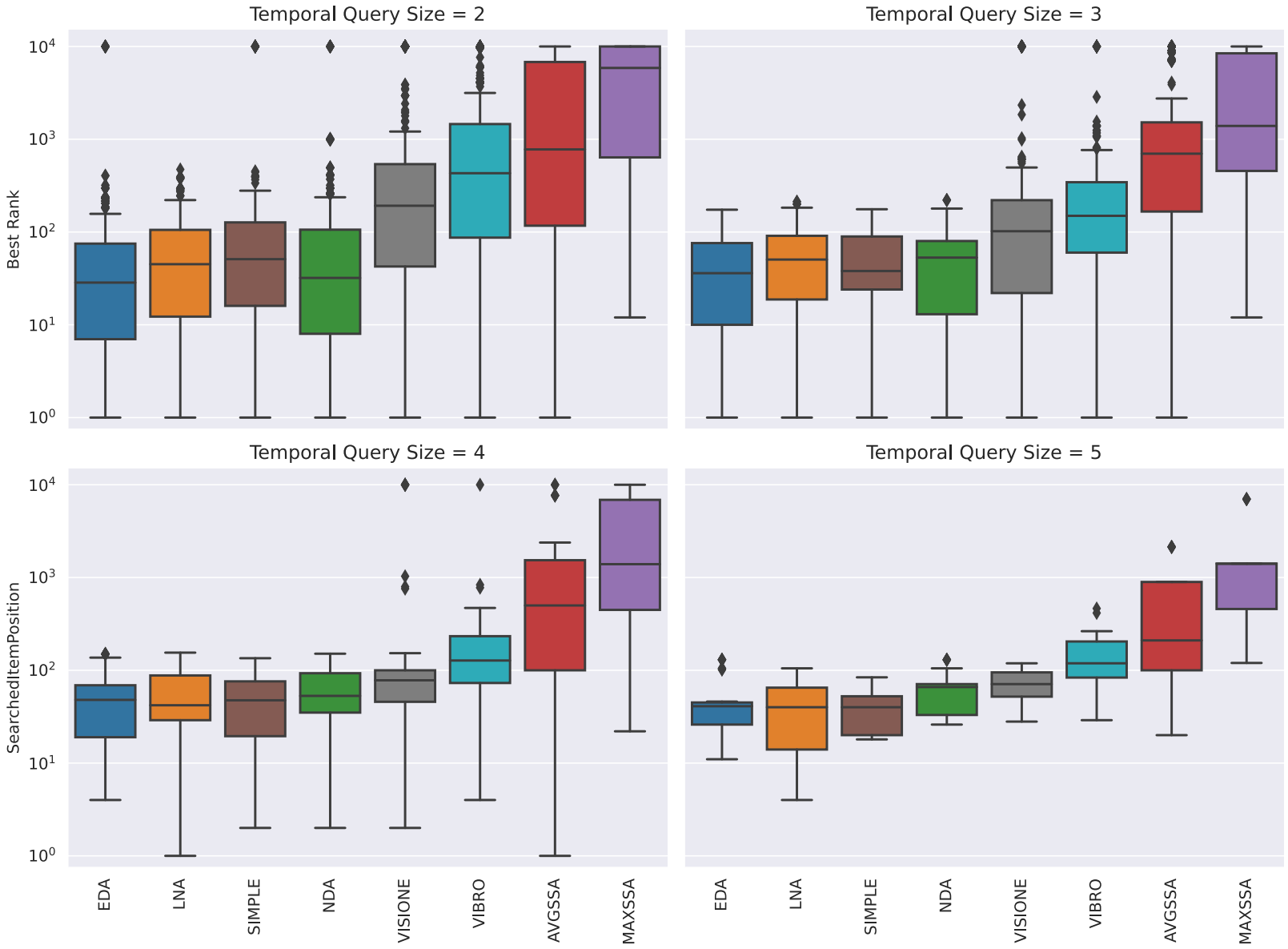


Figure A.1 **Lifelog Retrieval: Algorithm Quality Comparison:** Comparison of the best rank of a desired item for different sizes of a temporal query, exploded by number of query elements. Runtime is capped at 10 seconds

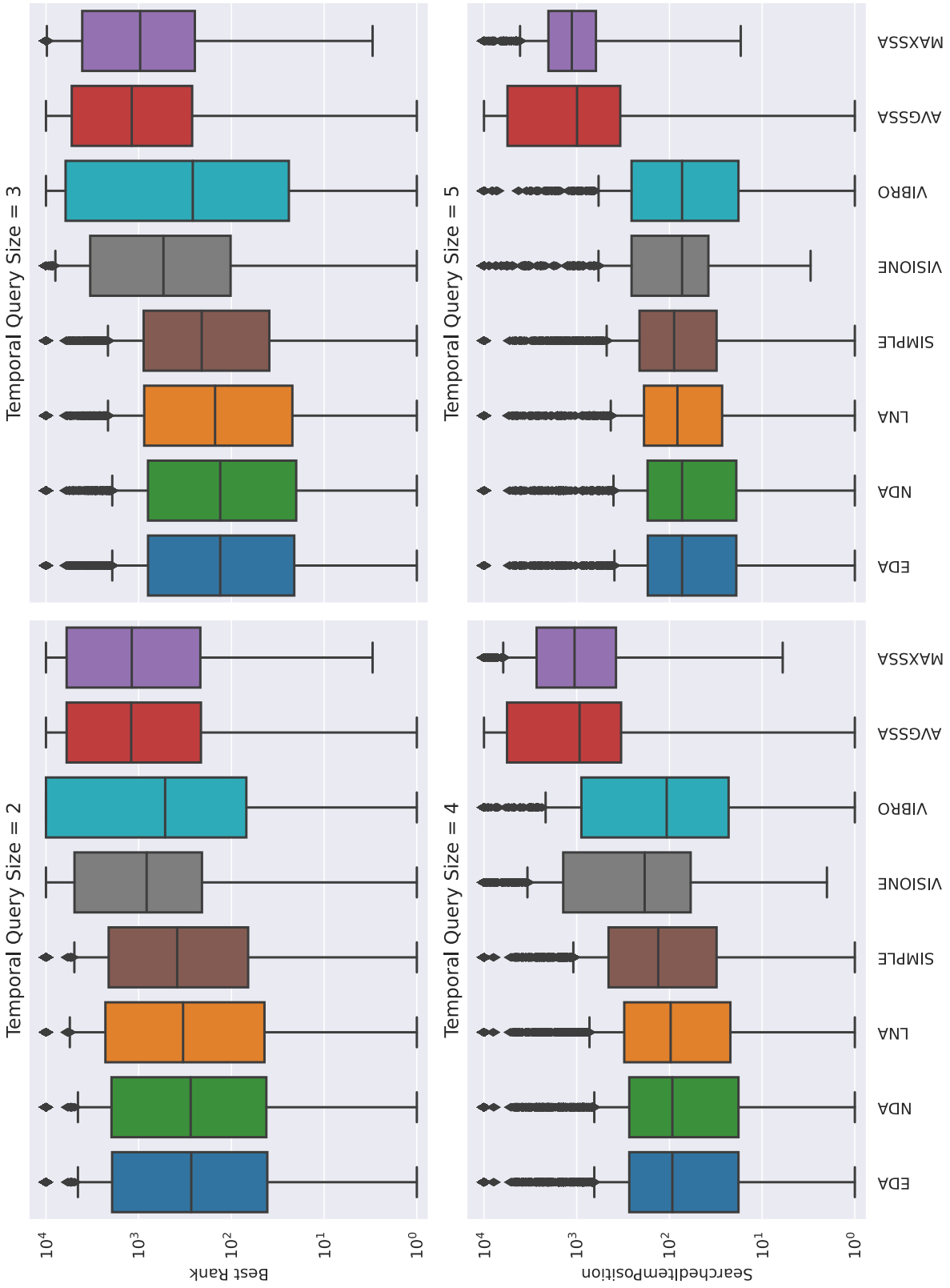


Figure A.2 Video Retrieval: Algorithm Quality Comparison: Comparison of the best rank of a desired item for different sizes of a temporal query, exploded by number of query elements. Ranks are capped at 10'000

A.2 Retrieval Runtime

We show how algorithm runtime changes with the number of subqueries for the lifelog dataset in Figure A.3 and for the video dataset in Figure A.4.

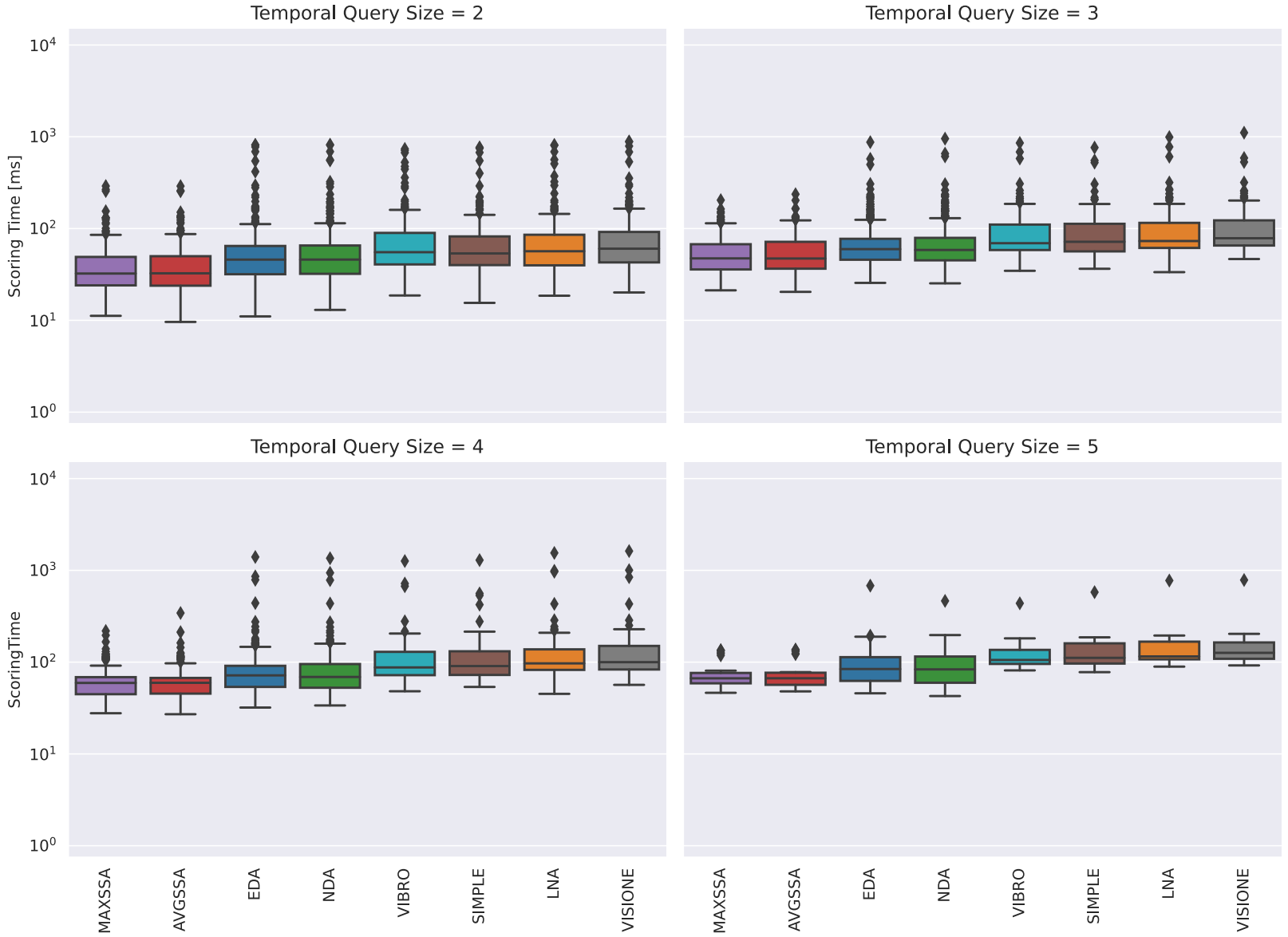


Figure A.3 **Lifelog Algorithm Runtime Comparison:** Comparison of the algorithm runtime over all lifelog tasks, exploded by number of query elements. Runtime is capped at 10 seconds

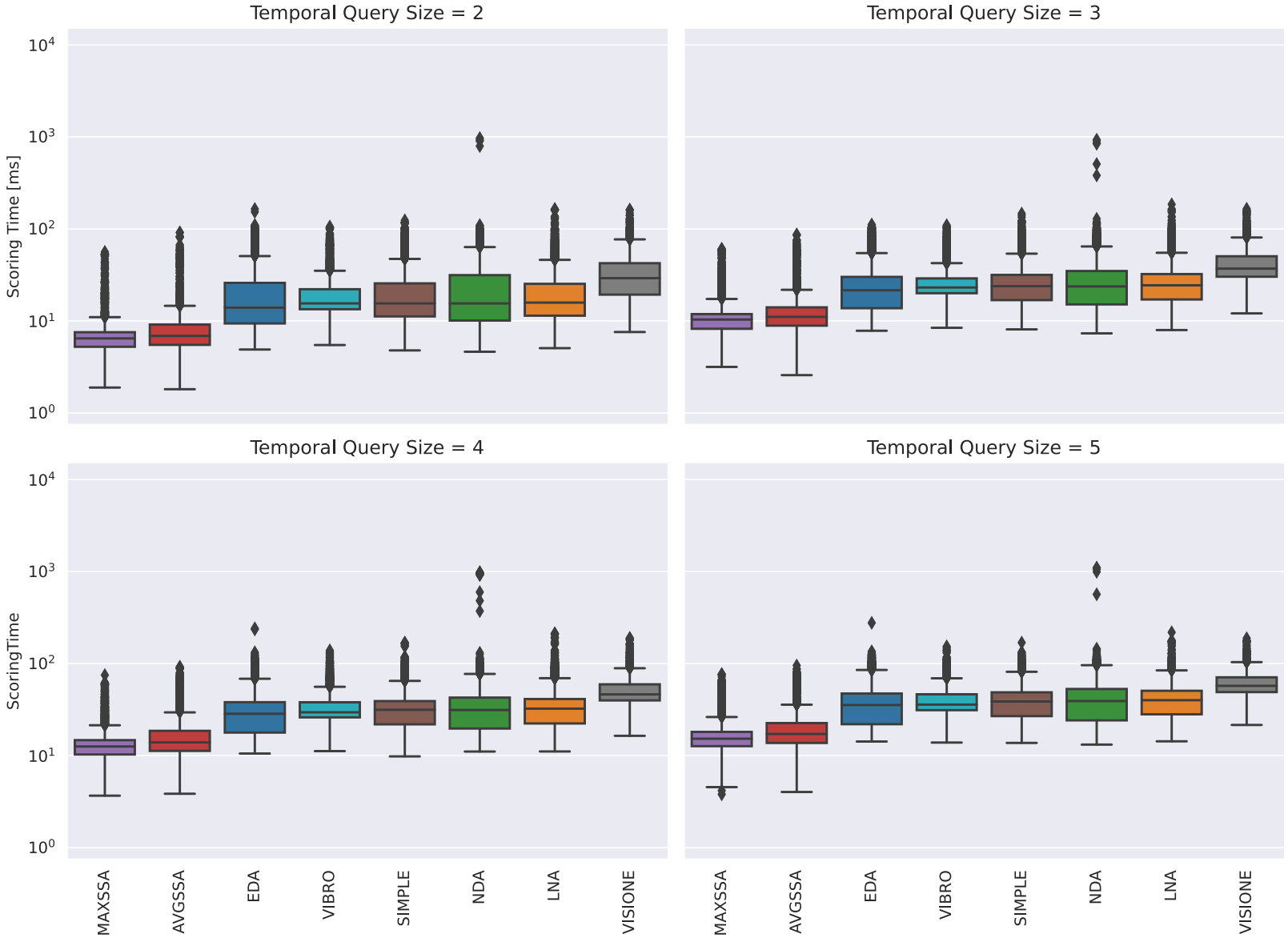


Figure A.4 **Video Algorithm Runtime Comparison:** Comparison of the algorithm runtime over all video tasks, exploded by number of query elements

Table A.1 Tabular overview of best rank significance results for original Lifelog queries

	EDA	LNA	SIMPLE	NDA	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		1.0	**0.022**	1.0	**0.013**	**0.001**	**0.004**	**0.001**
LNA			0.454	1.0	**0.004**	**0.002**	**0.049**	**0.001**
SIMPLE				0.092	0.143	**0.001**	0.077	**0.001**
NDA					**0.013**	**0.001**	**0.004**	**0.001**
VISIONE						1.0	1.0	**0.001**
VIBRO							1.0	**0.002**
AVGSSA								**0.013**
MAXSSA								

Table A.2 Tabular overview of best rank significance results for original Video queries

	EDA	NDA	LNA	SIMPLE	VISIONE	VIBRO	AVGSSA	MAXSSA
EDA		1.0	**0.001**	0.055	**0.001**	**0.007**	**0.001**	**0.001**
NDA			**0.001**	0.055	**0.001**	**0.007**	**0.001**	**0.001**
LNA				0.323	**0.001**	**0.002**	**0.001**	**0.001**
SIMPLE					**0.001**	0.088	**0.001**	**0.001**
VISIONE						0.073	**0.008**	0.052
VIBRO							**0.001**	**0.001**
AVGSSA								**0.001**
MAXSSA								

A.3 Significance Tests

A.3.1 Retrieval Quality

We additionally show significance results on only the original queries without expansion in Tables A.1 and A.2 for lifelog and video retrieval. These show similar results as the ones in Chapter 6, with lower significance values which is to be expected as the dataset is smaller.

To demonstrate the impact design choices of the algorithms and various parameters, such as pre- and post-aggregation, number of candidates to be generated or system configurations, we show significance values when comparing different parameters for the lifelog and video dataset in Tables A.3 and A.4. We will briefly discuss and contextualize the tables.

Lifelog The table shows that while for pre-aggregation, no significant difference can be seen except for NDA, there are significant differences when en-

abling post-aggregation across all algorithms. This makes sense as enabling post-aggregation reduces the number of duplicates in the result set and thus makes the correct item appear higher up in the results. Specifically for the lifelog retrieval, we have tested whether leaving out information about the dates such as day of the month makes a difference, and we can see that there is no statistically significant difference for any of the algorithms, meaning that content-based features are more important.

Looking at candidate numbers to be generated, while there seems to be a sweet spot from a performance perspective between looking at too few or too many segments, we do not see robust differences across algorithms.

For system configurations, the maximum number of results overall or per feature has similar considerations as with candidate numbers. As expected, 100 results are too few and significant differences to the chosen number of 10'000 are present across all algorithms. While there are some quality considerations between 1'000, 10'000 and 50'000, the differences are not significant across all algorithms. This makes intuitive sense and is also consistent with the assumptions discussed in Section 3.5, where the top ranked items are the most relevant ones and there is a significant drop-off in usefulness after a certain point. The numbers for maximum results per feature are similar and thus similar considerations apply.

Video For video retrieval, the results are much more robust which also makes sense due to the larger dataset. Similar to the lifelog retrieval case though, we can see that increasing the number of results returned from the system has diminishing returns after a certain point.

A.3.2 Retrieval Runtime

We show the results from the significance tests for execution time in Tables A.5 and A.6. Algorithms are ordered by median execution time on both axes, which is why the ordering is slightly different for Lifelog and Video data. The reason we only put these numbers in the appendix is because the sign test becomes less meaningful for execution times where differences of milliseconds are weighted the same as differences of half a second. Nevertheless, we see robust differences for video retrieval overall and mostly robust differences for lifelog retrieval.

Table A.3 Tabular overview of significance results for Lifelog queries, all parameters. For parameters with multiple possible values, the best value was compared to all others

Method	EDA	LNA	SIMPLE	NDA	VISIONE	VIBRO	AVGSSA	MAXSSA
CLIP v VTE	0.335	**0.006**	0.258	**0.001**	0.861	**0.001**	**0.001**	**0.001**
Pre-Aggregation	0.451	0.857	0.098	**0.029**				
Post-Aggregation	**0.001**	**0.001**	**0.001**	**0.001**				
Dates in Query	0.082	0.72	0.457	0.05	0.497	0.374	0.362	0.774
C: 10 v 1	0.5	0.221	**0.008**	**0.001**				
C: 10 v 5	0.5	0.059	**0.008**	0.124				
C: 10 v 20	1.0	**0.006**	0.125	0.092				
Sys: 10000 v 100	**0.001**	**0.001**	**0.001**	**0.001**	**0.022**	**0.001**	**0.006**	0.92
Sys: 10000 v 1000	0.214	0.191	0.791	0.188	1.0	**0.019**	0.855	**0.001**
Sys: 10000 v 50000	0.275	0.125	**0.041**	0.148	0.336	0.069	0.256	**0.002**
Feature: 20000 v 10	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Feature: 20000 v 100	**0.021**	**0.004**	**0.017**	**0.017**	0.188	**0.001**	0.235	0.207
Feature: 20000 v 1000	0.784	0.225	0.86	0.589	0.661	0.098	0.077	**0.002**
Feature: 20000 v 10000	0.615	0.905	0.545	0.712	0.795	0.081	0.864	1.0
Feature: 20000 v 50000	0.289	0.532	0.175	0.366	1.0	**0.026**	0.229	0.549

Table A.4 Tabular overview of significance results for Video queries, all parameters. For parameters with multiple possible values, the best value was compared to all others

Method	EDA	NDA	LNA	SIMPLE	VISIONE	VIBRO	AVGSSA	MAXSSA
CLIP v VTE	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Pre-Aggregation	**0.001**	**0.001**	**0.001**	**0.001**				
Post-Aggregation	**0.001**	**0.001**	**0.001**	**0.001**				
C: 5 v 1	**0.001**	**0.001**	**0.001**	**0.001**				
C: 5 v 10	1.0	1.0	**0.007**	1.0				
C: 5 v 20	**0.031**	**0.001**	0.097	**0.004**				
Sys: 10000 v 100	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Sys: 10000 v 1000	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.047**
Sys: 10000 v 50000	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	0.072	1.0	1.0
Feature: 20000 v 10	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Feature: 20000 v 100	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Feature: 20000 v 1000	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.007**
Feature: 20000 v 10000	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.021**	1.0	1.0
Feature: 20000 v 50000	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.01**	1.0	1.0

Table A.5 Tabular overview of execution time significance results for all Lifelog queries

	MAXSSA	AVGSSA	EDA	NDA	VIBRO	SIMPLE	LNA	VISIONE
MAXSSA		0.347	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
AVGSSA			**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
EDA				**0.001**	**0.001**	0.181	**0.036**	**0.001**
NDA					**0.001**	0.059	0.543	**0.016**
VIBRO						**0.001**	**0.001**	**0.004**
SIMPLE							0.729	0.085
LNA								**0.001**
VISIONE								

Table A.6 Tabular overview of execution time significance results for all Video queries

	MAXSSA	AVGSSA	EDA	VIBRO	SIMPLE	NDA	LNA	VISIONE
MAXSSA		**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
AVGSSA			**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
EDA				**0.001**	**0.001**	**0.001**	**0.001**	0.55
VIBRO					**0.001**	**0.001**	**0.001**	**0.001**
SIMPLE						**0.001**	**0.001**	**0.001**
NDA							**0.001**	0.55
LNA								**0.001**
VISIONE								

I could make it longer if you like
the style
I can change it 'round

— The Beatles,
Paperback Writer

B

Dataset Information

This appendix chapter contains additional information references through the thesis such as details about the evaluation dataset in Appendix B.1 and about the query collection methodology for the video retrieval evaluation in Appendix B.2.

B.1 Task Data

Table B.1 shows which videos were shown to participants. Tasks were selected from previous VBS tasks during a student project supervised in the course of this dissertation project [Ben22b].

Table B.1 Videos which were shown to participants. Original Type refers to the task type which was used at VBS

V3C Video Id	Original Type	Start	End
2224	T-KIS	70.00	78.00
5146	V-KIS	6.00	26.00
4316	T-KIS	1390.00	1397.00
3317	V-KIS	178.00	203.00
6228	T-KIS	0.00	13.00
6561	T-KIS	670.00	689.00
3870	V-KIS	179.00	202.00
7421	T-KIS	32.00	49.00
88	V-KIS	264.00	281.00
4035	V-KIS	296.00	313.00

4979	V-KIS	333.00	352.00
4225	V-KIS	131.00	144.00
6979	V-KIS	122.00	139.00
4888	V-KIS	199.00	218.00
2034	V-KIS	38.00	57.00
2519	V-KIS	24.00	39.00
7116	V-KIS	113.00	128.00
6246	V-KIS	22.00	35.00
5531	V-KIS	74.00	87.00
6827	V-KIS	189.00	204.00
1871	T-KIS	166.00	179.00
4312	T-KIS	260.00	272.00
2630	V-KIS	63.88	82.76
3937	T-KIS	142.00	161.96
6029	V-KIS	64.40	84.36
2398	T-KIS	170.56	190.52
6195	V-KIS	47.20	71.16
5423	T-KIS	231.28	243.44
2274	V-KIS	260.40	280.36
6962	V-KIS	58.68	78.64
767	V-KIS	112.00	135.92
2630	V-KIS	160.24	180.20
6924	V-KIS	161.04	184.96
2148	T-KIS	36.00	52.96
2733	T-KIS	93.76	114.08
387	V-KIS	73.32	97.24
1263	V-KIS	95.16	119.80
7443	V-KIS	123.04	143.00
4835	V-KIS	6.12	26.08
4495	V-KIS	348.00	367.16
943	V-KIS	28.96	52.88

2700	T-KIS	290.52	309.68
6200	V-KIS	58.28	78.24
2457	T-KIS	15.20	30.76
5497	V-KIS	408.00	427.16
4612	V-KIS	49.64	69.60
4468	V-KIS	238.96	258.12
4161	T-KIS	137.12	157.24
4619	V-KIS	1003.72	1022.88
4500	V-KIS	245.60	269.52
4408	T-KIS	107.00	126.96
3589	T-KIS	284.60	304.56
7258	V-KIS	80.52	99.68
1693	T-KIS	297.00	316.96
3482	V-KIS	177.76	197.72
156	V-KIS	23.16	43.12
12	T-KIS	64.00	78.36
2423	T-KIS	13.00	32.96
2801	V-KIS	62.36	86.28
6963	V-KIS	88.76	108.72
2380	V-KIS	147.68	171.60
2332	T-KIS	154.00	171.96
4795	V-KIS	562.00	581.96
5925	V-KIS	123.60	142.76
4791	V-KIS	60.64	84.56
3919	V-KIS	121.92	145.84
2637	T-KIS	50.12	65.24
986	V-KIS	133.20	150.16
3807	T-KIS	137.00	153.96
3624	V-KIS	279.08	303.00
4887	V-KIS	163.56	183.52

B.2 Query Collection

In this section, we briefly recap the query collection process done during a master project supervised in the course of this dissertation project [Ben22b].

Each participant described videos during a time period of 30 minutes, with a limit of 2 minutes per video. Videos were played on loop during those two minutes. The specific prompt given was as follows:

> Describe the video that we show you. Try to answer the following questions:
>
> – What is the sequence of video?
> – What do you see?
> – What do you hear?
> – Is something written on the screen?

13 people described videos. These descriptions were afterwards mapped to queries for the system using the original wording. The reason we did not ask people to formulate the queries to the system directly is that we wanted to ensure people formulated descriptions as close to their perception as possible, without limiting themselves through the query formulation process of vitrivr.

Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: `https://www.tensorflow.org/`.

[Abe16] Peter Abeles. BoofCV v0.25, 2016. URL: `http://boofcv.org/`.

[AGG21] Naushad Alam, Yvette Graham, and Cathal Gurrin. Memento: A Prototype Lifelog Search Engine for LSC'21. In *Workshop on Lifelog Search Challenge*, pages 53–58. Association for Computing Machinery, 2021. ISBN: 978-1-4503-8533-6. DOI: `10.1145/3463948.3469069`.

[AGG22] Naushad Alam, Yvette Graham, and Cathal Gurrin. Memento 2.0: An Improved Lifelog Search Engine for LSC'22. In *Workshop on Lifelog Search Challenge*, pages 2–7. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: `10.1145/3512729.3533006`.

[ARG22] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. Voxento 3.0: A Prototype Voice-Controlled Interactive Search Engine for Lifelog. In *Workshop on Lifelog Search Challenge*, pages 43–47. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: `10.1145/3512729.3533009`.

[AGB07] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the value of temporal information in information retrieval. *ACM SIGIR Forum*, 41(2):35–41, 2007. ISSN: 0163-5840. DOI: `10.1145/1328964.1328968`.

[ASB+11] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. Temporal information retrieval: Challenges and opportunities. In *WWW Workshop on Linked Data on the Web*, volume 813, pages 1–8, 2011.

[ABC+21a] Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Franca Debole, Fabrizio Falchi, Claudio Gennaro, Lucia Vadicamo, and Claudio Vairo. The VISIONE Video Search System: Exploiting Off-the-Shelf Text Search Engines for Large-Scale Video Retrieval. *Journal of Imaging*, 7(5):76, 2021. ISSN: 2313-433X. DOI: 10.3390/jimaging7050076.

[ABC+22] Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Nicola Messina, Lucia Vadicamo, and Claudio Vairo. VISIONE at Video Browser Showdown 2022. In *MultiMedia Modeling*, pages 543–548. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_52.

[ABF+21] Giuseppe Amato, Paolo Bolettieri, Fabrizio Falchi, Claudio Gennaro, Nicola Messina, Lucia Vadicamo, and Claudio Vairo. VISIONE at Video Browser Showdown 2021. In *MultiMedia Modeling*, pages 473–478. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_47.

[AFG+17] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. Searching and annotating 100M Images with YFCC100M-HNfc6 and MI-File. In *International Workshop on Content-Based Multimedia Indexing*, pages 1–4. Association for Computing Machinery, 2017. ISBN: 978-1-4503-5333-5. DOI: 10.1145/3095713.3095740.

[APRS+20] Mahnaz Amiri Parian, Luca Rossetto, Heiko Schuldt, and Stéphane Dupont. Are You Watching Closely? Content-based Retrieval of Hand Gestures. In *International Conference on Multimedia Retrieval*, pages 266–270. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7087-5. DOI: 10.1145/3372278.3390723.

[AMG+22] Stelios Andreadis, Anastasia Moumtzidou, Damianos Galanopoulos, Nick Pantelidis, Konstantinos Apostolidis, Despoina Touska, Konstantinos Gkountakos, Maria Pegia, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris. VERGE in VBS 2022. In *MultiMedia Modeling*,

pages 530–536. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_50.

[AMG+21] Stelios Andreadis, Anastasia Moumtzidou, Konstantinos Gkountakos, Nick Pantelidis, Konstantinos Apostolidis, Damianos Galanopoulos, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris. VERGE in VBS 2021. In *MultiMedia Modeling*, pages 398–404. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_35.

[Ang21] Carmen Ang. Ranked: The World's Most Popular Social Networks, and Who Owns Them, 2021. URL: https://www.visualcapitalist.com/ranked-social-networks-worldwide-by-users/.

[App22] Apple. Search for photos on iPhone, 2022. URL: https://support.apple.com/en-gb/guide/iphone/iph392d77d5f/ios.

[arg22] argman. EAST: An Efficient and Accurate Scene Text Detector, 2022. URL: https://github.com/argman/EAST.

[AY99] Y Alp Aslandogan and Clement T. Yu. Techniques and systems for image and video retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, 1999. ISSN: 1558-2191. DOI: 10.1109/69.755615.

[Ass22] American Psychological Association. APA Style numbers and statistics guide, 2022. URL: https://apastyle.apa.org/instructional-aids/numbers-statistics-guide.pdf.

[AHES+10] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010. ISSN: 1432-1882. DOI: 10.1007/s00530-010-0182-0.

[ABC+21b] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quenot. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains, 2021. DOI: 10.48550/arXiv.2104.13473.

[BR11] Ricardo Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the Concepts and Technology behind Search, Second Edition*. Pearson Education Ltd., Harlow, England, 2011. ISBN: 978-0-321-41691-9.

[BSK+21] Alexandra M. Bagi, Kim I. Schild, Omar Shahbaz Khan, Jan Zahálka, and Björn Þór Jónsson. XQM: Interactive Learning on Mobile Phones. In *MultiMedia Modeling*, pages 281–293. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_24.

[BC17] Seyed Ali Bahrainian and Fabio Crestani. Towards the Next Generation of Personal Assistants: Systems that Know When You Forget. In *International Conference on Theory of Information Retrieval*, pages 169–176. Association for Computing Machinery, 2017. ISBN: 978-1-4503-4490-6. DOI: 10.1145/3121050.3121071.

[BNV+21] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *International Conference on Computer Vision*, pages 1728–1738, 2021. DOI: 10.1109/ICCV48922.2021.00175.

[BL15] Sándor Baran and Sebastian Lerch. Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society*, 141(691):2289–2299, 2015. ISSN: 1477-870X. DOI: 10.1002/qj.2521.

[BGS+10] Connelly Barnes, Dan B. Goldman, Eli Shechtman, and Adam Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics*, 29(4):89:1–89:9, 2010. ISSN: 0730-0301. DOI: 10.1145/1778765.1778826.

[BADDS+20] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.12.012.

[BCO+07] Ilaria Bartolini, Paolo Ciaccia, Vincent Oria, and M. Tamer Özsu. Flexible integration of multimedia sub-queries with qualitative preferences. *Multimedia Tools and Applications*, 33(3):275–300, 2007. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-007-0103-1.

[Bat89] Marcia J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989. ISSN: 0309-314X. DOI: 10.1108/eb024320.

[BBB+20] Valérie Beaudouin, Isabelle Bloch, David Bounie, Stéphan Clémençon, Florence d'Alché-Buc, James Eagan, Winston Maxwell, Pavlo Mozharovskyi, and Jayneel Parekh. Flexible and Context-Specific AI Explainability: A Multidisciplinary Approach, 2020. DOI: 10.48550/arXiv.2003.07703.

[BS16] Lukas Beck and Heiko Schuldt. City-Stories: A Spatio-Temporal Mobile Multimedia Search System. In *IEEE International Symposium on Multimedia*, pages 193–196. IEEE, 2016. DOI: 10.1109/ISM.2016.0046.

[BH63] Joseph Becker and Robert Hayes. Information storage and retrieval: Tools, elements, theories. *Theories. Wiley, New York*, 1963.

[BCQ+21] Callista Bee, Yuan-Jyue Chen, Melissa Queen, David Ward, Xiaomeng Liu, Lee Organick, Georg Seelig, Karin Strauss, and Luis Ceze. Molecular-level similarity search brings computing to DNA data storage. *Nature Communications*, 12(4764), 2021. DOI: 10.1038/s41467-021-24991-z.

[BMC93] Nicholas J. Belkin, Pier Giorgio Marchetti, and C. Cool. BRAQUE: Design of an interface to support user interaction in information retrieval. *Information Processing & Management*, 29(3):325–344, 1993. ISSN: 0306-4573. DOI: 10.1016/0306-4573(93)90059-M.

[Ben22a] Vera Benz. *Evaluation of Temporal Queries in Lifelog Retrieval*. Bachelor Thesis, University of Basel, 2022.

[Ben22b] Vera Benz. Result Robustness in Multimedia Retrieval Evaluations. Master Project, University of Basel, 2022.

[BL17] Adam Berger and John Lafferty. Information Retrieval as Statistical Translation. *ACM SIGIR Forum*, 51(2):219–226, 2017. ISSN: 0163-5840. DOI: 10.1145/3130348.3130371.

[BRS+19] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. V3C1 Dataset: An Evaluation of Content Characteristics. In *International Conference on Multimedia Retrieval*, pages 334–338. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6765-3. DOI: 10.1145/3323873.3325051.

[Bla88] David C. Blair. An extended relational document retrieval model. *Information Processing & Management*, 24(3):349–371, 1988. ISSN: 0306-4573. DOI: 10.1016/0306-4573(88)90101-X.

[BBF+07] Henk M. Blanken, Henk Ernst Blok, Ling Feng, and Arjen P. de Vries. *Multimedia Retrieval*. Springer, 2007. ISBN: 978-3-540-72894-8. DOI: 10.1007/978-3-540-72895-5.

[BN07] Erik Boertjes and Anton Nijholt. Interaction. In *Multimedia Retrieval*, pages 295–320. Springer, 2007. DOI: 10.1007/978-3-540-72895-5.

[BMS+01] Klemens Böhm, Michael Mlivoncic, Hans-Jörg Schek, and Roger Weber. Fast evaluation techniques for complex similarity queries. In *International Conference on Very Large Data Bases*, pages 211–220. Morgan Kaufmann, 2001.

[BGS+20] Samuel Börlin, Ralph Gasser, Florian Spiess, and Heiko Schuldt. 3D Model Retrieval Using Constructive Solid Geometry in Virtual Reality. In *International Conference on Artificial Intelligence and Virtual Reality*, pages 373–374. IEEE, 2020. DOI: 10.1109/AIVR50618.2020.00077.

[Bor03] Pia Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3):8–3, 2003.

[Bra00] G. Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.

[BVL+22] Alex Brandsen, Suzan Verberne, Karsten Lambers, and Milco Wansleeben. Can BERT Dig It? Named Entity Recognition for Information Retrieval in the Archaeology Domain. *Journal on Computing and Cultural Heritage*, 15(3):51:1–51:18, 2022. ISSN: 1556-4673. DOI: 10.1145/3497842.

[BMR+07] Gert Brettlecker, Diego Milano, Paola Ranaldi, Hans-Jörg Schek, Heiko Schuldt, and Michael Springmann. ISIS and OSIRIS: A Process-Based Digital Library Application on Top of a Distributed Process Support Middleware. In *Digital Libraries: Research and Development*, pages 46–55. Springer, 2007. ISBN: 978-3-540-77088-6. DOI: 10.1007/978-3-540-77088-6_5.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, 2020. DOI: 10.48550/arXiv.2005.14165.

[BCB+05] George Buchanan, Sally Jo Cunningham, Ann Blandford, Jon Rimmer, and Claire Warwick. Information Seeking by Humanities Scholars. In *Research and Advanced Technology for Digital Libraries*, pages 218–229. Springer, 2005. ISBN: 978-3-540-31931-3. DOI: 10.1007/11551362_20.

[Bus45] Vannevar Bush. As we may think. *The atlantic monthly*, 176(1):101–108, 1945.

[BM14] György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014. ISSN: 1471-0048. DOI: 10.1038/nrn3687.

[CDJ+14] Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys*, 47(2):15:1–15:41, 2014. ISSN: 0360-0300. DOI: 10.1145/2619088.

[Can86] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1986.4767851.

[CWZ+11] Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang. Edgel index for large-scale sketch-based image search. In *Conference on Computer Vision and Pattern Recognition*, pages 761–768. IEEE, 2011. DOI: 10.1109/CVPR.2011.5995460.

[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 *International Conference on Multimedia*, pages 1605–1608. Association for Computing Machinery, 2010. ISBN: 978-1-60558-933-6. DOI: 10.1145/1873951.1874299.

[CHS+19] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, 2019. DOI: 10.48550/arXiv.1812.08008.

[CGR+20] Mariona Carós, Maite Garolera, Petia Radeva, and Xavier Giro-i-Nieto. Automatic Reminiscence Therapy for Dementia. In *International Conference on Multimedia Retrieval*, pages 383–387. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7087-5. DOI: 10.1145/3372278.3391927.

[CPK+08] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 651–658. Association for Computing Machinery, 2008. ISBN: 978-1-60558-164-4. DOI: 10.1145/1390334.1390445.

[CG16] Donald O. Case and Lisa M. Given. *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior*. Emerald Group Publishing, 2016. ISBN: 978-1-78560-967-1. DOI: 10.1002/asi.23778.

[CZK+21] Floris Chabert, Jingwen Zhu, Brett Keating, and Vinay Sharma. Recognizing People in Photos Through Private On-Device Machine Learning, 2021. URL: https://machinelearning.apple.com/research/recognizing-people-photos.

[CMN04] Abdolah Chalechale, Alfred Mertins, and G Naghdy. Edge image description using angular radial partitioning. *IEE Proceedings-Vision, Image and Signal Processing*, 151(2):93–101, 2004. DOI: 10.1049/ip-vis:20040332.

[CHM+19] Shih-Fu Chang, Alex Hauptmann, Louis-Philippe Morency, Sameer Antani, Dick Bulterman, Carlos Busso, Joyce Chai, Julia Hirschberg, Ramesh Jain, Ketan Mayer-Patel, Reuven Meth, Raymond Mooney, Klara Nahrstedt, Shri Narayanan, Prem Natarajan, Sharon Oviatt, Balakrishnan Prabhakaran, Arnold Smeulders, Hari Sundaram, Zhengyou Zhang, and Michelle Zhou. Report of 2017 NSF Workshop on Multimedia Challenges, Opportunities and Research Roadmaps, 2019. DOI: 10.48550/arXiv.1908.02308.

[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. ISSN: 1558-1756. DOI: 10.1109/MCG.2010.92.

[Chr07] Michael G. Christel. Carnegie Mellon University traditional informedia digital video retrieval system. In *International Conference on Image and Video Retrieval*, page 647. Association for Computing Machinery, 2007. ISBN: 978-1-59593-733-9. DOI: 10.1145/1282280.1282374.

[CR95] Tat-Seng Chua and Li-Qun Ruan. A video retrieval and sequencing system. *ACM Transactions on Information Systems*, 13(4):373–407, 1995. ISSN: 1046-8188. DOI: 10.1145/211430.211431.

[CSC+07] Karen Church, Barry Smyth, Paul Cotter, and Keith Bradley. Mobile information access: A study of emerging search behavior on the mobile Internet. *ACM Transactions on the Web*, 1(1):4–es, 2007. ISSN: 1559-1131. DOI: 10.1145/1232722.1232726.

[CPZ98] Paolo Ciaccia, Marco Patella, and Pavel Zezula. Processing complex similarity queries with distance-based access methods. In *Advances in Database Technology*, pages 9–23. Springer, 1998. ISBN: 978-3-540-69709-1. DOI: 10.1007/BFb0100974.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970. ISSN: 0001-0782. DOI: 10.1145/362384.362685.

[CHL+20] Mihai Gabriel Constantin, Steven Hicks, Martha Larson, and Ngoc-Thanh Nguyen. MediaEval multimedia evaluation bench-

mark: Tenth anniversary and counting. *ACM SIGMM Records*, 12(2), 2020. DOI: 10.1145/3548562.3548568.

[Coo76] William Cooper. A General Mathematical Model for Information Retrieval Systems. *The Library Quarterly*, 46(2):153–167, 1976. ISSN: 0024-2519. DOI: 10.1086/620501.

[CMO+96] I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos. PicHunter: Bayesian relevance feedback for image retrieval. In *International Conference on Pattern Recognition*, volume 3, 361–369 vol.3, 1996. DOI: 10.1109/ICPR.1996.546971.

[CLVR+98] Fabio Crestani, Mounia Lalmas, Cornelis J. Van Rijsbergen, and Iain Campbell. ''Is this document relevant?. . . probably'': a survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998. ISSN: 0360-0300. DOI: 10.1145/299917.299920.

[CMS17] Fabio Crestani, Stefano Mizzaro, and Ivan Scagnetto. *Mobile Information Retrieval*. Springer International Publishing, 2017. ISBN: 978-3-319-60776-4 978-3-319-60777-1. DOI: 10.1007/978-3-319-60777-1.

[DPR+18] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, and Cathal Gurrin. Overview of ImageCLEFlifelog 2018: daily living understanding and lifelog moment retrieval. In *Conference and Labs of the Evaluation Forum*. CEUR-WS, 2018.

[DNPR+19] Duc Tien Dang Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, Minh Triet Tran, Tu-Khiem Le, Van-Tu Ninh, and Cathal Gurrin. Overview of ImageCLEFlifelog 2019: Solve My Life Puzzle and Lifelog Moment Retrieval. *CEUR Workshop Proceedings*, 2019. ISSN: 1613-0073.

[DJL+08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5:1–5:60, 2008. ISSN: 0360-0300. DOI: 10.1145/1348246.1348248.

[DM10] Adrien Depeursinge and Henning Müller. Fusion Techniques for Combining Textual and Visual Information Retrieval. In *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*,

pages 95–114. Springer, 2010. ISBN: 978-3-642-15181-1. DOI: 10.1007/978-3-642-15181-1_6.

[Dia98] Ted Diamond. *Information Retrieval Using Dynamic Evidence Combination*. PhD thesis, Syracuse University, 1998.

[Die00] Thomas G. Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, pages 1–15. Springer, 2000. ISBN: 978-3-540-45014-6. DOI: 10.1007/3-540-45014-9_1.

[DM46] Wilfrid J Dixon and Alexander M Mood. The statistical sign test. *Journal of the American Statistical Association*, 41(236):557–566, 1946.

[DK07] Martin Dodge and Rob Kitchin. 'Outlines of a World Coming into Existence': Pervasive Computing and the Ethics of Forgetting. *Environment and Planning B: Planning and Design*, 34(3):431–445, 2007. ISSN: 0265-8135. DOI: 10.1068/b32041t.

[Dom08] Sándor Dominich. *The Modern Algebra of Information Retrieval*, volume 24. Springer, 2008. ISBN: 978-3-540-77658-1. DOI: 10.1007/978-3-540-77659-8.

[DCZ+22] Jianfeng Dong, Xianke Chen, Minsong Zhang, Xun Yang, Shujie Chen, Xirong Li, and Xun Wang. Partially Relevant Video Retrieval. In *International Conference on Multimedia*, pages 246–257, 2022. DOI: 10.1145/3503161.3547976.

[DSP+20] Yiming Dong, Fajia Sun, Zhi Ping, Qi Ouyang, and Long Qian. DNA storage: research landscape and future prospects. *National Science Review*, 7(6):1092–1107, 2020. ISSN: 2095-5138. DOI: 10.1093/nsr/nwaa007.

[DG20] Aaron Duane and Cathal Gurrin. Baseline Analysis of a Conventional and Virtual Reality Lifelog Retrieval System. In *MultiMedia Modeling*, pages 412–423. Springer International Publishing, 2020. ISBN: 978-3-030-37734-2. DOI: 10.1007/978-3-030-37734-2_34.

[DJ22a] Aaron Duane and Björn Þór Jónsson. ViRMA: Virtual Reality Multimedia Analytics at Video Browser Showdown 2022. In *MultiMedia Modeling*, pages 580–585. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_58.

[DJ22b] Aaron Duane and Björn Pór Jónsson. ViRMA: Virtual Reality Multimedia Analytics. In *International Conference on Multimedia Retrieval*, pages 211–214. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9238-9. DOI: 10.1145/3512527.3531352.

[Ebb85] Hermann Ebbinghaus. *Über Das Gedächtnis: Untersuchungen Zur Experimentellen Psychologie*. Duncker & Humblot, 1885.

[EML+18] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, and Michael Rubinstein. Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation. *ACM Transactions on Graphics*, 37(4):1–11, 2018. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3197517.3201357.

[Fag99] Ronald Fagin. Combining Fuzzy Information from Multiple Systems. *Journal of Computer and System Sciences*, 58(1):83–99, 1999. ISSN: 0022-0000. DOI: 10.1006/jcss.1998.1600.

[FKM+06] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing Partial Rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648, 2006. ISSN: 0895-4801. DOI: 10.1137/05063088X.

[FKS03] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *ACM SIGMOD International Conference on Management of Data*, pages 301–312. Association for Computing Machinery, 2003. ISBN: 978-1-58113-634-0. DOI: 10.1145/872757.872795.

[FLN01] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In *Symposium on Principles of Database Systems*, pages 102–113. Association for Computing Machinery, 2001. ISBN: 978-1-58113-361-5. DOI: 10.1145/375551.375567.

[Fer17] Nicola Ferro. Reproducibility Challenges in Information Retrieval Evaluation. *Journal of Data and Information Quality*, 8(2):1–4, 2017. ISSN: 1936-1955, 1936-1963. DOI: 10.1145/3020206.

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the QBIC system.

Computer, 28(9):23–32, 1995. ISSN: 1558-0814. DOI: 10.1109/2.410146.

[Fou21] Apache Software Foundation. TFIDFSimilarity (Lucene 9.0.0 core API), 2021. URL: https://lucene.apache.org/core/9_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html.

[FCS+13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[Fuh92] Norbert Fuhr. Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3):243–255, 1992. ISSN: 0010-4620. DOI: 10.1093/comjnl/35.3.243.

[Fuh12] Norbert Fuhr. Salton award lecture: Information retrieval as engineering science. *Special Interest Group on Information Retrieval (SIGIR) Forum*, 46(2):19–28, 2012. DOI: 10.1145/2422256.2422259.

[Fuh14] Norbert Fuhr. Bridging information retrieval and databases. In *Bridging between Information Retrieval and Databases. PROMISE Winter School 2013*, pages 97–115. Springer, 2014. DOI: 10.1007/978-3-642-54798-0_5.

[Gad45] J. H. Gaddum. Lognormal Distributions. *Nature*, 156(3964):463–466, 1945. ISSN: 1476-4687. DOI: 10.1038/156463a0.

[GM20] Damianos Galanopoulos and Vasileios Mezaris. Attention Mechanisms, Signal Encodings and Fusion Strategies for Improved Ad-hoc Video Search with Dual Encoding Networks. In *International Conference on Multimedia Retrieval*, pages 336–340. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7087-5. DOI: 10.1145/3372278.3390737.

[Gal79] Francis Galton. XII. The geometric mean, in vital and social statistics. *Proceedings of the Royal Society of London*, 29(196-199):365–367, 1879.

[Gas17] Ralph Gasser. *Towards an All-Purpose, Content-Based Multimedia Information Retrieval System*. Master's thesis, University of Basel, 2017.

[GRH+20] Ralph Gasser, Luca Rossetto, Silvan Heller, and Heiko Schuldt. Cottontail DB: An Open Source Database System for Multimedia Retrieval and Analysis. In *International Conference on Multimedia*, pages 4465–4468. Association for Computing Machinery, 2020. DOI: 10.1145/3394171.3414538.

[GRH+21] Ralph Gasser, Luca Rossetto, Silvan Heller, and Heiko Schuldt. Multimedia Retrieval and Analysis with Cottontail DB. *ACM SIGMM Records*, 2021. ISSN: 1947-4598. DOI: 10.1145/3577934.3577940.

[GRS19a] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. Multimodal Multimedia Retrieval with vitrivr. In *International Conference on Multimedia Retrieval*, pages 391–394. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6765-3. DOI: 10.1145/3323873.3326921.

[GRS19b] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. Towards an All-Purpose Content-Based Multimedia Information Retrieval System, 2019. DOI: 10.48550/arXiv.1902.03878.

[Gas23] Ralph Marc Philipp Gasser. *Data Management for Dynamic Multimedia Analytics and Retrieval*. PhD thesis, University of Basel, 2023.

[Gau23] Carl Friedrich Gauss. *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae*, volume 2. H. Dieterich, 1823.

[GBL+02] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. MyLifeBits: fulfilling the Memex vision. In *International Conference on Multimedia*, pages 235–238. Association for Computing Machinery, 2002. ISBN: 978-1-58113-620-3. DOI: 10.1145/641007.641053.

[GLC+95] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C Smith. Query by humming: Musical information retrieval in an audio database. In *International Conference on Multimedia*, pages 231–236. Association for Computing Machinery, 1995.

[Gia18] Ivan Giangreco. *Database Support for Large-Scale Multimedia Retrieval*. PhD thesis, University of Basel, 2018. DOI: 10.5451/unibas-006827345.

[GS16] Ivan Giangreco and Heiko Schuldt. ADAMpro: Database Support for Big Multimedia Retrieval. *Datenbank-Spektrum*, 16(1):17–26, 2016. ISSN: 1610-1995. DOI: 10.1007/s13222-015-0209-y.

[GJS18] Daniele Giunchi, Stuart James, and Anthony Steed. 3D sketching for interactive model retrieval in virtual reality. In *Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 1–12. Association for Computing Machinery, 2018. ISBN: 978-1-4503-5892-7. DOI: 10.1145/3229147.3229166.

[GGR+17] Prateek Goel, Ivan Giangreco, Luca Rossetto, Claudiu Tănase, and Heiko Schuldt. "Hey, vitrivr!" – A Multimodal UI for Video Retrieval. In *Advances in Information Retrieval*, pages 749–752. Springer International Publishing, 2017. ISBN: 978-3-319-56608-5. DOI: 10.1007/978-3-319-56608-5_75.

[GF17] Bryce Goodman and Seth Flaxman. European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine*, 38(3):50–57, 2017. ISSN: 2371-9621. DOI: 10.1609/aimag.v38i3.2741.

[Goo22] Google. Search by people, things & places in your photos - Android - Google Photos Help, 2022. URL: https://support.google.com/photos/answer/6128838.

[Gro96] E.E.W. Group. Evaluation of Natural Language Processing Systems. Technical report, ISSCO, 1996.

[Gst21] Viktor Gsteiger. *Evaluating Algorithms for Temporal Queries in Ad-Hoc Video Retrieval*. Bachelor Thesis, University of Basel, 2021.

[GWG+03] Maël Guillemot, Pierre Wellner, Daniel Gatica-Perez, and Jean-Marc Odobez. A Hierarchical Keyframe User Interface for Browsing Video over the Internet. Technical report, IDIAP, 2003.

[GJH+19] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, V.-T. Ninh, T.-K. Le, R. Albatal, D.-T. Dang-Nguyen, and G. Healy. Overview of the NTCIR-14 Lifelog-3 task, Proceedings Paper, 2019. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/01-NTCIR14-OV-LIFELOG-GurrinC.pdf.

[GJH+16] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. NTCIR Lifelog: The First Test Collection for Lifelog Research. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 705–708. Association for Computing Machinery, 2016. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914680.

[GJS+21] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. Introduction to the Fourth Annual Lifelog Search Challenge, LSC'21. In *International Conference on Multimedia Retrieval*, pages 690–691. Association for Computing Machinery, 2021. ISBN: 978-1-4503-8463-6. DOI: 10.1145/3460426.3470945.

[GJS+22] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. Introduction to the fifth annual lifelog search challenge, LSC'22. In *International Conference on Multimedia Retrieval*. Association for Computing Machinery, 2022. DOI: 10.1145/3512527.3531439.

[GLN+20] Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hürst, Minh-Triet Tran, and Klaus Schöffmann. Introduction to the Third Annual Lifelog Search Challenge (LSC'20). In *International Conference on Multimedia Retrieval*, pages 584–585. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7087-5. DOI: 10.1145/3372278.3388043.

[GSH+19] Cathal Gurrin, Klaus Schoeffmann, Joho Hideo, Duc-Tien Dang-Nguyen, Michael Riegler, and Luca Piras. *LSC '19: Proceedings of the ACM Workshop on Lifelog Search Challenge*. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6781-3.

[GSJ+18] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Duc-Tien Dang-Nguyen, Michael Riegler, and Luca Piras. *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge.* Association for Computing Machinery, 2018. ISBN: 978-1-4503-5796-8.

[GSJ+19] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Duc-Tien Dang-Nguyen, Michael Riegler, Luca Piras, Minh-Triet Tran, Jakub Lokoč, and Wolfgang Hürst. [Invited papers] Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). *ITE Transactions on Media Technology and Applications*, 7(2):46–59, 2019. DOI: 10.3169/mta.7.46.

[GSD14] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. LifeLogging: Personal Big Data. *Foundations and Trends® in Information Retrieval*, 8(1):1–125, 2014. ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000033.

[HKB+09] Josef Hallberg, Basel Kikhia, Johan Bengtsson, Stefan Sävenstedt, and Kåre Synnes. Reminiscence processes using life-log entities for persons with mild dementia. In *Workshop on Reminiscence Systems*, pages 16–21, 2009.

[HBH+04] Samira Hammiche, Salima Benbernou, Mohand-Saïd Hacid, and Athena Vakali. Semantic retrieval of multimedia data. In *International Workshop on Multimedia Databases*, pages 36–44. Association for Computing Machinery, 2004. ISBN: 978-1-58113-975-4. DOI: 10.1145/1032604.1032612.

[HCC+14] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep Speech: Scaling up end-to-end speech recognition, 2014. DOI: 10.48550/arXiv.1412.5567.

[HC04] Alexander G. Hauptmann and Michael G. Christel. Successful approaches in the TREC video retrieval evaluations. In *International Conference on Multimedia*, pages 668–675. Association for Computing Machinery, 2004. ISBN: 978-1-58113-893-1. DOI: 10.1145/1027527.1027681.

[Hel16] Silvan Heller. *Index-Partitioning in the Distributed Database System ADAMpro*. Bachelor Thesis, University of Basel, 2016.

[Hel18] Silvan Heller. *Scalable Near-duplicate Detection*. Master's thesis, University of Basel, 2018.

[HAG+22] Silvan Heller, Rahel Arnold, Ralph Gasser, Viktor Gsteiger, Mahnaz Parian-Scherb, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. Multi-modal Interactive Video Retrieval with Temporal Queries. In *MultiMedia Modeling*, pages 493–498. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_44.

[HGG+23] Silvan Heller, Ralph Gasser, Ivan Giangreco, Mahnaz Parian-Scherb, Loris Sauter, Heiko Schuldt, Florian Spiess, and Luca Rossetto. Vitrivr: a Flexible Multimodal Multimedia Retrieval System, 2023 (Unpublished, Work In Progress).

[HGI+21] Silvan Heller, Ralph Gasser, Cristina Illi, Maurizio Pasquinelli, Loris Sauter, Florian Spiess, and Heiko Schuldt. Towards Explainable Interactive Multi-modal Video Retrieval with Vitrivr. In *MultiMedia Modeling*, pages 435–440. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_41.

[HGP+21] Silvan Heller, Ralph Gasser, Mahnaz Parian-Scherb, Sanja Popovic, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. Interactive Multimodal Lifelog Retrieval with vitrivr at LSC 2021. In *Workshop on Lifelog Search Challenge*, pages 35–39. Association for Computing Machinery, 2021. DOI: 10.1145/3463948.3469062.

[HGB+22] Silvan Heller, Viktor Gsteiger, Werner Bailer, Cathal Gurrin, Björn Þór Jónsson, Jakub Lokoč, Andreas Leibetseder, František Mejzlík, Ladislav Peška, Luca Rossetto, Konstantin Schall, Klaus Schoeffmann, Heiko Schuldt, Florian Spiess, Ly-Duyen Tran, Lucia Vadicamo, Patrik Veselý, Stefanos Vrochidis, and Jiaxin Wu. Interactive video retrieval evaluation at a distance: comparing sixteen interactive video search systems in a remote setting at the 10th Video Browser Showdown. *International Journal of Multimedia Information Retrieval*, 11(1):1–18, 2022. ISSN: 2192-662X. DOI: 10.1007/s13735-021-00225-2.

[HPP+20] Silvan Heller, Mahnaz Parian, Maurizio Pasquinelli, and Heiko Schuldt. Vitrivr-Explore: Guided Multimedia Collection Exploration for Ad-hoc Video Search. In *Similarity Search and Applica-*

tions, volume 12440, pages 379–386. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-60936-8_30.

[HPG+20] Silvan Heller, Mahnaz Amiri Parian, Ralph Gasser, Loris Sauter, and Heiko Schuldt. Interactive Lifelog Retrieval with vitrivr. In *Workshop on Lifelog Search Challenge*, pages 1–6. Association for Computing Machinery, 2020. DOI: 10.1145/3379172.3391715.

[HRS+22] Silvan Heller, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Vitrivr at the Lifelog Search Challenge 2022. In *Workshop on Lifelog Search Challenge*, pages 27–31. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: 10.1145/3512729.3533003.

[HSS+20] Silvan Heller, Loris Sauter, Heiko Schuldt, and Luca Rossetto. Multi-Stage Queries and Temporal Scoring in Vitrivr. In *International Conference on Multimedia Expo Workshops*, pages 1–5. IEEE, 2020. DOI: 10.1109/ICMEW46912.2020.9105954.

[HSS23] Silvan Heller, Florian Spiess, and Heiko Schuldt. A tale of two interfaces: vitrivr at the lifelog search challenge. *Multimedia Tools and Applications*, 2023. ISSN: 1573-7721. DOI: 10.1007/s11042-023-15082-w.

[HSJ+21] Nico Hezel, Konstantin Schall, Klaus Jung, and Kai Uwe Barthel. Video Search with Sub-Image Keyword Transfer Using Existing Image Archives. In *MultiMedia Modeling*, pages 484–489. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_49.

[HSJ+22] Nico Hezel, Konstantin Schall, Klaus Jung, and Kai Uwe Barthel. Efficient Search and Browsing of Large-Scale Video Collections with Vibro. In *MultiMedia Modeling*, pages 487–492. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_43.

[HK92] Kyoji Hirata and Toshikazu Kato. Query by visual example. In *Advances in Database Technology*, pages 56–71. Springer, 1992. ISBN: 978-3-540-47003-8. DOI: 10.1007/BFb0032423.

[HDN+22] Khanh Ho, Vu Xuan Dinh, Hong-Quang Nguyen, Khiem Le, Khang Dinh Tran, Tien Do, Tien-Dung Mai, Thanh Duc Ngo, and Duy-Dinh Le. UIT at VBS 2022: An Unified and Interactive Video

Retrieval System with Temporal Search. In *MultiMedia Modeling*, pages 556–561. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_54.

[HTN+22] Nhat Hoang-Xuan, Hoang-Phuc Trang-Trung, E-Ro Nguyen, Thanh-Cong Le, Mai-Khiem Tran, Tu-Khiem Le, Van-Tu Ninh, Cathal Gurrin, and Minh-Triet Tran. Flexible Interactive Retrieval SysTem 3.0 for Visual Lifelog Exploration at LSC 2022. In *Workshop on Lifelog Search Challenge*, pages 20–26. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: 10.1145/3512729.3533013.

[HCC+15] Min-Chun Hu, Chi-Wen Chen, Wen-Huang Cheng, Che-Han Chang, Jui-Hsin Lai, and Ja-Ling Wu. Real-Time Human Movement Retrieval and Assessment With Kinect Sensor. *IEEE Transactions on Cybernetics*, 45(4):742–753, 2015. ISSN: 2168-2275. DOI: 10.1109/TCYB.2014.2335540.

[HXL+11] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A Survey on Visual Content-Based Video Indexing and Retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819, 2011. ISSN: 1558-2442. DOI: 10.1109/TSMCC.2011.2109710.

[Hul93] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Conference on Research and Development in Information Retrieval*, pages 329–338. Association for Computing Machinery, 1993. ISBN: 978-0-89791-605-9. DOI: 10.1145/160688.160758.

[HA99] Jane Hunter and Liz Armstrong. A comparison of schemas for video metadata representation. *Computer Networks*, 31(11):1431–1451, 1999. ISSN: 1389-1286. DOI: 10.1016/S1389-1286(99)00053-5.

[HGS19] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. VideoGraph: Recognizing Minutes-Long Human Activities in Videos, 2019. DOI: 10.48550/arXiv.1905.05143.

[IM18] Mostafa S. Ibrahim and Greg Mori. Hierarchical Relational Networks for Group Activity Recognition and Retrieval. In *European Conference on Computer Vision*, pages 721–736, 2018.

[Ill21] Cristina Illi. Why, Vitrivr? Understanding Results in Multimedia Retrieval. Master Project, University of Basel, 2021.

[IMP+19] Bogdan Ionescu, Henning Müller, Renaud Péteri, Yashin Dicente Cid, Vitali Liauchuk, Vassili Kovalev, Dzmitri Klimuk, Aleh Tarasau, Asma Ben Abacha, Sadid A. Hasan, Vivek Datla, Joey Liu, Dina Demner-Fushman, Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Minh-Triet Tran, Mathias Lux, Cathal Gurrin, Obioma Pelka, Christoph M. Friedrich, Alba Garcìa Seco de Herrera, Narciso Garcia, Ergina Kavallieratou, Carlos Roberto del Blanco, Carlos Cuevas, Nikos Vasillopoulos, Konstantinos Karampidis, Jon Chamberlain, Adrian Clark, and Antonio Campello. ImageCLEF 2019: Multimedia Retrieval in Medicine, Lifelogging, Security and Nature. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 358–386. Springer International Publishing, 2019. ISBN: 978-3-030-28577-7. DOI: 10.1007/978-3-030-28577-7_28.

[IMP+22] Bogdan Ionescu, Henning Müller, Renaud Péteri, Johannes Rückert, Asma Ben Abacha, Alba G. Seco de Herrera, Christoph M. Friedrich, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Serge Kozlovski, Yashin Dicente Cid, Vassili Kovalev, Liviu-Daniel Ştefan, Mihai Gabriel Constantin, Mihai Dogariu, Adrian Popescu, Jérôme Deshayes-Chossart, Hugo Schindler, Jon Chamberlain, Antonio Campello, and Adrian Clark. Overview of the ImageCLEF 2022: Multimedia Retrieval in Medical, Social Media and Nature Applications. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 541–564. Springer International Publishing, 2022. ISBN: 978-3-031-13643-6. DOI: 10.1007/978-3-031-13643-6_31.

[JSG06] Alejandro Jaimes, Nicu Sebe, and Daniel Gatica-Perez. Human-centered computing: a multimedia perspective. In *International Conference on Multimedia*, pages 855–864. Association for Computing Machinery, 2006. ISBN: 978-1-59593-447-5. DOI: 10.1145/1180639.1180829.

[JDS11] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. ISSN: 1939-3539. 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 *MultiMedia Modeling*, pages 290–302. Springer International Publishing, 2016. ISBN: 978-3-319-27674-8. DOI: 10.1007/978-3-319-27674-8_26.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.

[KW07] Vaiva Kalnikaité and Steve Whittaker. Software or wetware? discovering when and why people use digital prosthetic memory. In *Conference on Human Factors in Computing Systems*, pages 71–80. Association for Computing Machinery, 2007. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240635.

[KHZ09] Martin Kampel, Reinhold Huber-Mörk, and Maia Zaharieva. Image-Based Retrieval and Identification of Ancient Coins. *IEEE Intelligent Systems*, 24(2):26–34, 2009. ISSN: 1941-1294. DOI: 10.1109/MIS.2009.29.

[KBN15] Nattiya Kanhabua, Roi Blanco, and Kjetil Nørvåg. Temporal Information Retrieval. *Foundations and Trends® in Information Retrieval*, 9(2):91–208, 2015. ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000043.

[KAG10] Rajkumar Kannan, Frederic Andres, and Christian Guetl. DanVideo: an MPEG-7 authoring and retrieval system for dance videos. *Multimedia Tools and Applications*, 46(2):545–572, 2010. ISSN: 1573-7721. DOI: 10.1007/s11042-009-0388-3.

[KLBV10] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnović. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390, 2010. ISSN: 1558-0660. DOI: 10.1109/TMC.2010.99.

[KL16] Owen Kaser and Daniel Lemire. Compressed bitmap indexes: beyond unions and intersections. *Software: Practice and Experience*, 46(2):167–198, 2016. ISSN: 00380644. DOI: 10.1002/spe.2289.

[KKO+92] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database-query by visual example. In *IAPR International Conference on Pattern Recognition*, pages 530–533, 1992. DOI: 10.1109/ICPR.1992.201616.

[KGC11] Mostafa Keikha, Shima Gerani, and Fabio Crestani. Time-based relevance models. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1087–1088. Association for Computing Machinery, 2011. ISBN: 978-1-4503-0757-4. DOI: 10.1145/2009916.2010062.

[Kel09] Diane Kelly. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends® in Information Retrieval*, 3(1–2):1–224, 2009. ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000012.

[KPZ+04] Eamonn J. Keogh, Themis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. Indexing large human-motion databases. In *International Conference on Very Large Data Bases*, pages 780–791. Morgan Kaufmann, 2004. DOI: 10.1016/B978-012088469-8.50069-3.

[KJL+21] Omar Shahbaz Khan, Björn Þór Jónsson, Mathias Larsen, Liam Poulsen, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. Exquisitor at the Video Browser Showdown 2021: Relationships Between Semantic Classifiers. In *MultiMedia Modeling*, pages 410–416. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_37.

[KJR+20] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. Interactive Learning for Multimedia at Large. In *Advances in Information*

Retrieval, pages 495–510. Springer International Publishing, 2020. ISBN: 978-3-030-45439-5. DOI: 10.1007/978-3-030-45439-5_33.

[KSJ+22] Omar Shahbaz Khan, Ujjwal Sharma, Björn Þór Jónsson, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. Exquisitor at the Video Browser Showdown 2022. In *MultiMedia Modeling*, pages 511–517. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_47.

[KTS+17] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. What Snippet Size is Needed in Mobile Web Search? In *Conference on Human Information Interaction and Retrieval*, pages 97–106. Association for Computing Machinery, 2017. ISBN: 978-1-4503-4677-1. DOI: 10.1145/3020165.3020173.

[KLP13] Yelin Kim, Honglak Lee, and Emily Mower Provost. Deep learning for robust feature generation in audiovisual emotion recognition. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3687–3691, 2013. DOI: 10.1109/ICASSP.2013.6638346.

[Koh90] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. ISSN: 1558-2256. DOI: 10.1109/5.58325.

[KNK+99] N. Kosugi, Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima. Music retrieval by humming-using similarity retrieval over high dimensional feature vector space. In *Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 404–407, 1999. DOI: 10.1109/PACRIM.1999.799561.

[KK09] Anna Khudyak Kozorovitzky and Oren Kurland. From "Identical" to "Similar": Fusing Retrieved Lists Based on Inter-document Similarities. In *Advances in Information Retrieval Theory*, pages 212–223. Springer, 2009. ISBN: 978-3-642-04417-5. DOI: 10.1007/978-3-642-04417-5_19.

[KVM+20] Miroslav Kratochvíl, Patrik Veselý, František Mejzlík, and Jakub Lokoč. SOM-Hunter: Video Browsing with Relevance-to-SOM Feedback Loop. In *MultiMedia Modeling*, pages 790–795. Springer International Publishing, 2020. ISBN: 978-3-030-37734-2. DOI: 10.1007/978-3-030-37734-2_71.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. ISSN: 0001-0782. DOI: 10.1145/3065386.

[KGU10] Onur Küçüktunç, Uğur Güdükbay, and Özgür Ulusoy. Fuzzy color histogram-based video segmentation. *Computer Vision and Image Understanding*, 114(1):125–134, 2010. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2009.09.008.

[KC18] Oren Kurland and J. Shane Culpepper. Fusion in Information Retrieval: SIGIR 2018 Half-Day Tutorial. In *International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1383–1386. Association for Computing Machinery, 2018. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3210186.

[LKO02] J. Laaksonen, M. Koskela, and E. Oja. PicSOM-self-organizing image retrieval with MPEG-7 content descriptors. *IEEE Transactions on Neural Networks*, 13(4):841–853, 2002. ISSN: 1941-0093. DOI: 10.1109/TNN.2002.1021885.

[LZM18] Katrien Laenen, Susana Zoghbi, and Marie-Francine Moens. Web Search of Fashion Items with Multimodal Querying. In *International Conference on Web Search and Data Mining*, pages 342–350. Association for Computing Machinery, 2018. ISBN: 978-1-4503-5581-0. DOI: 10.1145/3159652.3159716.

[LLC+15] Kuan-Ting Lai, Dong Liu, Shih-Fu Chang, and Ming-Syan Chen. Learning Sample Specific Weights for Late Fusion. *IEEE Transactions on Image Processing*, 24(9):2772–2783, 2015. ISSN: 1941-0042. DOI: 10.1109/TIP.2015.2423560.

[LFF22] Walter Leimgruber, Peter Fornaro, and Ulrike Felsing. PIA, 2022. URL: https://about.participatory-archives.ch.

[LKA10] Daniel Lemire, Owen Kaser, and Kamel Aouiche. Sorting improves word-aligned bitmap indexes. *Data & Knowledge Engineering*, 69(1):3–28, 2010. ISSN: 0169023X. DOI: 10.1016/j.datak.2009.08.006.

[LLG+15] Bo Li, Yijuan Lu, Azeem Ghumman, Bradley Strylowski, Mario Gutierrez, Safiyah Sadiq, Scott Forster, Natacha Feola, and Travis Bugerin. KinectSBR: A Kinect-Assisted 3D Sketch-Based 3D

Model Retrieval System. In *International Conference on Multimedia Retrieval*, pages 655–656. Association for Computing Machinery, 2015. ISBN: 978-1-4503-3274-3. DOI: 10.1145/2671188.2749350.

[LXY+19] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. W2VV++: Fully Deep Learning for Ad-hoc Video Search. In *International Conference on Multimedia*, pages 1786–1794. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3350906.

[LL18] Yi Li and Wenzhao Li. A survey of sketch-based image retrieval. *Machine Vision and Applications*, 29(7):1083–1100, 2018. ISSN: 1432-1769. DOI: 10.1007/s00138-018-0953-8.

[LMR+18] Shangsong Liang, Ilya Markov, Zhaochun Ren, and Maarten de Rijke. Manifold Learning for Rank Aggregation. In *World Wide Web Conference*, pages 1735–1744. Association for Computing Machinery, 2018. ISBN: 978-1-4503-5639-8. DOI: 10.1145/3178876.3186085.

[LRL+10] Andreas Lingnau, Ian Ruthven, Monica Landoni, and Frans van der Sluis. Interactive Search Interfaces for Young Children - The PuppyIR Approach. In *IEEE International Conference on Advanced Learning Technologies*, pages 389–390, 2010. DOI: 10.1109/ICALT.2010.111.

[LLC19] Tzu-En Liu, Shih-Hung Liu, and Berlin Chen. A Hierarchical Neural Summarization Framework for Spoken Documents. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7185–7189, 2019. DOI: 10.1109/ICASSP.2019.8683758.

[LAB+23] Jakub Lokoč, Stelios Andreadis, Werner Bailer, Aaron Duane, Cathal Gurrin, Ma, Nicola Messina, Thao-Nhu Nguyen, Ladislav Peška, Luca Rossetto, Loris Sauter, Konstantin Schall, Klaus Schoeffmann, Omar Shahbaz Khan, Florian Spiess, Lucia Vadicamo, and Stefanos Vrochidis. Interactive Video Retrieval in the Age of Effective Joint Embedding Deep Models: Lessons from the 11th VBS. *Multimedia Systems*, 2023 (Under Review).

[LBB+22] Jakub Lokoč, Werner Bailer, Kai Uwe Barthel, Cathal Gurrin, Silvan Heller, Björn þór Jónsson, Ladislav Peška, Luca Rossetto, Klaus Schoeffmann, Lucia Vadicamo, Stefanos Vrochidis, and

Jiaxin Wu. A Task Category Space for User-Centric Comparative Multimedia Search Evaluations. In *MultiMedia Modeling*, pages 193–204. Springer International Publishing, 2022. ISBN: 978-3-030-98358-1. DOI: 10.1007/978-3-030-98358-1_16.

[LBS+21] Jakub Lokoč, Jana Bátoryová, Dominik Smrž, and Marek Dobranský. Video Search with Collage Queries. In *MultiMedia Modeling*, pages 429–434. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_40.

[LKS20] Jakub Lokoč, Gregor Kovalčík, and Tomáš Souček. VIRET at Video Browser Showdown 2020. In *MultiMedia Modeling*, pages 784–789. Springer International Publishing, 2020. ISBN: 978-3-030-37734-2. DOI: 10.1007/978-3-030-37734-2_70.

[LKS+19a] Jakub Lokoč, Gregor Kovalčik, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. A Framework for Effective Known-item Search in Video. In *International Conference on Multimedia*, pages 1777–1785. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6889-6. DOI: 10.1145/3343031.3351046.

[LKS+19b] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. VIRET: A Video Retrieval Tool for Interactive Known-item Search. In *International Conference on Multimedia Retrieval*, pages 177–181. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6765-3. DOI: 10.1145/3323873.3325034.

[LMS+22] Jakub Lokoč, František Mejzlík, Tomáš Souček, Patrik Dokoupil, and Ladislav Peška. Video Search with Context-Aware Ranker and Relevance Feedback. In *MultiMedia Modeling*, pages 505–510. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_46.

[LSV+20] Jakub Lokoć, Tomáš Soućek, Patrik Veselý, František Mejzlík, Jiaqi Ji, Chaoxi Xu, and Xirong Li. A W2VV++ Case Study with Automated and Interactive Text-to-Video Retrieval. In *International Conference on Multimedia*, pages 2553–2561. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7988-5. DOI: 10.1145/3394171.3414002.

[LVM+21] Jakub Lokoč, Patrik Veselý, František Mejzlík, Gregor Kovalčík, Tomáš Souček, Luca Rossetto, Klaus Schoeffmann, Werner Bailer,

Cathal Gurrin, Loris Sauter, Jaeyub Song, Stefanos Vrochidis, Jiaxin Wu, and Björn þóR Jónsson. Is the Reign of Interactive Search Eternal? Findings from the Video Browser Showdown 2020. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(3):91:1–91:26, 2021. ISSN: 1551-6857. DOI: 10.1145/3445031.

[LRH+16] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. LIRE: open source visual information retrieval. In *International Conference on Multimedia Systems*, pages 1–4. Association for Computing Machinery, 2016. ISBN: 978-1-4503-4297-1. DOI: 10.1145/2910017.2910630.

[MLK+18] Jiaxin Mao, Yiqun Liu, Noriko Kando, Cheng Luo, Min Zhang, and Shaoping Ma. Investigating Result Usefulness in Mobile Search. In *Advances in Information Retrieval*, pages 223–236. Springer International Publishing, 2018. ISBN: 978-3-319-76941-7. DOI: 10.1007/978-3-319-76941-7_17.

[Mar06] Gary Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, 2006. DOI: 10.1145/1121949.1121979.

[MMG+18] Foteini Markatopoulou, Anastasia Moumtzidou, Damianos Galanopoulos, Konstantinos Avgerinakis, Stelios Andreadis, Ilias Gialampoukidis, Stavros Tachos, Stefanos Vrochidis, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Patras. ITI-CERTH participation in TRECVID 2017, 2018. DOI: 10.5281/zenodo.1183440.

[MS98] Marjo Markkula and Eero Sormunen. Searching for Photos - Journalists' Practices in Pictorial IR. In *Challenge of Image Retrieval*. BCS Learning & Development, 1998. DOI: 10.14236/ewic/CIR1998.8.

[MHG10] Michael McCandless, Erik Hatcher, and Otis Gospodnetić. *Lucene in Action*, volume 2. Manning Greenwich, 2010. ISBN: 978-1-933988-17-7.

[MFE+21] Nicola Messina, Fabrizio Falchi, Andrea Esuli, and Giuseppe Amato. Transformer Reasoning Network for Image- Text Matching and Retrieval. In *International Conference on Pattern Recognition*, pages 5222–5229, 2021. DOI: 10.1109/ICPR48806.2021.9413172.

[MKS20] Pascal Mettes, Dennis C. Koelma, and Cees G. M. Snoek. Shuffled ImageNet Banks for Video Event Detection and Search. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(2):44:1–44:21, 2020. ISSN: 1551-6857. DOI: 10.1145/3377875.

[Mof13] Alistair Moffat. Seven Numeric Properties of Effectiveness Metrics. In *Information Retrieval Technology*, pages 1–12. Springer, 2013. ISBN: 978-3-642-45068-6. DOI: 10.1007/978-3-642-45068-6_1.

[Moo50] Calvin N Mooers. *The Theory of Digital Handling of Non-Numerical Information and Its Implications to Machine Economics*, number 48. Zator Company, 1950.

[MTY16] Bilel Moulahi, Lynda Tamine, and Sadok Ben Yahia. When time meets information retrieval: Past proposals, current plans and future trends. *Journal of Information Science*, 42(6):725–747, 2016. ISSN: 0165-5515. DOI: 10.1177/0165551515607277.

[Moz22] Mozilla. DeepSpeech: An Open-Source Speech-To-Text Engine by Mozilla. Mozilla, 2022. URL: https://github.com/mozilla/DeepSpeech.

[MMB+04] Henning Müller, Nicolas Michoux, David Bandon, and Antoine Geissbuhler. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, 2004. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2003.11.024.

[MD15] Jaap M. J. Murre and Joeri Dros. Replication and analysis of ebbinghaus' forgetting curve. *PLOS ONE*, 10(7):1–23, 2015. DOI: 10.1371/journal.pone.0120644.

[NYA+21] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention Bottlenecks for Multimodal Fusion. In *Advances in Neural Information Processing Systems*, volume 34, pages 14200–14213. Curran Associates, Inc., 2021.

[Nai04] Omar Naim. The Final Cut, 2004.

[Nem20] Kalthoum Nemmour. Why, Vitrivr? Understanding Results in Multimedia Retrieval. Bachelor Project, University of Basel, 2020.

[NLZ+18] Phuong Anh Nguyen, Yi-Jie Lu, Hao Zhang, and Chong-Wah Ngo. Enhanced VIREO KIS at VBS 2018. In *MultiMedia Modeling*, pages 407–412. Springer International Publishing, 2018. ISBN: 978-3-319-73600-6. DOI: 10.1007/978-3-319-73600-6_42.

[NWN+20] Phuong Anh Nguyen, Jiaxin Wu, Chong-Wah Ngo, Danny Francis, and Benoit Huet. VIREO @ Video Browser Showdown 2020. In *MultiMedia Modeling*, pages 772–777. Springer International Publishing, 2020. ISBN: 978-3-030-37734-2. DOI: 10.1007/978-3-030-37734-2_68.

[NLN+22] Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Sinéad Smyth, Annalina Caputo, and Cathal Gurrin. LifeSeeker 4.0: An Interactive Lifelog Search Engine for LSC'22. In *Workshop on Lifelog Search Challenge*, pages 14–19. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: 10.1145/3512729.3533014.

[NBE+93] Carlton Wayne Niblack, Ron Barber, Will Equitz, Myron D. Flickner, Eduardo H. Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, and Gabriel Taubin. QBIC project: querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187. SPIE, 1993. DOI: 10.1117/12.143648.

[NLZ+20] Van-Tu Ninh, Tu-Khiem Le, Liting Zhou, Luca Piras, Michael Alexander Riegler, Pål Halvorsen, Mathias Lux, Minh-Triet Tran, Cathal Gurrin, and Duc Tien Dang Nguyen. Overview of ImageCLEF Lifelog 2020: Lifelog Moment Retrieval and Sport Performance Lifelog. *CEUR Workshop Proceedings*, 2020. ISSN: 1613-0073.

[NMB+14] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado, and Jeffrey Dean. Zero-Shot Learning by Convex Combination of Semantic Embeddings, 2014. DOI: 10.48550/arXiv.1312.5650.

[NED+18] Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences of the United States of America*, 115(11):2600–2606, 2018. ISSN: 0027-8424. DOI: 10.1073/pnas.1708274114.

[OS95] V.E. Ogle and M. Stonebraker. Chabot: retrieval from a relational database of images. *Computer*, 28(9):40–48, 1995. ISSN: 1558-0814. DOI: 10.1109/2.410150.

[PWR+21] Mahnaz Parian, Claire Walzer, Luca Rossetto, Silvan Heller, Stéphane Dupont, and Heiko Schuldt. Gesture of Interest: Gesture Search for Multi-Person, Multi-Perspective TV Footage. In *International Conference on Content-Based Multimedia Indexing*, pages 1–6, 2021. DOI: 10.1109/CBMI50038.2021.9461887.

[Par21] Mahnaz Parian-Scherb. *Gesture Similarity Learning and Retrieval in Large-Scale Real-world Video Collections*. Thesis, University of Basel, 2021. DOI: 10.5451/unibas-ep84855.

[PP07] Devi Parikh and Robi Polikar. An Ensemble-Based Incremental Learning Approach to Data Fusion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):437–450, 2007. ISSN: 1941-0492. DOI: 10.1109/TSMCB.2006.883873.

[PJW00] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *ACM Workshops on Multimedia*, pages 51–54. Association for Computing Machinery, 2000. ISBN: 978-1-58113-311-0. DOI: 10.1145/357744.357758.

[PC11] Han-Saem Park and Sung-Bae Cho. A personalized summarization of video life-logs from an indoor multi-camera system using a fuzzy rule-based system with domain knowledge. *Information Systems*, 36(8):1124–1134, 2011. ISSN: 0306-4379. DOI: 10.1016/j.is.2011.04.005.

[Pas20] Maurizio Pasquinelli. *Using Self-Organizing Maps to Explore and Query Multimedia Collections*. Bachelor Thesis, University of Basel, 2020.

[PT16] Daniel Carlos Pedronette and Ricardo Da Torres. Combining re-ranking and rank aggregation methods for image retrieval. *Multimedia Tools and Applications*, 75(15):9121–9144, 2016. ISSN: 1380-7501. DOI: 10.1007/s11042-015-3044-0.

[PPS96] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996. ISSN: 1573-1405. DOI: 10.1007/BF00123143.

[PKS+21] Ladislav Peška, Gregor Kovalčík, Tomáš Souček, Vít Škrhák, and Jakub Lokoč. W2VV++ BERT Model at VBS 2021. In *MultiMedia Modeling*, pages 467–472. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_46.

[Pet22] Simon Peterhans. *Multi-Platform Data Collection for Social Media Analytics*. Master's thesis, University of Basel, 2022.

[PSS+22] Simon Peterhans, Loris Sauter, Florian Spiess, and Heiko Schuldt. Automatic Generation of Coherent Image Galleries in Virtual Reality. In *Linking Theory and Practice of Digital Libraries*, pages 282–288. Springer International Publishing, 2022. ISBN: 978-3-031-16802-4. DOI: 10.1007/978-3-031-16802-4_23.

[PPE+21] Robin Piening, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, Sarah Prange, Philippe Schröder, and Florian Alt. Looking for Info: Evaluation of Gaze Based Information Retrieval in Augmented Reality. In *Human-Computer Interaction – INTERACT 2021*, pages 544–565. Springer International Publishing, 2021. ISBN: 978-3-030-85623-6. DOI: 10.1007/978-3-030-85623-6_32.

[PG17] Luca Piras and Giorgio Giacinto. Information fusion in content based image retrieval: A comprehensive overview. *Information Fusion*, 37:50–60, 2017. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2017.01.003.

[PC95] Peter Pirolli and Stuart K. Card. Information foraging in information access environments. In *Human Factors in Computing Systems*, pages 51–58. ACM/Addison-Wesley, 1995. DOI: 10.1145/223904.223911.

[PMM+17] Nikiforos Pittaras, Foteini Markatopoulou, Vasileios Mezaris, and Ioannis Patras. Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks. In *MultiMedia Modeling*, pages 102–114. Springer International Publishing, 2017. ISBN: 978-3-319-51811-4. DOI: 10.1007/978-3-319-51811-4_9.

[Pop21] Sanja Popovic. *Location-Based Queries and Query Representation for the Lifelog Search Challenge 2021*. Bachelor Thesis, University of Basel, 2021.

[PCI+03] James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. TimeML: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering, 2003 AAAI Spring Symposium*, pages 28–34. AAAI Press, 2003.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. DOI: 10.48550/arXiv.2103.00020.

[RKX+22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *OpenAI Blog*, 2022.

[RPG+21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation, 2021. DOI: 10.48550/arXiv.2102.12092.

[Ray03] Eric S. Raymond. *The Art of Unix Programming*. Addison-Wesley Professional, 2003.

[RS03] M. Elena Renda and Umberto Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *ACM Symposium on Applied Computing*, pages 841–846. Association for Computing Machinery, 2003. ISBN: 978-1-58113-624-1. DOI: 10.1145/952532.952698.

[RSS+21] Laura Rettig, Shaban Shabani, Loris Sauter, Philippe Cudré-Mauroux, Maria Sokhn, and Heiko Schuldt. City-Stories: Combining Entity Linking, Multimedia Retrieval, and Crowdsourcing to Make Historical Data Accessible. In *Web Engineering*, pages 521–524. Springer International Publishing, 2021. ISBN: 978-3-030-74296-6. DOI: 10.1007/978-3-030-74296-6_43.

[RAR+19] Jerome Revaud, Jon Almazan, Rafael Rezende, and Cesar De Souza. Learning With Average Precision: Training Image Retrieval With a Listwise Loss. In *International Conference on Computer Vision*, pages 5106–5115, 2019. DOI: 10.1109/ICCV.2019.00521.

[RTN22] Ricardo Ribeiro, Alina Trifan, and António J. R. Neves. Lifelog Retrieval From Daily Digital Data: Narrative Review. *JMIR mHealth and uHealth*, 10(5):e30517, 2022. DOI: 10.2196/30517.

[RRT22] Antonio M. Rinaldi, Cristiano Russo, and Cristian Tommasino. An Approach Based on Linked Open Data and Augmented Reality for Cultural Heritage Content-Based Information Retrieval. In *Computational Science and Its Applications*, pages 99–112. Springer International Publishing, 2022. ISBN: 978-3-031-10450-3. DOI: 10.1007/978-3-031-10450-3_8.

[Rob00] David Robins. Interactive information retrieval: Context and basic notions. *Informing Sci. Int. J. an Emerg. Transdiscipl.*, 3:57–62, 2000. DOI: 10.28945/577.

[RM19] Haggai Roitman and Yosi Mass. Utilizing Passages in Fusion-based Document Retrieval. In *International Conference on Theory of Information Retrieval*, pages 59–66. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6881-0. DOI: 10.1145/3341981.3344212.

[Ros18] Luca Rossetto. *Multi-Modal Video Retrieval*. PhD thesis, University of Basel, 2018. DOI: 10.5451/unibas-006859522.

[RBB21] Luca Rossetto, Werner Bailer, and Abraham Bernstein. Considering Human Perception and Memory in Interactive Multimedia Retrieval Evaluations. In *MultiMedia Modeling*, pages 605–616. Springer International Publishing, 2021. ISBN: 978-3-030-67832-6. DOI: 10.1007/978-3-030-67832-6_49.

[RBA+20] Luca Rossetto, Matthias Baumgartner, Narges Ashena, Florian Ruosch, Romana Pernischová, and Abraham Bernstein. LifeGraph: A Knowledge Graph for Lifelogs. In *Workshop on Lifelog Search Challenge*, pages 13–17. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7136-0. DOI: 10.1145/3379172.3391717.

[RBG+21] Luca Rossetto, Matthias Baumgartner, Ralph Gasser, Lucien Heitz, Ruijie Wang, and Abraham Bernstein. Exploring Graph-querying approaches in LifeGraph. In *Workshop on Lifelog Search Challenge*, pages 7–10. Association for Computing Machinery, 2021. ISBN: 978-1-4503-8533-6. DOI: 10.1145/3463948.3469068.

[RGH+19] Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Amiri Parian, and Heiko Schuldt. Retrieval of Structured and Unstructured Data with vitrivr. In *Workshop on Lifelog Search Challenge*, pages 27–31. Association for Computing Machinery, 2019. DOI: 10.1145/3326460.3329160.

[RGH+21] Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Parian-Scherb, Loris Sauter, Florian Spiess, Heiko Schuldt, Ladislav Peška, Tomáš Souček, Miroslav Kratochvíl, František Mejzlík, Patrik Veselý, and Jakub Lokoč. On the User-Centric Comparative Remote Evaluation of Interactive Video Search Systems. *IEEE MultiMedia*, 28(4):18–28, 2021. ISSN: 1941-0166. DOI: 10.1109/MMUL.2021.3066779.

[RGL+21] Luca Rossetto, Ralph Gasser, Jakub Lokoč, Werner Bailer, Klaus Schoeffmann, Bernd Muenzer, Tomáš Souček, Phuong Anh Nguyen, Paolo Bolettieri, Andreas Leibetseder, and Stefanos Vrochidis. Interactive Video Retrieval in the Age of Deep Learning – Detailed Evaluation of VBS 2019. *IEEE Transactions on Multimedia*, 23:243–256, 2021. ISSN: 1941-0077. DOI: 10.1109/TMM.2020.2980944.

[RGS+21] Luca Rossetto, Ralph Gasser, Loris Sauter, Abraham Bernstein, and Heiko Schuldt. A System for Interactive Multimedia Retrieval Evaluations. In *MultiMedia Modeling*, pages 385–390. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_33.

[RGS19] Luca Rossetto, Ralph Gasser, and Heiko Schuldt. Query by Semantic Sketch, 2019. DOI: 10.48550/arXiv.1909.12526.

[RGH+16] Luca Rossetto, Ivan Giangreco, Silvan Heller, Claudiu Tănase, Heiko Schuldt, Stéphane Dupont, Omar Seddati, Metin Sezgin, Ozan Can Altıok, and Yusuf Sahillioğlu. IMOTION – Searching for Video Sequences Using Multi-Shot Sketch Queries. In *MultiMedia Modeling*, pages 377–382. Springer International Publishing, 2016. ISBN: 978-3-319-27674-8. DOI: 10.1007/978-3-319-27674-8_36.

[RGS14] Luca Rossetto, Ivan Giangreco, and Heiko Schuldt. Cineast: A Multi-feature Sketch-Based Video Retrieval Engine. In *IEEE Inter-*

national Symposium on Multimedia, pages 18–23. IEEE, 2014. DOI: 10.1109/ISM.2014.38.

[RGS+15] Luca Rossetto, Ivan Giangreco, Heiko Schuldt, Stéphane Dupont, Omar Seddati, Metin Sezgin, and Yusuf Sahillioğlu. IMOTION — A Content-Based Video Retrieval Engine. In *MultiMedia Modeling*, pages 255–260. Springer International Publishing, 2015. ISBN: 978-3-319-14442-9. DOI: 10.1007/978-3-319-14442-9_24.

[RGT+16] Luca Rossetto, Ivan Giangreco, Claudiu Tanase, and Heiko Schuldt. Vitrivr: A Flexible Retrieval Stack Supporting Multiple Query Modes for Searching in Multimedia Collections. In *International Conference on Multimedia*, pages 1183–1186. Association for Computing Machinery, 2016. ISBN: 978-1-4503-3603-1. DOI: 10.1145/2964284.2973797.

[RGT+17] Luca Rossetto, Ivan Giangreco, Claudiu Tănase, Heiko Schuldt, Stéphane Dupont, and Omar Seddati. Enhanced Retrieval and Browsing in the IMOTION System. In *MultiMedia Modeling*, pages 469–474. Springer International Publishing, 2017. ISBN: 978-3-319-51814-5. DOI: 10.1007/978-3-319-51814-5_43.

[RPG+19] Luca Rossetto, Mahnaz Amiri Parian, Ralph Gasser, Ivan Giangreco, Silvan Heller, and Heiko Schuldt. Deep Learning-Based Concept Detection in vitrivr. In *MultiMedia Modeling*, volume 11296, pages 616–621. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-05716-9_55.

[RSB21] Luca Rossetto, Klaus Schoeffmann, and Abraham Bernstein. Insights on the V3C2 Dataset, 2021. DOI: 10.48550/arXiv.2105.01475.

[RSA+19] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A. Butt. V3C – A Research Video Collection. In *MultiMedia Modeling*, pages 349–360. Springer International Publishing, 2019. ISBN: 978-3-030-05710-7. DOI: 10.1007/978-3-030-05710-7_29.

[RHC99] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, 1999. ISSN: 1047-3203. DOI: 10.1006/jvci.1999.0413.

[RHO+98] Yong Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998. ISSN: 1558-2205. DOI: 10.1109/76.718510.

[SCS+22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, 2022. DOI: 10.48550/arXiv.2205.11487.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. ISSN: 0001-0782. DOI: 10.1145/361219.361220.

[Sal68] Gerard Salton. *Automatic Information Organization and Retrieval*. McGraw Hill, 1968. ISBN: 978-0-07-054485-7.

[Sal89] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989. ISBN: 0-201-12227-8.

[SFW83] Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983. ISSN: 0001-0782. DOI: 10.1145/182.358466.

[SC12] Mark Sanderson and W. Bruce Croft. The History of Information Retrieval Research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012. ISSN: 1558-2256. DOI: 10.1109/JPROC.2012.2189916.

[SBH+16] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 35(4):119:1–119:12, 2016. ISSN: 0730-0301. DOI: 10.1145/2897824.2925954.

[SGB+22] Loris Sauter, Ralph Gasser, Abraham Bernstein, Heiko Schuldt, and Luca Rossetto. An Asynchronous Scheme for the Distributed Evaluation of Interactive Multimedia Retrieval. In *International Workshop on Interactive Multimedia Retrieval*, pages 33–39. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9497-0. DOI: 10.1145/3552467.3554797.

[SGH+23] Loris Sauter, Ralph Gasser, Silvan Heller, Luca Rossetto, Colin Saladin, Florian Spiess, and Heiko Schuldt. Exploring Effective Interactive Text-Based Video Search in vitrivr. In *MultiMedia Modeling*, pages 646–651. Springer International Publishing, 2023. ISBN: 978-3-031-27077-2. DOI: 10.1007/978-3-031-27077-2_53.

[SPG+20] Loris Sauter, Mahnaz Amiri Parian, Ralph Gasser, Silvan Heller, Luca Rossetto, and Heiko Schuldt. Combining Boolean and Multimedia Retrieval in vitrivr for Large-Scale Video Search. In *MultiMedia Modeling*, volume 11962, pages 760–765. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-37734-2_66.

[SRS18] Loris Sauter, Luca Rossetto, and Heiko Schuldt. Exploring Cultural Heritage in Augmented Reality with GoFind! In *IEEE International Conference on Artificial Intelligence and Virtual Reality*, pages 187–188. IEEE, 2018. DOI: 10.1109/AIVR.2018.00041.

[Sch80] Hans-Jörg Schek. Methods for the Administration of Textual Data in Database Systems. *Conference on Research and Development in Information Retrieval*:218–235, 1980.

[Sch21a] Maurice Schleußinger. Information retrieval interfaces in virtual reality—A scoping review focused on current generation technology. *PLOS ONE*, 16(2):e0246398, 2021. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0246398.

[Sch19] Klaus Schoeffmann. Video Browser Showdown 2012-2019: A Review. In *International Conference on Content-Based Multimedia Indexing*, pages 1–4. IEEE, 2019. DOI: 10.1109/CBMI.2019.8877397.

[Sch21b] Klaus Schoeffmann. VBS 2021 overview, Youtube, 2021. URL: https://www.youtube.com/watch?v=8Kg_5BQon9I&t=587s.

[SBL+16] Klaus Schoeffmann, Christian Beecks, Mathias Lux, Merih Seran Uysal, and Thomas Seidl. Content-based retrieval in videos from laparoscopic surgery. In *Medical Imaging 2016: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 9786, pages 562–571. SPIE, 2016. DOI: 10.1117/12.2216864.

[Sch18] Heiko Schuldt. Multitier Architecture. In *Encyclopedia of Database Systems*, pages 2443–2446. Springer, 2018. ISBN: 978-1-4614-8265-9. DOI: 10.1007/978-1-4614-8265-9_652.

[SDM17] Omar Seddati, Stéphane Dupont, and Saïd Mahmoudi. DeepSketch 3. *Multimedia Tools and Applications*, 76(21):22333–22359, 2017. ISSN: 1573-7721. DOI: 10.1007/s11042-017-4799-2.

[SCIG+21] K Selçuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro. *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. 2021. DOI: 10.1007/978-3-030-85251-1.

[SW10] Abigail J. Sellen and Steve Whittaker. Beyond total capture: a constructive critique of lifelogging. *Communications of the ACM*, 53(5):70–77, 2010. ISSN: 0001-0782. DOI: 10.1145/1735223.1735243.

[SF94] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In *The Second Text REtrieval Conference*, volume 500–215, pages 243–252. National Institute of Standards and Technology (NIST), 1994.

[SBH97] W. M. Shaw, Robert Burgin, and Patrick Howell. Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing & Management*, 33(1):1–14, 1997. ISSN: 0306-4573. DOI: 10.1016/S0306-4573(96)00043-X.

[SBY17] Baoguang Shi, Xiang Bai, and Cong Yao. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2016.2646371.

[Sie57] Sidney Siegel. Nonparametric Statistics for the Behavioural Sciences. *The Journal of Nervous and Mental Disease*, 125(3):497, 1957. ISSN: 0022-3018.

[SHS+18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. DOI: 10.1126/science.aar6404.

[Skl93] Robert Sklar. *Film: An International History of the Medium*. Prentice Hall, 1993.

[Sme07] Alan F. Smeaton. Techniques used and open challenges to the analysis, indexing and retrieval of digital video. *Information Systems*, 32(4):545–559, 2007. ISSN: 0306-4379. DOI: 10.1016/j.is.2006.09.001.

[SWS+00] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000. ISSN: 1939-3539. DOI: 10.1109/34.895972.

[Smi07] R. Smith. An Overview of the Tesseract OCR Engine. In *International Conference on Document Analysis and Recognition*, volume 2, pages 629–633, 2007. DOI: 10.1109/ICDAR.2007.4376991.

[SW09] Cees G. M. Snoek and Marcel Worring. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval*, 4:215–322, 2009. ISSN: 1554-0669. DOI: 10.1561/1500000014.

[SWS05] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *International Conference on Multimedia*, pages 399–402. Association for Computing Machinery, 2005. ISBN: 978-1-59593-044-6. DOI: 10.1145/1101149.1101236.

[SSX+16] Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Xiang Ruan. Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In *British Machine Vision Conference*, volume 1, page 3, 2016. DOI: 10.5244/C.30.132.

[SMW+13] Yang Song, Hao Ma, Hongning Wang, and Kuansan Wang. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *International Conference on World Wide Web*, pages 1201–1212. Association for Computing Machinery, 2013. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488493.

[SL20] Tomáš Souček and Jakub Lokoč. TransNet V2: An effective deep network architecture for fast shot transition detection, 2020. DOI: 10.48550/arXiv.2008.04838.

[SS99] Murray R Spiegel and Larry J Stephens. *Schaum's Outline of Theory and Problems of Statistics*. Erlangga, 1999.

[SGH+22] Florian Spiess, Ralph Gasser, Silvan Heller, Mahnaz Parian-Scherb, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Multimodal Video Retrieval in Virtual Reality with vitrivr-VR. In *MultiMedia Modeling*, pages 499–504. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_45.

[SGH+21a] Florian Spiess, Ralph Gasser, Silvan Heller, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Competitive Interactive Video Retrieval in Virtual Reality with vitrivr-VR. In *MultiMedia Modeling*, pages 441–447. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_42.

[SGH+21b] Florian Spiess, Ralph Gasser, Silvan Heller, Luca Rossetto, Loris Sauter, Milan van Zanten, and Heiko Schuldt. Exploring Intuitive Lifelog Retrieval and Interaction Modes in Virtual Reality with vitrivr-VR. In *Workshop on Lifelog Search Challenge*, pages 17–22. Association for Computing Machinery, 2021. DOI: 10.1145/3463948.3469061.

[SS22] Florian Spiess and Heiko Schuldt. Multimodal Interactive Lifelog Retrieval with vitrivr-VR. In *Workshop on Lifelog Search Challenge*, pages 38–42. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: 10.1145/3512729.3533008.

[SLT+21] Newton Spolaôr, Huei Diana Lee, Weber Shoity Resende Takaki, Leandro Augusto Ensina, Antonio Rafael Sabino Parmezan, Jefferson Tales Oliva, Claudio Saddy Rodrigues Coy, and Feng Chung Wu. A video indexing and retrieval computational prototype based on transcribed speech. *Multimedia Tools and Applications*, 80(25):33971–34017, 2021. ISSN: 1573-7721. DOI: 10.1007/s11042-021-11401-1.

[Spo02] Jared Spool. Usability beyond common sense, 2002.

[Spr11] Peter Sprent. Sign Test. In *International Encyclopedia of Statistical Science*, pages 1316–1317. Springer, 2011. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_515.

[Spr14] Michael Springmann. *Building Blocks for Adaptable Image Search in Digital Libraries*. Thesis, University of Basel, 2014. DOI: 10.5451/unibas-006244034.

[SG14] Grant Strong and Minglun Gong. Self-Sorting Map: An Efficient Algorithm for Presenting Multimedia Data in Structured Layouts. *IEEE Transactions on Multimedia*, 16(4):1045–1058, 2014. ISSN: 1941-0077. DOI: 10.1109/TMM.2014.2306183.

[SE98] Alistair Sutcliffe and Mark Ennis. Towards a cognitive theory of information retrieval. *Interacting with Computers*, 10(3):321–351, 1998. ISSN: 1873-7951. DOI: 10.1016/S0953-5438(98)00013-7.

[Sut09] Alex J. Sutton. Publication bias. *The Handbook of Research Synthesis and Meta-Analysis*, 2:435–452, 2009.

[SC06] Simon Sweeney and Fabio Crestani. Effective search results summary size and device screen size: Is there a relationship? *Information Processing & Management*, 42(4):1056–1074, 2006. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2005.06.007.

[TLS+16] Giovanni Taverriti, Stefano Lombini, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Real-time Wearable Computer Vision System for Improved Museum Experience. In *International Conference on Multimedia*, pages 703–704. Association for Computing Machinery, 2016. ISBN: 978-1-4503-3603-1. DOI: 10.1145/2964284.2973813.

[Tay62] Robert S. Taylor. The process of asking questions. *American Documentation*, 13(4):391–396, 1962. ISSN: 1936-6108. DOI: 10.1002/asi.5090130405.

[TRB22] Alexander Theus, Luca Rossetto, and Abraham Bernstein. HyText – A Scene-Text Extraction Method for Video Retrieval. In *MultiMedia Modeling*, pages 182–193. Springer International Publishing, 2022. ISBN: 978-3-030-98355-0. DOI: 10.1007/978-3-030-98355-0_16.

[Tof84] Alvin Toffler. *The Third Wave*. Penguin Random House, 1984. ISBN: 978-0-553-24698-8.

[TFG+09] Ricardo da S. Torres, Alexandre X. Falcão, Marcos A. Gonçalves, João P. Papa, Baoping Zhang, Weiguo Fan, and Edward A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2008.04.010.

[TND+23] Ly-Duyen Tran, Manh-Duy Nguyen, Duc-Tien Dang-Nguyen, Silvan Heller, Florian Spiess, Jakub Lokoč, Ladislav Peška, Thao-Nhu Nguyen, Omar Shahbaz Khan, Aaron Duane, Björn Þór Jónsson, Luca Rossetto, An-Zi Yen, Ahmed Alateeq, Naushad Alam, Minh-Triet Tran, Graham Healy, Klaus Schoeffmann, and Cathal Gurrin. Comparing Interactive Retrieval Approaches at the Lifelog Search Challenge 2021. *IEEE Access*:1–1, 2023. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3248284.

[TNN+22] Ly-Duyen Tran, Manh-Duy Nguyen, Binh Nguyen, Hyowon Lee, Liting Zhou, and Cathal Gurrin. E-Myscéal: Embedding-based Interactive Lifelog Retrieval System for LSC'22. In *Workshop on Lifelog Search Challenge*, pages 32–37. Association for Computing Machinery, 2022. ISBN: 978-1-4503-9239-6. DOI: 10.1145/3512729.3533012.

[V3c] V3C Video 07119.

[van79] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. ISBN: 0-408-70929-4.

[VG20] Aneesh Vartakavi and Amanmeet Garg. PodSumm – Podcast Audio Summarization, 2020. DOI: 10.48550/arXiv.2009.10315.

[VML21] Patrik Veselý, František Mejzlík, and Jakub Lokoč. SOMHunter V2 at Video Browser Showdown 2021. In *MultiMedia Modeling*, pages 461–466. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_45.

[VC99] Christopher C. Vogt and Garrison W. Cottrell. Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1(3):151–173, 1999. ISSN: 1573-7659. DOI: 10.1023/A:1009980820262.

[Vog22] Marco Dieter Vogt. *Adaptive Management of Multimodel Data and Heterogeneous Workloads*. Thesis, University of Basel, 2022.

[VH05] Ellen M Voorhees and Donna K Harman. *TREC: Experiment and Evaluation in Information Retrieval*, volume 63. Citeseer, 2005.

[Wan03] Avery Wang. An industrial strength audio search algorithm. In *International Conference on Music Information Retrieval*, 2003.

[Wan06] Avery Wang. The Shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006. ISSN: 0001-0782. DOI: 10.1145/1145287.1145312.

[WCM+19] Zhipeng Wei, Jingjing Chen, Zhaoyan Ming, Chong-Wah Ngo, Tat-Seng Chua, and Fengfeng Zhou. DietLens-Eout: Large Scale Restaurant Food Photo Recognition. In *International Conference on Multimedia Retrieval*, pages 399–403. Association for Computing Machinery, 2019. ISBN: 978-1-4503-6765-3. DOI: 10.1145/3323873.3326923.

[Wel22] Deutsche Welle. XRECO, 2022. URL: https://cordis.europa.eu/project/id/101070250.

[Wel11] Brian Welsh. Black Mirror: The Entire History of You. Season 1 Episode 3. 2011.

[WSd+06] Marcel Worring, Cees Snoek, Ork de Rooij, Giang Nguyen, Richard van Balen, and Dennis Koelma. Mediamill: advanced browsing in news video archives. In *International Conference on Image and Video Retrieval*, pages 533–536. Springer-Verlag, 2006. ISBN: 978-3-540-36018-6. DOI: 10.1007/11788034_62.

[WN20] Jiaxin Wu and Chong-Wah Ngo. Interpretable Embedding for Ad-Hoc Video Search. In *International Conference on Multimedia*, pages 3357–3366. Association for Computing Machinery, 2020. ISBN: 978-1-4503-7988-5. DOI: doi.org/10.1145/3394171.3413916.

[WNM+21] Jiaxin Wu, Phuong Anh Nguyen, Zhixin Ma, and Chong-Wah Ngo. SQL-Like Interpretable Interactive Video Search. In *MultiMedia Modeling*, pages 391–397. Springer International Publishing, 2021. ISBN: 978-3-030-67835-7. DOI: 10.1007/978-3-030-67835-7_34.

[WC02] Shengli Wu and Fabio Crestani. Data fusion with estimated weights. In *Internation Conference on Information and Knowledge Management*, pages 648–651. Association for Computing Machinery, 2002. ISBN: 978-1-58113-492-6. DOI: 10.1145/584792.584908.

[WCB06] Shengli Wu, Fabio Crestani, and Yaxin Bi. Evaluating Score Normalization Methods in Data Fusion. In *Information Retrieval Technology*, pages 642–648. Springer, 2006. ISBN: 978-3-540-46237-8. DOI: 10.1007/11880592_57.

[XGD+17] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017. DOI: 10.1109/CVPR.2017.634.

[YLS+16] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen-Change Loy. Sketch Me That Shoe. In *Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.

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

[ZM12] Cha Zhang and Yunqian Ma. *Ensemble Machine Learning: Methods and Applications*. Springer, 2012. ISBN: 978-1-4419-9325-0. DOI: 10.1007/978-1-4419-9326-7.

[ZZX+19] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2018.2876865.

[ZSY+17] Xiaoqiang Zhu, Lei Song, Lihua You, Mengyao Zhu, Xiangyang Wang, and Xiaogang Jin. Brush2Model: Convolution surface-based brushes for 3D modelling in head-mounted display-based virtual environments. *Computer Animation and Virtual Worlds*, 28(3-4):e1764, 2017. ISSN: 1546-427X. DOI: 10.1002/cav.1764.