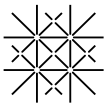


University of Basel

vitrivr, why? Understanding Results in Multimedia Retrieval

Master Project Report

Faculty of Sciences, University of Basel
Department of Mathematics and Computer Science
Databases and Information Systems Group

Examiner: Prof. Dr. Heiko Schuldt
Supervisor: Silvan Heller, MSc.

Cristina Illi
cristina.illi@unibas.ch
15-053-184

January 29^{th}, 2021

Acknowledgments

First, I would like to thank Silvan Heller, who provided me with valuable input and guidance throughout the course of this project. Additionally, I would like to thank Prof. Dr. Heiko Schuldt for giving me the opportunity to do another project in the Database and Information Systems Group.
Most importantly, all my gratitude goes out to the many lovely people who spent their time to take part in the evaluation. It cannot be taken for granted that they gave their time for the evaluation conducted as part of my project.

Abstract

As the amount of digitally available information is consistently growing, the demand of effective retrieval systems becomes of greater importance in order to make efficient use of the data. To reach a maximum synergy between user and system, a higher transparency of the inner workings and results is vital.
This project focused on extending the functionality of vitrivr, an open-source content-based multimedia retrieval stack, in order to create more transparency and therefore bringing the user and the system closer together. A prioritisation feature for the search tags, statistical as well as relational insights on the result set and feature information about individual elements were added to vitrivr. An evaluation was carried out to examine the impact of the added changes on user performance and understanding. The participants using the new version of vitrivrappeared to be more successful. Indicators that the additions made it harder for the participants to use the systems were not found. The added functionality to vitrivr presents a good step towards higher query quality and system transparency through explainability.

Table of Contents

1 Introduction

As the amount of digitally available information is consistently growing, the demand of effective retrieval systems becomes of greater importance in order to make efficient use of the data. To formulate a query and find matching results, it is the users' responsibility to make use of concepts known to the system. In order to close this semantic gap and thus reach a maximum synergy between user and system, a higher transparency of the inner workings and results is vital.
Higher transparency brings the user and the system closer together. On the one hand, the user can better exploit the systems' functionality. On the other, it supports the developers in better evaluating the system and tailor it more precisely to the user.
When using multimedia retrieval systems, instead of relying on relevance feedback from the user, the information conveyed by well explained results can inspire to think of different ways of expressing the query. This is especially useful if the user is stuck and does not receive satisfying results. Furthermore, a better understanding of the system decreases the number of necessary iterations of query formulation and browsing through the result set. Ultimately, a user's trust in the presented output and rankings depends on avoiding black boxes.

1.1 vitrivr

vitrivr[13] is an open-source content-based multimedia retrieval stack that supports many different query formulation modalities. Those include tag-based queries with a type-ahead functionality that suggests the tags known to the system, textual queries which enable to search for text on screen and audio transcripts as well as generated scene captions. Finally, the users can also query by sketch by providing an example, a drawing or using semantic sketches. All query types can be freely combined in various ways such as in the manner of multi-stage queries or temporal ordering [5]. vitrivr is made up of three main components: the user interface vitrivr-ng [1], the retrieval engine Cineast [12] and the database Cottontail DB [2]. An illustration is given in Figure 1.1.

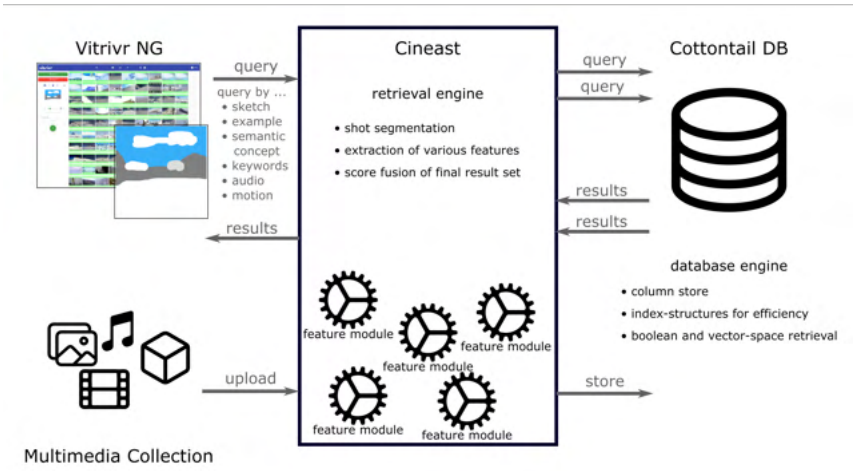


Figure 1.1: The overall architecture of vitrivr consisting of vitrivr-ng, Cineast and Cottontail DB.

vitrivr-ng The user interface (UI) is called vitrivr-ng and is written in Angular. Here, users can express and submit their queries and browse the corresponding results. Using a WebSocket API, the user interface interacts with Cineast. When a query is made of multiple subqueries, vitrivr-ng receives partial results for each component and merges them into a complete result set. A customisable scoring function can be used to influence the weight of each subquery and further refine the order of items displayed. Segments that appear in multiple subqueries experience a boost in their score. The result set can be browsed through and individual elements can be viewed in an embedded video player.

Cineast Cineast is vitrivr's retrieval engine written in Java. It is the connector between Cottontail DB and vitrivr-ng as it translates the users' queries to database queries and returns the results back to the user. Cineast can handle multi-feature content-based multimedia. This is done by performing shot segmentation, feature extraction and score fusion of the final result set.

Cottontail DB The custom database used in vitrivr is Cottontail DB, a column store with index structures. It uses boolean as well as vector-space retrieval such as k-nearest-neighbours search to efficiently process queries coming from Cineast.

1.2 Goals

The aim of this project was to improve the vitrivr system by adding functionality which supports the users in increasing their understanding of its inner workings and why the presented result has been returned. A focus was laid on improving tag-based queries by introducing the functionality of assigning a preference to a tag. Additionally, information about the result set was added to increase the transparency why the shown set was returned to the user. Finally, the possibility to display feature information for individual video segments was added to the user interface.
Some of the contributions will be used for the Video Browser Showdown 2021 [6].

1.3 Evaluation

A comparison between the current version of vitrivr and the one containing the new features was conducted. This evaluation was modelled after the Video Browser Showdown (VBS) [19] since

vitrivr is a long-running participant in interactive evaluation campaigns such as LSC (Lifelog Search Challenge) [3, 14] and VBS [6, 15, 18] at which vitrivr will participate also with a virtual reality (VR) interface [20]. 17 people participated in the evaluation which generated the needed data to create insight in the usefulness of the added functionality. This was the chosen mechanism to evaluate which contributions were meaningful and help the user to better understand the results and consequently improve their queries.

1.4 Outline

The remainder of this report is structured as follows: Chapter 2 provides an overview of other retrieval systems. A detailed description of the contributions made in the course of this project can be found in Chapter 3. The conducted evaluation, the collected results and gained insights are presented in Chapter 4. A conclusion and outlook on possible future work can be found in Chapter 5.

2
Related Work

This chapter mentions other retrieval systems, what their approach of query formulation looks like and how increasing the transparency of the results is achieved.

2.1 Exquisitor

To fully exploit the user's feedback when browsing large multimedia collections, Jónsson et al. [7] proposed the Exquisitor system that aims at learning the user's preference through simple mouse-based feedback. Starting from a random selection of video scenes, the users are asked to assign each segment with relevance feedback (positive or negative) by simply dragging the segment to either the left (positive) or right (negative) side of the screen. A third feedback option, the 'next'-button, marks all shown video segments as seen and presents a new set of scenes. A generated classification model based on the assigned feedback is used to suggest new video segments for the next round of user interaction. In this case, the user explicitly tells the system whether the retrieved items are what the user was looking for or not. The user is limited to accept or reject an item but cannot justify the decision.

2.2 SOM-Hunter

Kratochvíl et al. [9] proposed SOM-Hunter, a retrieval engine with a focus on known-item search workflow. The search is initiated with a query from the user that leads to a candidate set of results. This set is explored in an iterative fashion by providing positive, negative or no feedback to the system. Not assigning any reaction to an item is implicitly rated as negative. Browsing of a larger result set is made more time-efficient using self-organising maps [8] that are based on the scores created from the user's feedback. As vitrivr has also experimented with SOMs [4], where the added functionality makes use of relevance feedback to explore multimedia collections.

2.3 VIRET

VIRET offers three flavours of query formulation (by keyword, colour sketching and query-by-example) which can further be combined and ordered temporally. The system was extended for the 2020 VBS (Video Browser Showdown) with features a focus on reducing the number of query (re)formulations Lokoč et al. [10]. Informative visualizations that help the users (with a high focus

on novices) with query specification were added to the user interface. Upon entering a keyword not only suggestions of matching keywords appear but also a selection of the top ranked frames matching the provided keyword. This is especially useful for novices to get acquainted with the automatic annotations and interactively improve their query with the provided inspiration.

2.4 Video Browser Showdown

The first editions of the annual Video Browser Showdown was held in 2012 [19]. In the form of a competition, participating teams get the chance to evaluate their video retrieval systems. To ensure fairness, all teams have to solve the same tasks at the same time, on the same dataset. Over the course of hours, the participants perform different kinds of tasks such as Visual Known-Item-Search (KIS V), Textual Known-Item-Search (KIS T) or Ad-Hoc Video Search (AVS) tasks. The submissions for KIS V and KIS T are automatically assessed by the competition server whereas the submissions for AVS tasks are manually evaluated by live judges. The winner of the competition is determined by the number of collected points, given on the basis of correctness and time to submission. Usually, the VBS consists of an expert round where each participating team uses their own retrieval system and a novice round with participants who are unfamiliar with the competing candidates.

3
Contributions

To achieve the goal of furthering the user's understanding of the system and the delivered results, several new features were introduced to vitrivr. Those additions include a prioritisation feature for the search tags, additional information on the result set and its individual elements that are described in this chapter.

3.1 Prioritise Tags

So far, vitrivr only considered multiple tags in an *OR* fashion but it was not possible to associate them with a specific priority. To make tag-based queries more powerful, the following prioritisation of tags was introduced:

MUST: Segments in the final result set *must* be annotated with this tag.

COULD: Segments annotated with these tags are ranked higher.

NOT: Segments associated with these tags are excluded from the final result set.

vitrivr-ng was extended so that the prioritisation of a tag can be chosen using one of three icons: a green heart for *must*, an orange thumb for *could* and a red forbidden icon for *not*. This is presented in Figure 3.1. The default priority for a tag is set to *could* and if only tags with this priority are chosen, the new version of vitrivr behaves the same way as the previous one.

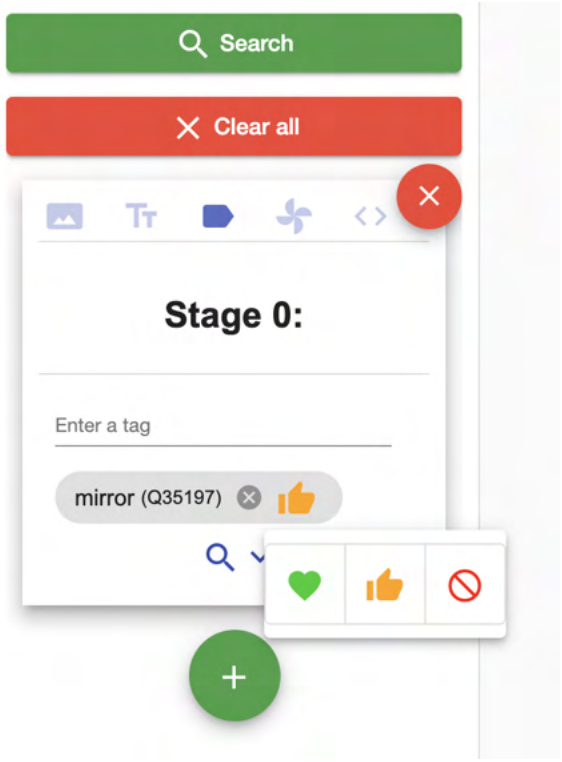


Figure 3.1: A tag can be associated with a preference. The default value is *could* (yellow thumb) but can be changed to *must* (green heart) or *not* (red forbidden icon)

Once a query is submitted and a result set is returned from Cottontail DB to Cineast, the segments are then triaged according to the tags of them is associated with. The goal is to eliminate all segments that contain a *not* tag and keep those that are annotated with each of the *must* tags. If a segment contains a *must* and a *not* tag, the latter is weighted stronger and the segment will not be included in the final result set. An example with one or more *must* and *not* and two *could* tags is given in Figure 3.2. Starting with the segments that are associated with all of the *must* tags, those segments that also contain one or more *not* tags are eliminated. Each *could* tag that the segments are also labelled with improves the final score of the segments.

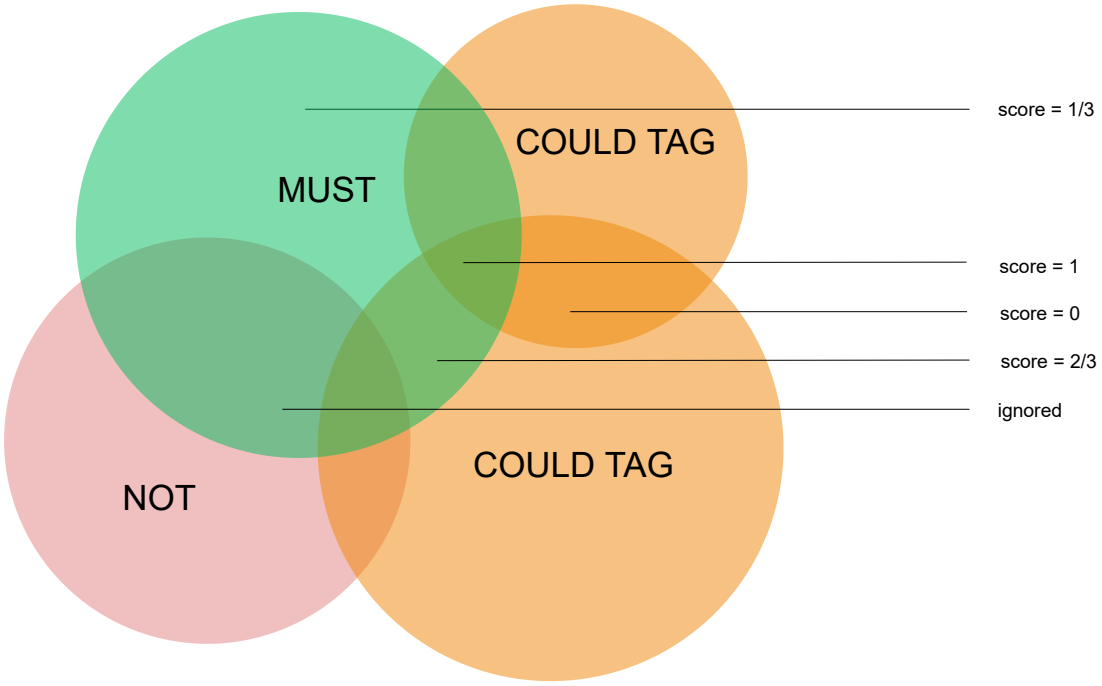


Figure 3.2: Scoring for an example with two *could* tags (matching segments shown in orange). Green represents the segments that are associated with all *must* tags, red shows the union of the segments containing one or more *not* tags. The score for each segment depends on whether it also contains *could* or *not* tags.

The final result set will be made of the segments that contain the *must* tags, minus the the ones annotated with a *not* tag. The score for each element in the final set is improved if it is also contained in one or more *could* sets. An algorithm to score each segment in the final result set was implemented and is presented in Algorithm 3.1.

Algorithm 3.1: Scoring of each element in final result set

```
1   mustSegements = intersection(m_1, m_2, ..., m_n)
2   notSegments = union(n_1, n_2, ..., n_k)
3   couldSet = {c_1, c_2, ..., c_j }
4   Map<id, score> scores
5   for(m_i:mustSegements)
6       if(notSegments.contains(m_i))
7           continue
8       score = 1/(couldSet.size() + 1)
9       for(c_l: couldSet)
10          if(c_l.contains(m_i)
11          score += 1/(couldSet.size() + 1)
12      scores.put(m_i, score)
```

3.2 Information about Result Set

The query refinement sidebar on the right side was extended with an addition tab (see Figure 3.3) to display simple statistical information about the result set. The goal of the information contained in this tab is to assist the user in obtaining a deeper understanding of the result set and to directly refine the query. Furthermore, this should also increase the transparency of how the results came about. This is also displayed for queries that are not based on tags.

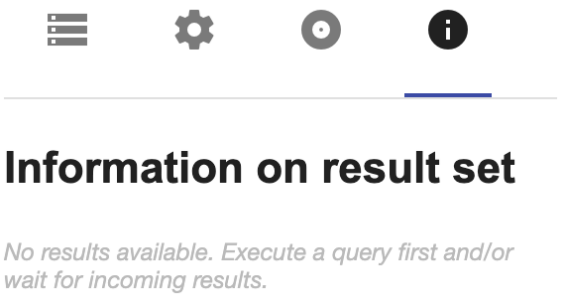


Figure 3.3: An additional tab was added to the query refinement bar located on the right side of vitrivr-ng. Information about the result set as a whole can be obtained from here.

Number of Elements in Result Set The size of the result set is mentioned here. It can give context on how broad or narrow the query was formulated. This number is mostly needed to put the remaining statistical information in this tab in perspective.

Related Tags and Their Occurrences Looking at the result set as a whole, the related tag of all segments are counted and displayed with the number of occurrence. Each tag in this list can be added directly to the query by setting the preference to either *must*, *could* or *not*. The number of shown tags is set to ten as a default but can be changed via an input field to any number if needed, so also the rarely occurring tags can be inspected. The added feature can be seen in Section 3.2. This addition was created so that the users could receive more background information on the result. It can also serve as a source of inspiration in order to find other tags or exclude them for a next query.

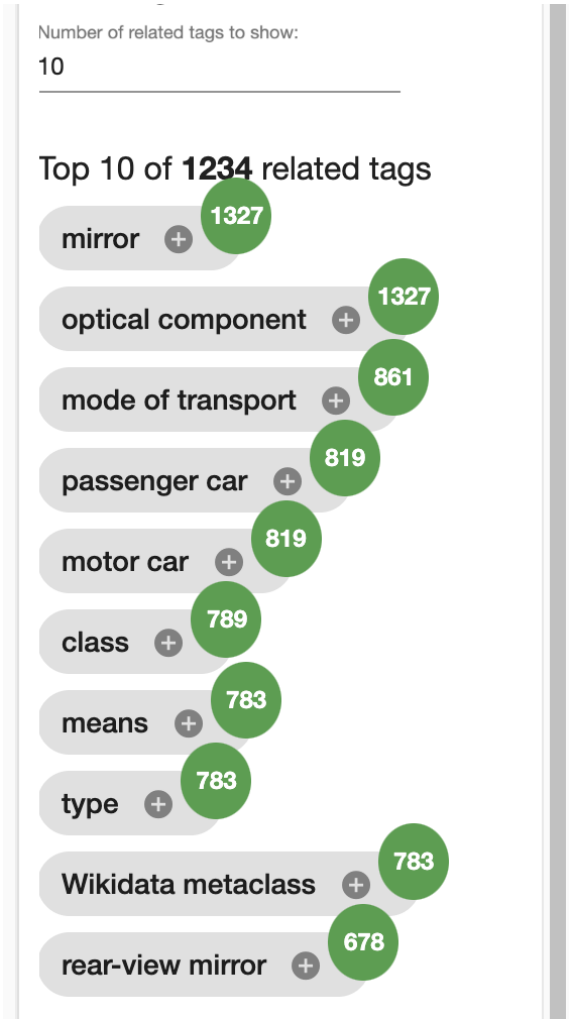


Figure 3.4: The most frequent tags in relation to the segments in the result set are displayed. Also, each number of occurrence is mentioned. The quantity of related tags can be adjusted manually via an input field.

Terms in Captions This part follows a similar approach as with the related tags. For each segment the captions are cleaned of stop words such as 'we', 'or', 'if', etc. The sum of term occurrences is added over the whole result set. For an easier visual presentation a word cloud was implemented. The higher the count of a term, the larger it appears in the word cloud. In order to maintain a manageable overview, only the 25 most frequently occurring terms are included. The total number of terms in the captions (stop words excluded) is also shown to create more context. A list with the exact values can be uncovered via a toggle button located above the word cloud. This could be useful to get a more detailed feeling about the relevance of some terms or to find further inspiration for a next query iteration.

Figure 3.5: The 25 most frequent terms occurring in the captions of the individual segments are displayed as a word cloud. A list with the exact values can be shown via the toggle button above the word cloud.

Score Distribution To further discover how *good* a query was, the score distribution of the result set can be consulted. Most often it is useful to see if the query scored more videos in the lower range of scores which could indicate the query to be very selective. The opposite verdict would hold for too many scores in the upper range because the query matched for a lot of targets.

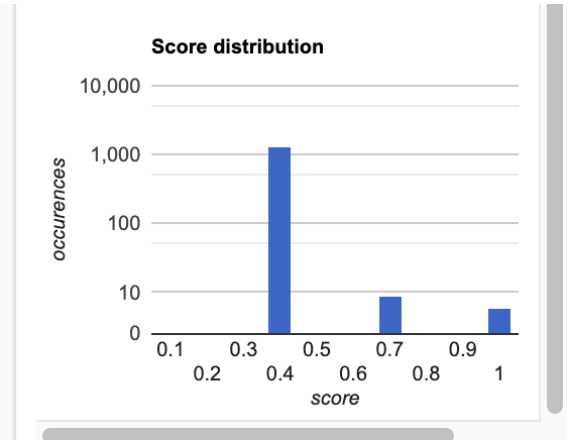


Figure 3.6: The number of scores per tenth is summed up and displayed in a bar plot. This could shed light on the quality of the query.

3.3 Feature Information for Individual Elements

So the users can further examine an individual segment, the 'more details' section was extended. When hovering over an individual segment, the 'eye' symbol (see Figure 3.7) appears and by clicking it the user can navigate to the page that contains further information.

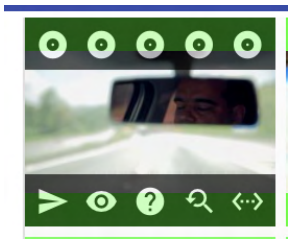


Figure 3.7: More information about an individual segment can be found via the 'eye' symbol (second from left) upon hovering over an element.

When a segment is associated with tags, is described with a caption, ASR (automatic speech recognition) or OCR (optical character recognition) data is available, these details are shown in the respective tab as presented in Figure 3.8.

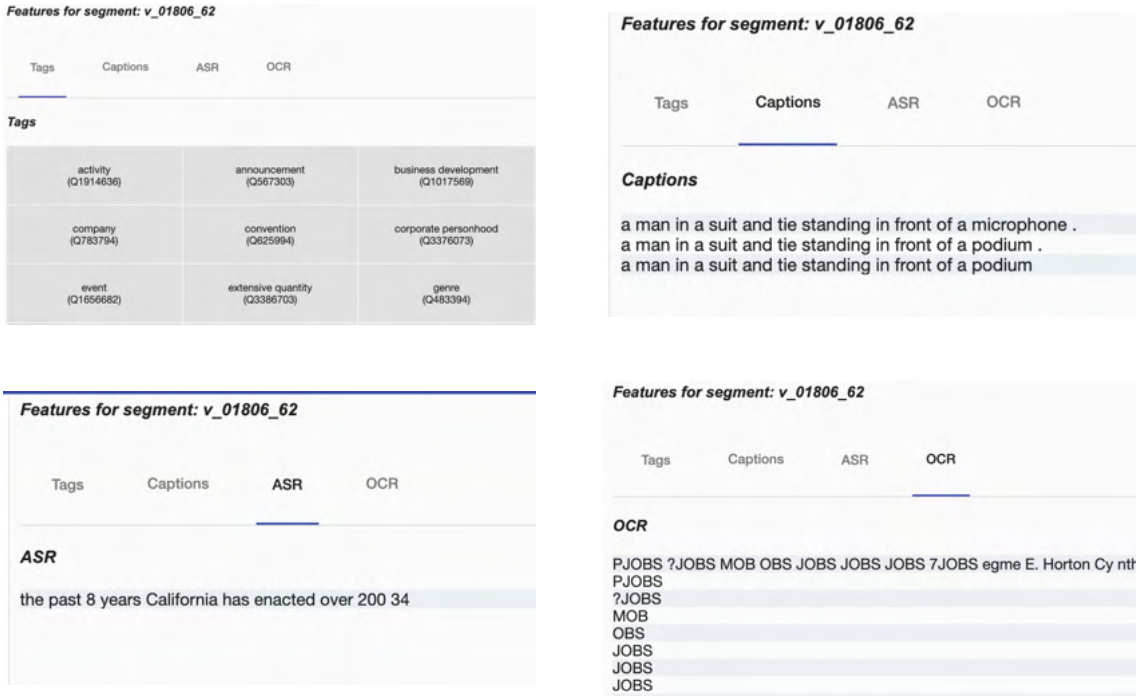


Figure 3.8: The feature information such as related tags, captions, ASR and OCR data for an individual segment can be accessed by navigating through the tab menu.

4 Evaluation

The aim of this evaluation is to examine what impact the changes had on the users' performances and their understanding of the system. To be able to compare user performances, participants were required to solve the same tasks with either an old vitrivr setup, or one including the new features, similar to the evaluation in the Video Browser Showdown [19]. This chapter presents the methods and results of the conducted evaluation.

4.1 Setup

Due to the pandemic the participants and the organisers met virtually via zoom[1]. Eight nodes (located in the basement of the Department of Computer Science (DMI)) were prepared. Seven of those nodes were only accessible from within the network of the university, so the participants had to log in via a VPN. One node was accessible from outside the uni-network so that two people who are currently not members of the University of Basel could also participate. Each node was equipped with Cottontail DB, Cineast and vitrivr-ng. Ten participants were provided with the vitrivr version that contained the added improvements mentioned in Chapter 3, seven used the current version of vitrivr. V3C1 [16], the dataset that is used for the VBS competitions, was also used for this evaluation.

One day before the evaluation, a cheat sheet corresponding to their assigned vitrivr version (see A.13 and A.14) was distributed to the participants. The idea was to give them a chance to get an overview of what vitrivr is capable of. They were allowed to use the cheat sheet during the evaluation.

17 people currently or formerly associated with the DMI participated in the evaluation. Ten were considered experts since they had prior experience, the remaining seven did not know anything about vitrivr. The meeting began with a ten minute introduction to vitrivr, primarily for the novices. Due to resource and scheduling constraints the evaluation was conducted on two separate days. The tasks were the same for both days and for all participants.

[1] https://zoom.us/meetings

4.2 DRES

For this evaluation DRES (Distributed Retrieval Evaluation Server) [17] was used as the central orchestration tool to conduct the evaluation. The participants were provided with login credentials to their respective DRES accounts. Once logged in, they can see the standings for the entire competition as well as for the current task type (see Figure 4.1). Most importantly, the participants see the description of the current task.
Using vitrivr, the correct segment had to be found and submitted to DRES in order to be either judged (in the case of an AVS task) or evaluated to be *correct* or *incorrect* (for KIS V and KIS T tasks).
Not only the task definition and the conduction of the evaluation is done with DRES, but also some of the evaluation statistics are automatically collected, aggregated & processed in DRES. Those statistics can then be exported for further data analysis.

4.3 Tasks

The participants are asked to solve different kinds of tasks using their respective vitrivr systems. For this evaluation, three types of tasks are used. The participants need to find either a specific video segment (Textual Known-Item-Search (KIS T) and Visual Known-Item-Search (KIS V)) or multiple scenes matching a given description (Ad-Hoc Video Search (AVS). Each task is described in textual form (KIS T and AVS) or the specific video scene is shown (KIS V).
As also experts who were well acquainted with the existing tasks participated in the evaluation, new tasks had to be created. A full documentation of the individual tasks can be found in Appendix A.1.

4.3.1 Visual Known-Item-Search (KIS V)

For this kind of task, the users see a 20 second segment of a video as depicted in Figure 4.1. It is played in an endless loop, so the participants can see the example video multiple times. The users were are asked to find the matching scene and submit it to DRES. The task was limited to five minutes.

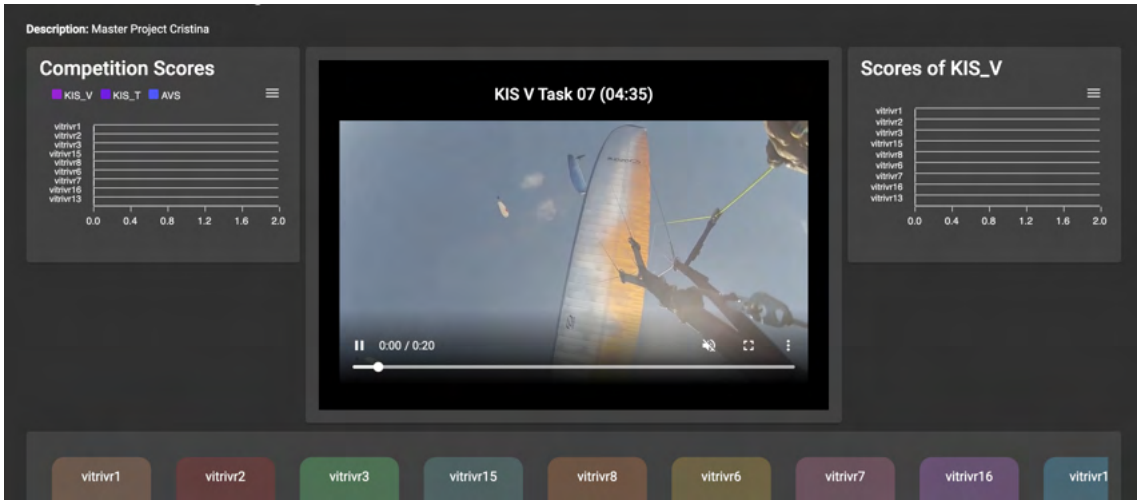


Figure 4.1: How the participants saw a KIS V (visual known-item search) task. The segment which the users had to find is played over and over again. The score and ranking of the overall competition is shown on the left, the one for the current task type on the right.

4.3.2 Textual Known-Item-Search (KIS T)

For the textual known-item-search (KIS T) tasks the participants were given three hints that uniquely described the correct scene. After 60 seconds a new hint appeared on the screen. The time limit for this task type was set to seven minutes. Figure 4.2 shows the first hint of the first textual task as seen by the participants in the competition. The three hints given to the participants were the following:

- A man starts a blue hover boat and drives it over a swamp.
- A man starts a blue hover boat and drives it over a swamp. He has a beard, wears sun glasses and red hearing protection.
- A man starts a blue hover boat and drives it over a swamp. He has a beard, wears sun glasses and read hearing protection. The sequence sometimes shows the swamp, sometimes the man who is talking about why he is there. His name is Humberto Jimenez.

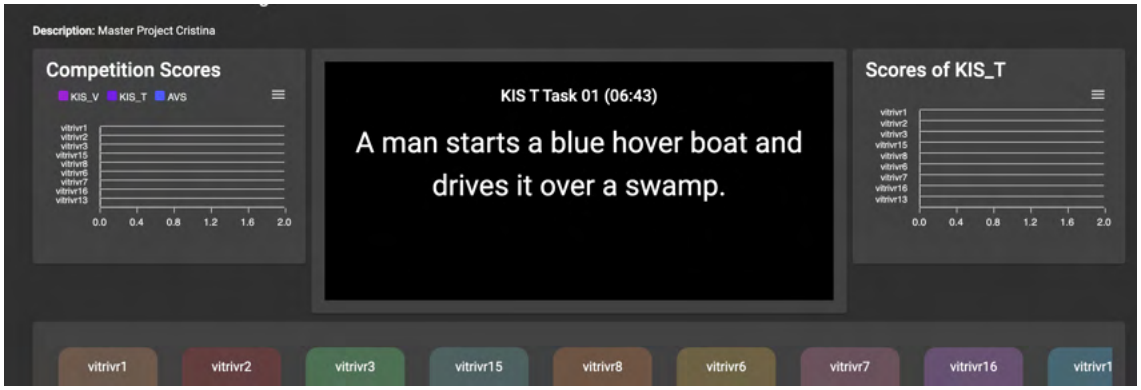


Figure 4.2: How the participants saw a textual known-item search (KIS T) task. After 60 seconds a new hint appeared on the screen in the centre.

4.3.3 Ad-Hoc Video Search (AVS)

The goal for the ad-hoc video search tasks (AVS) tasks is to find as many segments that match the description. This is the only task type where multiple submitted segments could be rated as 'correct'. A judge determined whether the submitted segments matched the desired description, an example i shown in Figure 4.3. Each AVS task was terminated after five minutes.

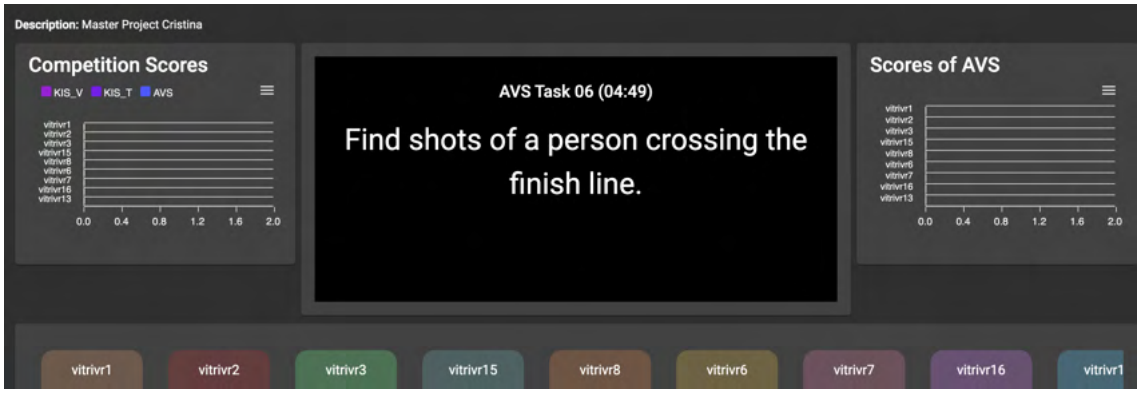


Figure 4.3: How the participants saw an ad-hoc video search (AVS) task. Based on the description, the users had to submit as many suitable segments as possible.

4.4 Results & Analysis

The collected logs from both evaluation rounds allowed to obtain insights into the users' query formulations, query results, interactions and submissions. Furthermore, the qualitative feedback provided by the participants is presented.

4.4.1 Overall Results

In the course of this evaluation 25 tasks were set in total. They were equally distributed among the three task types: eight AVS, eight KIS T nine KIS V tasks. The 17 participants made 814 submissions in total for all tasks combined.

The results presented in this section stem from both days (Monday and Thursday) on which we conducted the evaluation. Problems such as unreliable internet connections and inconsistent logging arose for some participants. This was taken into account for the entire analysis of the collected data by excluding their results for the respective tasks.

DRES awards points based on time to correct submission in relation to the other participants within the same run. Therefore, it is not possible to compare the points between the runs conducted on Monday and Thursday. Figure 4.4 shows the final ranking for both days separately. The upper ranking shows that two out of three podium spots were reached with the new version of vitrivr, notably by two novices. On the second run, the podium was occupied exclusively by users with the new version. On Monday, some participants did not score any points for the KIS V tasks. On Thursday, this was the case for the KIS T tasks. All participants on both days were able to collect points for the AVS tasks.

			SCORE				
	rank	user	overall	KIS V	KIS T	AVS	version
Monday	1	vitrivr5	282	**100**	**100**	82	new
	2	vitrivr10	229	72	83	74	current
	3	vitrivr14	214	78	51	85	new
	4	vitrivr11	194	0	100	94	current
	5	vitrivr4	190	0	95	95	current
	6	vitrivr12	163	54	62	47	new
	7	vitrivr17	156	0	56	**100**	new
	8	vitrivr9	141	50	45	46	new
Thursday	1	vitrivr13	237	**100**	**100**	37	new
	2	vitrivr16	220	44	83	93	new
	3	vitrivr6	171	31	40	**100**	new
	4	vitrivr3	166	41	64	61	current
	5	vitrivr2	149	69	0	80	current
	6	vitrivr7	109	24	0	85	current
	7	vitrivr8	99	14	0	85	current
	8	vitrivr1	93	32	0	61	new
	9	vitrivr15	87	27	0	60	new

total number of novices:	7	number of new version:	10
total number of experts:	10	number of current version:	7

Figure 4.4: The final ranking for the two runs conducted during the evaluation. For each user the final score and the score for the individual query types are shown. Additionally, it is indicated which version (new in blue or current in red) was used and whether the participant had prior knowledge about the system (expert in yellow) or not (novice in green).

Normally in the VBS evaluations, the scores are calculated for a team of multiple participants collectively. This was not the case in this evaluations as each participant performed as a one-person-team. Consequently, it was possible to study the various combinations of two participants. As mentioned in Section 4.2, DRES automatically delivers statistical analyses. All submissions are collected and re-scored for all possible combinations of two-person-teams. Figure 4.5 and Figure 4.6 show the results for the runs on Monday and Thursday respectively. The score for the individual participant are located on the diagonal. In Figure 4.5 and Figure 4.6 there is no clear hint on the perfect combination of expert and novice or new and old version.

Monday	new_expert_vitrivr12	new_novice_vitrivr14	new_expert_vitrivr17	new_novice_vitrivr5	new_expert_vitrivr9	old_expert_vitrivr10	old_expert_vitrivr11	old_expert_vitrivr4
new_expert_vitrivr12	163.00							
new_novice_vitrivr14	252.14	214.00						
new_expert_vitrivr17	259.73	397.54	156.00					
new_novice_vitrivr5	376.87	**490.51**	345.76	282.00				
new_expert_vitrivr9	128.62	175.95	182.70	248.56	141.00			
old_expert_vitrivr10	338.91	460.47	348.08	456.70	178.55	229.00		
old_expert_vitrivr11	312.04	**513.07**	454.80	370.17	311.55	331.37	194.00	
old_expert_vitrivr4	320.95	444.77	400.33	359.73	229.96	322.02	459.53	190.00

Figure 4.5: All possible pairings of participants for the first day of the evaluation and their respective score.

Thursday	new_novice_vitrivr1	new_expert_vitrivr13	new_novice_vitrivr15	new_expert_vitrivr6	new_novice_vitrivr16	old_expert_vitrivr2	old_novice_vitrivr3	old_novice_vitrivr7	old_expert_vitrivr8
new_novice_vitrivr1	93.00								
new_expert_vitrivr13	390.04	237.00							
new_novice_vitrivr15	230.63	178.80	87.00						
new_expert_vitrivr6	425.10	434.14	276.22	171.00					
new_novice_vitrivr16	349.75	393.57	223.50	459.63	220.00				
old_expert_vitrivr2	388.11	**583.64**	231.78	**502.68**	389.54	149.00			
old_novice_vitrivr3	290.41	403.57	179.75	271.62	403.33	222.77	166.00		
old_novice_vitrivr7	346.66	431.01	233.92	424.10	443.41	414.24	324.73	109.00	
old_expert_vitrivr8	354.69	414.36	236.14	429.36	368.27	448.44	261.63	277.95	99.00

Figure 4.6: All possible pairings of participants for the second day of the evaluation and their respective score.

For each task type with respect to the version (current or new) the average scores are shown in Figure 4.7. Again, the runs from Monday (columns two and three) and Thursday (columns four and five) cannot be compared as the scores are relative between the participants in the run. It can be seen that on Monday the participants using the current version achieved slightly better scores than the ones with the new version while on Thursday the new version clearly outperformed the current one. Overall, the variance seems to be high and the scores do not paint a very clear picture of the success for each version.

	new version average score per user	current version average score per user	new version average score per user	current version average score per user
	MONDAY		THURSDAY	
KIS V	56.4	24.0	46.8	29.6
KIS T	62.8	92.7	44.6	12.8
AVS	72.0	87.7	70.2	62.2
	191.2	204.4	161.6	104.6

Figure 4.7: The total score for both versions is broken up into the three query types.

4.4.2 Interaction and Query Formulations

5'508 files containing the 9'830 user interactions recorded in this evaluation were analysed. Unfortunately, for 447 of those interactions it was not possible to identify what the user's interaction was, probably due to bugs in the logging process.

According to the collected logs, six out of ten participants equipped with the new version used statistics about result set (ResultSetStatistics) that were included in the side bar on the right. Based on the interaction logs, only one participant (vitrivr9) used the feature data for and individual element. Therefore is can be concluded that the data is either not relevant or needs to be more accessible for a better workflow.

Table 4.2 shows the six out of ten new-version-users that interacted with the result set statistics. Although vitrivr17 (expert) had a lot more interactions than vitrivr5 (novice), the novice focused more on using the new additions than the expert who was probably more used to the known workflow without the additions. vitrivr5 was affected by logging problems and therefore the numbers

might be incomplete.

	total number of interactions	**usage of 'ResultSetStatisctics'**	**usage of 'ResultSetStatisctics' in % of interactions**
vitrivr5	276	17	6.2 %
vitrivr9	387	5	6.5 %
vitrivr13	387	1	0.2 %
vitrivr14	656	13	1.9 %
vitrivr15	790	4	0.5 %
vitrivr17	1316	7	0.5 %

Table 4.1: The number of times the results set information was used in relation to the total number of interactions.

The result set statistics in the side bar on the right contained the related tags described in Section 3.2. The interaction logging recorded that this functionality was only actively used by two users. Figure 4.8 gives information if a tag was added directly from information about the result set. It is possible that the users looked at the analysis in the side bar but added the tags manually to the query. This was not logged and not examined further. It seems that the related tags were mostly used during AVS tasks. The difference between the number for KIS V and KIS T tasks is negligible. Unfortunately, the new additions were not used as often as hoped. The participants that used the new functionality reported it to be very useful when tackling AVS tasks to insure the diversity of the results by getting inspiration for similar tags or even synonyms.

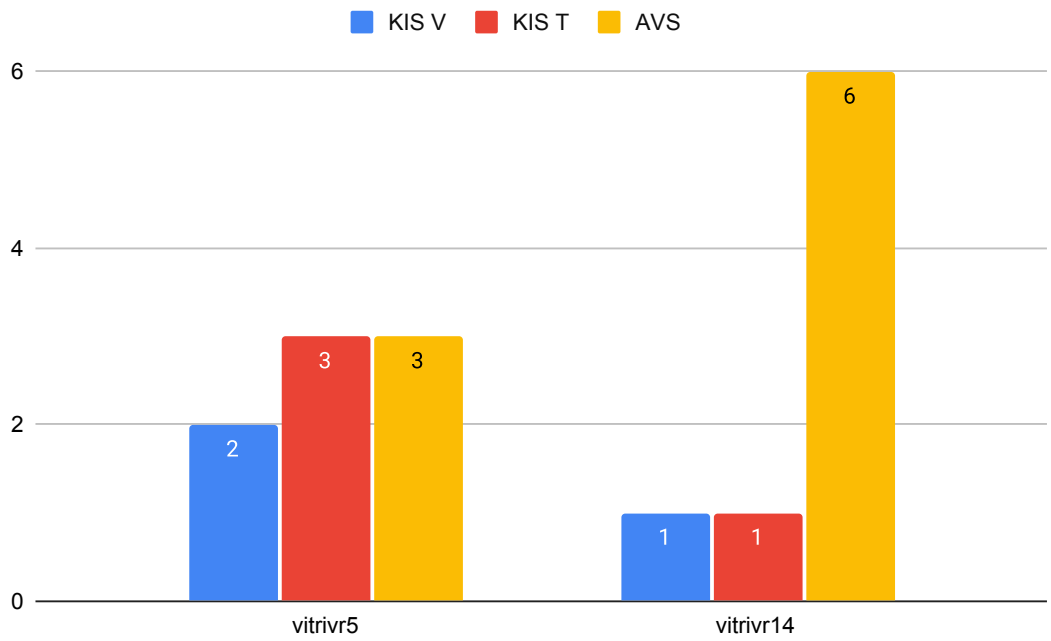


Figure 4.8: The number of times two out of ten participants added a tag directly from the result set information to their next query. The KIS V tasks are indicated in blue, the KIS T in red and the AVS tasks in yellow.

Figure 4.9 shows how many times each query type was used on average per version. 'Image' queries are more-like-this queries, 'Sketch' are the queries where the user draws something and the 'Text' queries are made up of descriptions, OCR, ASR and tag queries. For both versions, only the number of times each category was used varies but the trend is similar. The 'Text'-queries were used the most and the difference between the 'Text' and 'Sketch' queries is notably larger than it is for 'Sketch'- and 'Image'-queries for both versions.
A similar comparison is presented in Figure 4.10. On average, the 'Text' type was used way more often than the others, especially by the novices. Again, the differences for the other query types between the two levels are rather small.

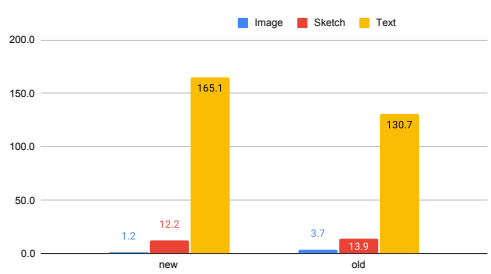


Figure 4.9: Comparing the usages of the different query types 'Images', 'Sketch' and 'Text' between the two vitrivr versions.

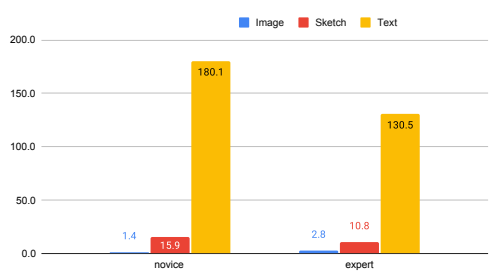


Figure 4.10: Comparing the usages of the different query types 'Images', 'Sketch' and 'Text' between novices and experts

The results of the deeper analysis of the 'Text' queries for each version is presented in Figure 4.11. The 'Text' queries summarise the OCR, ASR, tags and description queries. In both versions the

tag queries were used the most whereas the combination of OCR and ASR was the least popular query method. That the tags were used more often with the new version could be connected to the new feature where preferences can be set for each tag individually. A similar comparison is shown in Figure 4.12 but according to the level of expertise of the participants. Generally, it is interesting to see that the distribution of query types is similar for both experts and novices.

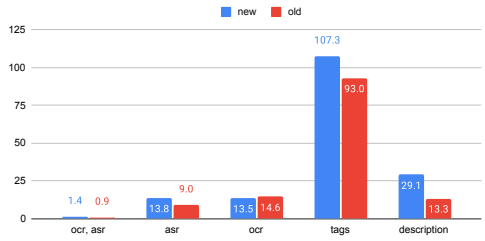


Figure 4.11: 'The number of times the different 'Text' queries were used per version. The numbers are averaged according to the number of users per version.

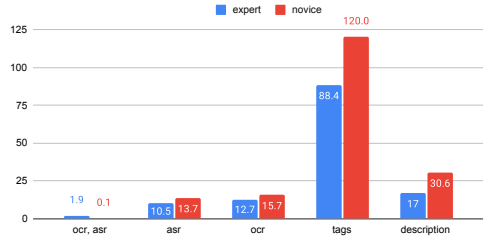


Figure 4.12: This figures gives insight on which query types of 'Text' were used how often per level of expertise. All combinations were used by experts and novices.

4.4.3 Results to the Queries

The user logs contained 1'784 files about the results to the queries. Based on those it was found that 16 out of 25 tasks were solved correctly and that nine were not figured out by at least one participant.

To examine on which rank the correct segment was positioned Figure 4.13 and Figure 4.14 were created. Both show a histogram over the 50 best ranks at which the correct segments were positioned if it was found. Out of 1'452 result sets, 1'001 did not contain the correct segments. For the KIS T tasks, 266 results contained the correct segment. In the case of KIS V tasks, the correct segments were found 185 times. This is not the number of correct submissions from the participants to DRES, but the number of results that contained the correct segment sent from Cineast to vitrivr-ng. Both histograms do not indicate clear differences between the versions used. Oftentimes, the queries are not selective enough in order to find the desired segments at lower (better) ranks. Similar data was collected for the scores of the individual segments but in this case the two versions are not comparable since the scores are calculated differently in each version.

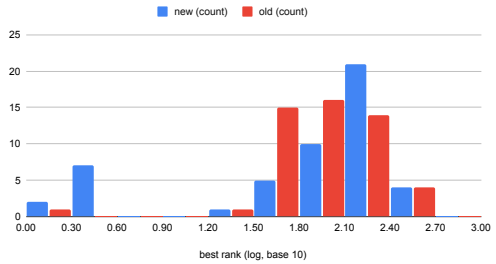


Figure 4.13: The rank distribution of the 50 best results for the KIS T tasks between the new and the old version of vitrivr.

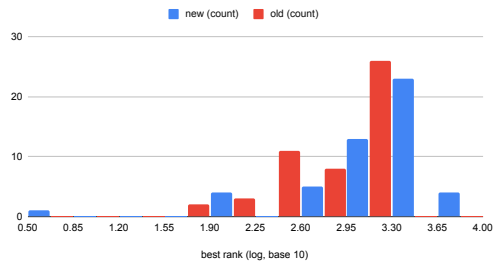


Figure 4.14: The rank distribution of the 50 best results for the KIS V tasks between the new and the old version of vitrivr.

4.4.4 Submissions

For the AVS tasks seven submissions per participant per task on average were recorded, six for KIS V and also six for KIS T tasks. If the users submitted a correct segment, it took them 153 seconds on average to do so. To evaluate if the new additions to vitrivr improved the participants results, the following analysis was conducted.

Figure 4.15 and Figure 4.16 show the average correct submissions the novices and the experts achieved for each of the AVS tasks. Again, the numbers are normalized by number of users that had log entries for the respective task. Both figures clearly indicate that more correct segments were submitted to DRES when using the new version. This is especially noticeable for the novices.

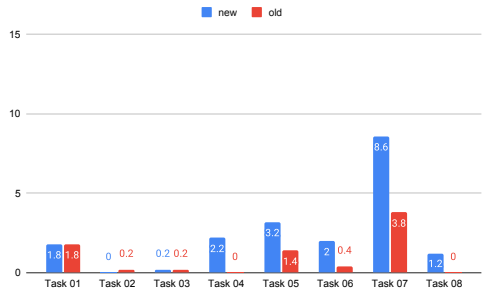


Figure 4.15: The average correct submissions the novices achieved for each AVS task.

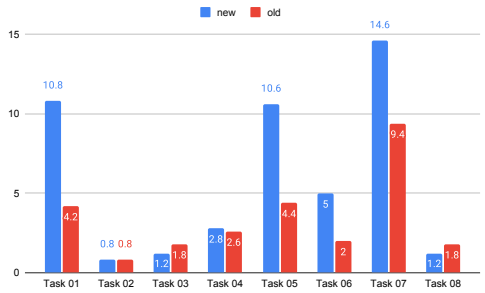


Figure 4.16: The average correct submissions the experts achieved for each AVS task.

The number of correct submissions was also examined for the KIS V tasks and is presented in Figure 4.17 and Figure 4.18. Again, the data is normalized by number of users that had log entries for the respective task. It seems that the novices generally had some problems completing the KIS V tasks but they were more successful using the new version.

Looking at the data for the experts, it was recorded that the number of correct submissions was higher when using the new vitrivr version. To conclude, the number of correct submissions was always higher when using the new version and therefore it can be said that the added features improved the performances for KIS V tasks.

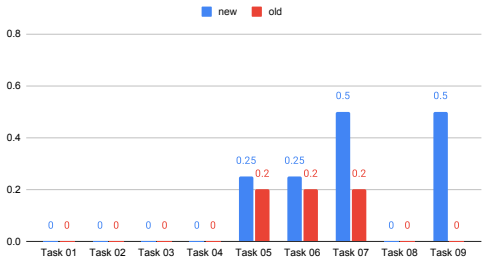


Figure 4.17: The average correct submissions the novices achieved for each KIS V task.

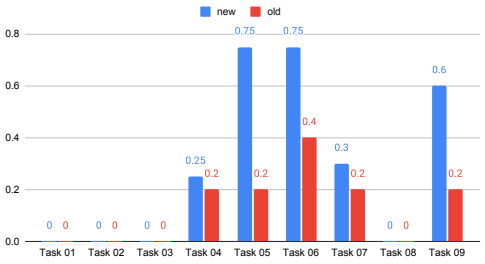


Figure 4.18: The average correct submissions the experts achieved for each KIS V task.

According to Figure 4.19 it is evident that the new additions to vitrivr clearly improved the success of novices for the KIS T tasks. Figure 4.20 shows the same comparison but for the expert group. They also profited from the improvements of the new version whereas the ones with the old version always performed worse. Both plots indicates the average correct submissions for each KIS T task,

normalized by number of users that had log entries for the respective task.
Based on those comparisons it can be said that the new features had a positive impact on the participants achievements.

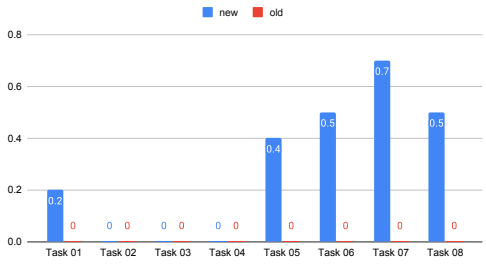


Figure 4.19: The average correct submissions the novices achieved for each KIS T task.

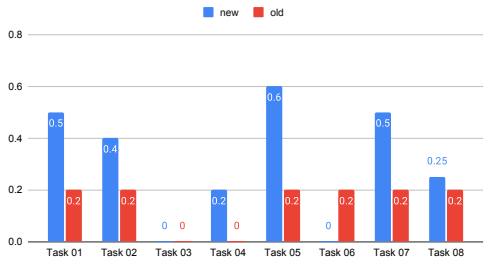


Figure 4.20: The average correct submissions the experts achieved for each KIS T task.

4.4.5 Qualitative Feedback

After the practical evaluation was completed, the participants were asked to fill in a questionnaire in order to get a more qualitative feedback. The survey was conducted using Google Forms[2], which turned out to be a suitable tool. The results can be exported to a spreadsheet for further processing. Unfortunately, we only received 15 out of 17 replies. One participant from each group (vitrivr as-is and vitrivr-new) did not fill in the form. To the questions mentioned below, the participants could answer by selecting a number between one and six, one being the lowest and six the highest possible value.

- How would you rate the usability of the user interface?
  (1 = not good, 6 = very good)
- How easy was it to formulate a query?
  (1 = very hard, 6 = very easy)
- Were you satisfied with the quality of the results?
  (1 = not at all, 6 = very satisfied)
- Do you find the user interface to be visually pleasing?
  (1 = not pleasing, 6 = very pleasing)
- Was it clear how the retrieved results came about?
  (1 = not clear, 6 = very clear)

The average score that was submitted to each of the five questions with respect to the version that was used during the evaluation is summarised in Table 4.2. The usability was rated lower for the new version which could trigger further improvements, so that the new features can be better integrated in the current/known workflow or placed somewhere more intuitively to improve the user experience. Also, the query formulation was rated equally easy with a score of 4.6 out of 6. This shows that the new additions did not make it harder to formulate a query. The participants

[2] Google Forms

seemed to be equally satisfied with the quality of the results by awarding it a rating of 3.4. The highest scores were given to the question about the visual appearance of the user interface which was 4.9 for the as-is and 4.8 for the new version. The fifth question on how results came about was rated higher for the new version at 4.2 whereas the as-is version scored 3.7 out of 6. This indicated that the added features proved to have fulfilled their purpose in creating more transparency of the results.

Which vitrivr **version** did you use?	How would you rate the **usability** of the user interface?	How **easy** was it to formulate a query?	Were you **satisfied** with the quality of the results?	Do you find the **user interface** to be visually pleasing?	Was it **clear** how the retrieved **results** came about?
vitrivr as-is	4.3	4.6	3.4	4.9	3.7
vitrivr new	4.1	4.6	3.4	4.8	4.2

Table 4.2: The average score that was given to each of the questions in the qualitative poll that the participants were asked to fill in after the practical evaluation.

Additionally, two open questions allowed the users to give more elaborate feedback:

> What could be improved to make it more visually pleasing?
> If you have any further comments, suggestions, wishes or input: please feel free to let us know!

In response to the first question a novice who used the old version wished for deeper knowledge about *how* to use vitrivr. The user was confused how the different query containers influenced the query and was unsure which steps to take in order to formulate a query. Before the actual evaluation started, a cheat sheet was distributed and a short introduction to vitrivr was given in order to avoid this. During discussions with the participants it was often mentioned that the UI (vitrivr-ng) was not that intuitive and very powerful. Mostly novices were insecure on how to get started when formulating a query. It has to be noted that it takes time to get acquainted with vitrivr and the entire mindset that is needed to formulate a query. The complete answers that were given to the second open question can be found in Appendix A.4. On the first day of the evaluation one person who used the old version mentioned that the drop down which suggests matching tags was not wide enough so the definition was cut off. This was very helpful feedback and this issue could be fixed for Thursday. Two other suggestions to improve the usability of the UI were added to the official vitrivr to do list.

4.5 Challenges and Lessons Learned

New Tasks and Metrics New Tasks and metrics had to be defined for this evaluation. Soon, it became very clear that the design of the tasks is not a trivial matter. The KIS T tasks need to be formulated so that the segments sought was uniquely identifiable. The challenge for the KIS V tasks was to select segments that contained many discriminative properties which made them unique.

Setup Since the evaluation had to be conducted in a distributed manner via VPN and zoom, connection issues and problems with latency were present during the entire evaluation. Furthermore, result set statistics were pretty slow, especially on nodes equipped with HDDs.

Qualitative Questionnaire That the qualitative questionnaire did not include a question about the cheat sheet was only noticed during the analysis of the results. This should not be forgotten in a future evaluation. The open questions are a curse and a blessing at the same time. They contained many inspiring ideas and suggestions for further brainstorming but sometimes were not clear without some follow-up questions. As the questionnaire was anonymous, it was not possible to identify the author of an idea.

DRES DRES could also profit a lot from this evaluation since we could test it, find issues and have them fixed before and after the evaluation. It became clearer what DRES is able to do and which ideas are not useful in a more general evaluation setting. Questions such as „What else would we like it to be able to do or to know?", „How good is the usability?" were raised and triggered discussions with the DRES-team. Finally, the logging of DRES should also be mentioned. A lot of data is generated during an evaluation that needs to be analysed. Currently, the log files are not of a consistent format or type which created small but surmountable challenge. There is one file with which the entire evaluation should be reconstructable. Also the complete result sets that the users receive from cineast are logged there. But, a log entry is only then added once the QR_END flag was sent to the UI which meant that many result sets were not logged at all and the information was lost. For the 1'724 queries only 1'452 result sets were recorded. This is probably connected to performance problems of the HDDs containing the multimedia data set which leads the user to become impatient and reload the UI. In this case, the result is not sent to the UI and will therefore not be logged.

Logging For quite some users the results to some tasks were missing in the log files. The participants submitted their local log files but unfortunately those did not complete the server-side logs. Especially vitrivr5 was only documented until the fourth task, the remaining ones were nowhere to be found. This had an impact on the reliability of the logs. The quality of the logs could be improved by also logging the queryID so that the queries and results can be better connected.

5 Conclusion

Augmenting a system's transparency can shorten the time it takes a user to get acquainted with the inner workings of the system and consequently lead to better queries and results. This report presented the functionalities which were added to vitrivr, an open-source content-based multimedia retrieval stack. The tag-based queries were improved by introducing the possibility to specify whether a tag *must*, *could* or must *not* be associated with segments in the result set. To further support the users' workflow and increase the system's transparency, attention was paid on making context information available. Hoping to inspire the users to think of new ways of expressing their queries, related tags and frequently occurring words in the captions are presented. Furthermore, feature information can be accessed for an individual element in the result set.

Using DRES, an evaluation comparing the current and new versions of vitrivr was conducted. Due to the distributed environment, technical difficulties arose during the first tasks. Result set statistics were pretty slow, especially on nodes equipped with HDDs. Extending the logged information by a unique query identifier could improve the quality and reliability of the logs, making it possible to map the queries to the results. The interpretation of the evaluation showed that the participants equipped with the new version submitted more correct results to DRES for each of the three task types AVS, KIS V and KIS T. The results did not show any indications that the new additions made it harder for the participants to use the system. Nevertheless,the qualitative feedback indicated that the transparency of the results could still be increased.

In summary, the added functionality to vitrivr presents a good step towards higher query quality and system transparency through explainability.

6
Future Work

The information about an individual element is not well-embedded in the current workflow as it requires multiple clicks to reach captions or related tags for one segment. A possible solution would be an overlay across the lower third of the page which can be faded in when needed instead of jumping to another page. The user would not loose sight of the result set as a whole but still be able to immediately access specific information.

Currently, the history function is not very useful and could be extended. The option to consult previous queries and, more importantly, see the related statistics at a glance would avoid having to recreate past queries from memory or even hand-written notes. Consequently, this could strengthen the foundation of the iterative and interactive process by explicitly presenting preceding iterations.

Other textual search engines [11] annotate each element in the result set with coloured bar graphs to present the contribution each individual query term made to the overall score of the retrieved item. Applying this to the video segments through meaningful sorting and colour-coding, the user could quickly identify the relevance of each search term (or other features) in the result set. A transparent and explainable presentation of the results could reduce the time needed to explore the generated output and augment the user's confidence in the results because they got a chance to better understand the scoring and ranking.

Organising the related tags by means of a thesaurus to expose the relationships between concepts and making them explicit could further contribute to inspire and guide the user when formulating a query. Be it in the form of synonyms or increasing clarity when selecting a preferred term or combination thereof in a query.

The qualitative feedback reported a slightly lower usability for the new version of vitrivr. When talking about its user experience in general, a possible redesign of vitrivr-ng was casually mentioned a few times. *Very powerful but a little cluttered* was often used to describe the user interface. A complete renovation seems drastic and could potentially have unwanted consequences. Nevertheless, identifying the broadly used workflows and features and pursuing their improvements could lead to an enhanced and more usable vitrivr-ng.

Bibliography

[1] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. Multimodal multimedia retrieval with vitrivr. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, page 391–394, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367653. doi: 10.1145/3323873.3326921. URL https://doi.org/10.1145/3323873.3326921.

[2] Ralph Gasser, Luca Rossetto, Silvan Heller, and Heiko Schuldt. Cottontail db: An open source database system for multimedia retrieval and analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 4465–4468, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379885. doi: 10.1145/3394171.3414538.

[3] Silvan Heller, Mahnaz Amiri Parian, Ralph Gasser, Loris Sauter, and Heiko Schuldt. Interactive lifelog retrieval with vitrivr. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*, pages 1–6, 2020.

[4] Silvan Heller, Mahnaz Parian, Maurizio Pasquinelli, and Heiko Schuldt. Vitrivr-explore: Guided multimedia collection exploration for ad-hoc video search. In *International Conference on Similarity Search and Applications*, pages 379–386. Springer, 2020.

[5] Silvan Heller, Loris Sauter, Heiko Schuldt, and Luca Rossetto. Multi-stage queries and temporal scoring in vitrivr. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–5. IEEE, 2020.

[6] 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 *International Conference on Multimedia Modeling*, pages 435–440. Springer, 2021.

[7] Björn Þór Jónsson, Omar Shahbaz Khan, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Jan Zahálka, Stevan Rudinac, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. Exquisitor: interactive learning at large. *arXiv preprint arXiv:1904.08689*, 2019.

[8] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[9] Miroslav Kratochvíl, Patrik Veselỳ, František Mejzlík, and Jakub Lokoč. Som-hunter: Video browsing with relevance-to-som feedback loop. In *International Conference on Multimedia Modeling*, pages 790–795. Springer, 2020.

[10] Jakub Lokoč, Gregor Kovalčík, and Tomáš Souček. Viret at video browser showdown 2020. In *International Conference on Multimedia Modeling*, pages 784–789. Springer, 2020.

[11] Jerome Ramos and Carsten Eickhoff. Explainability in transparent information retrieval systems, 2019. URL https://cs.brown.edu/research/pubs/theses/ugrad/2019/ramos.jerome.pdf.

[12] Luca Rossetto, Ivan Giangreco, and Heiko Schuldt. Cineast: a multi-feature sketch-based video retrieval engine. In *2014 IEEE International Symposium on Multimedia*, pages 18–23. IEEE, 2014.

[13] Luca Rossetto, Ivan Giangreco, Claudiu Tanase, and Heiko Schuldt. vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1183–1186, 2016.

[14] Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Amiri Parian, and Heiko Schuldt. Retrieval of structured and unstructured data with vitrivr. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pages 27–31, 2019.

[15] Luca Rossetto, Mahnaz Amiri Parian, Ralph Gasser, Ivan Giangreco, Silvan Heller, and Heiko Schuldt. Deep learning-based concept detection in vitrivr. In *International Conference on Multimedia Modeling*, pages 616–621. Springer, 2019.

[16] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A Butt. V3c–a research video collection. In *International Conference on Multimedia Modeling*, pages 349–360. Springer, 2019.

[17] Luca Rossetto, Ralph Gasser, Loris Sauter, Abraham Bernstein, and Heiko Schuldt. A system for interactive multimedia retrieval evaluations. In *Proceedings of the 27th International Conference on Multimedia Modeling (MMM 2021)*, 2021.

[18] 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 *International Conference on Multimedia Modeling*, pages 760–765. Springer, 2020.

[19] Klaus Schoeffmann. Video browser showdown 2012-2019: A review. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–4. IEEE, 2019.

[20] Florian Spiess, Ralph Gasser, Silvan Heller, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Competitive interactive video retrieval in virtual reality with vitrivr-vr. In *International Conference on Multimedia Modeling*, pages 441–447. Springer, 2021.

A
Appendix

A.1 Task definition in DRES

DRES offers a browser-based user interface to create tasks. This section presents the different tasks that were posed in the evaluation in more detail and shows what the task registration looks like in DRES.

A.1.1 Known-item-search visual task (KIS V)

Figure A.1 shows how a KIS V task is defined in DRES. In total, nine KIS V tasks were carried out. Screenshots of the individual video segments are shown in Figures A.2 – A.10.

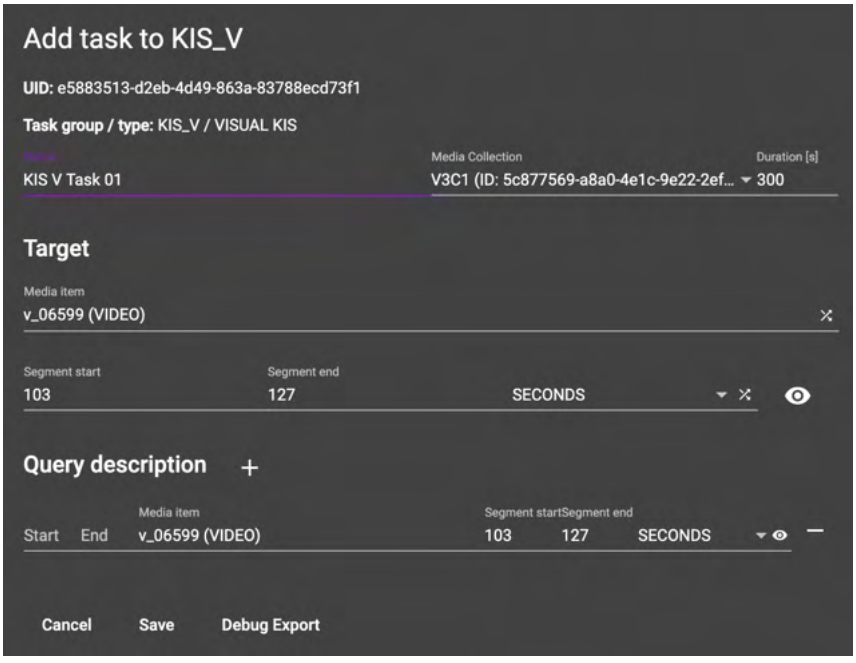


Figure A.1: The definition of a known-item-search visual task.

Figure A.2: KIS V 01

Figure A.3: KIS V 02

Figure A.4: KIS V 03

Figure A.5: KIS V 04

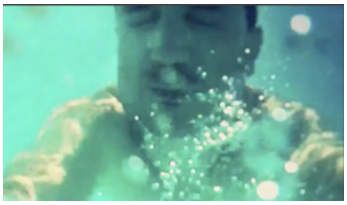


Figure A.6: KIS V 05

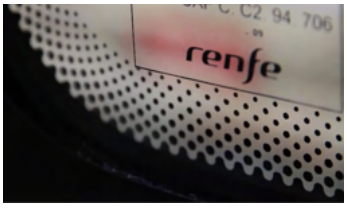


Figure A.7: KIS V 06

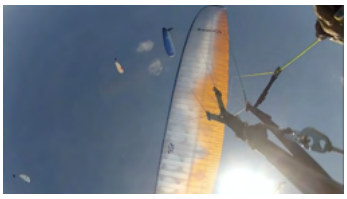


Figure A.8: KIS V 07

Figure A.9: KIS V 08

Figure A.10: KIS V 09

A.1.2 Known-item-search textual task (KIS T)

Figure A.11 shows how a textual KIS task is defined in DRES. In total, eight KIS T tasks were carried out. The individual descriptions can be found in Table A.1.

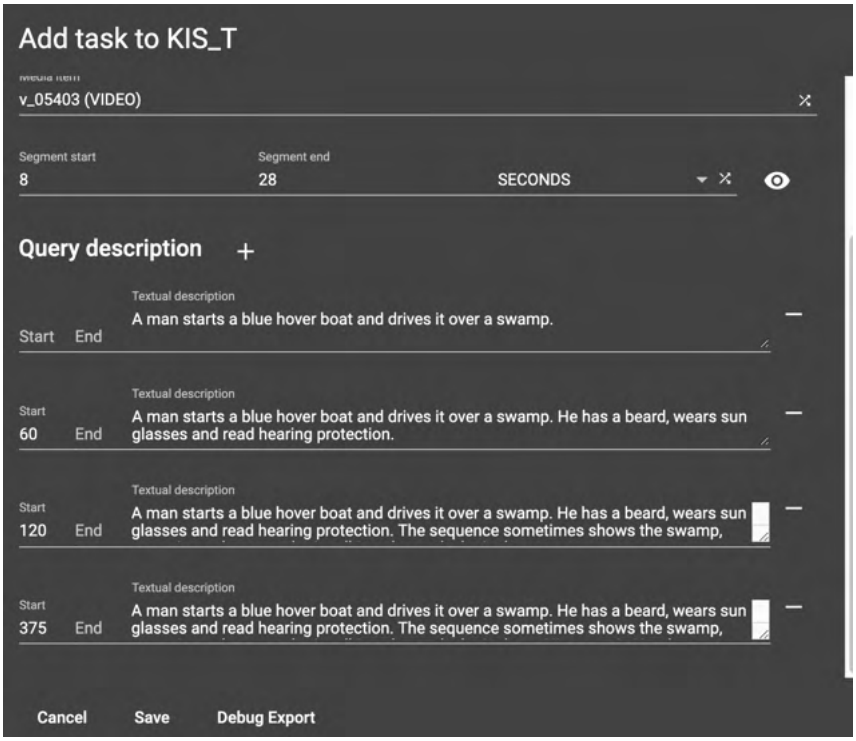


Figure A.11: The definition of a known-item-search textual task.

Task	Description
KIS T 01	A man starts a blue hover boat and drives it over a swamp. He has a beard, wears sun glasses and read hearing protection. The sequence sometimes shows the swamp, sometimes the man who is talking about why he is there. His name is Humberto Jimenez.
KIS T 02	A woman with black hair and a checkered jacket is interviewed and talks about the school that is portrayed. The school is called 'Renbrook School'.
KIS T 03	A time-lapse shows scientists collaboratively working in a huge lab. Most of the scenes are shown in a side-by-side view. Some of them are wearing white coats or overalls and purple gloves. At one time, a scientist is sitting on the floor, working on his laptop.
KIS T 04	New York City from a bike rider's perspective. The camera with a fish eye lens seems to be mounted on the helmet. The street is made up of three lanes and one parking lane (very left). The segment is a time-lapse. In the first shots we see a TGIF restaurant on the left. The bike rider is on the left lane and many taxis can be seen.
KIS T 05	A tent s placed in the snow and the stars in the sky can be seen. A woman on a sled is being pulled through the snow by a group of dogs. The sun is not up yet, it is 2 a.m.
KIS T 06	A guy who is wearing a red jumper is putting on a rubber glove. He seems to be in some kind of garage or workshop. He uses it to clean a bike. A boy in a dark green / brownish hoodie is interviewed in Italian. The workshop is called 'officine la strada'.
KIS T 07	The lead singer of the band 'Fireflight' is a blonde woman. The band is giving a rock concert on a channel called 'sound check'. From behind, red lights illuminate the stage. In the second verse of the song she sings about wanting to open up her eyes.
KIS T 08	A group of men are taking part in an MMA class. The instructor demonstrates a move where he picks up the opponent who is wearing red gloves. The instruction (Esteban) is seen from behind.

Table A.1: The descriptions of the eight KIS T tasks that were posed.

A.1.3 Ad-hoc video search task (AVS)

Figure A.12 shows how an AVS task is defined in DRES. In total, eight AVS tasks were carried out. The individual descriptions can be found in Table A.2.

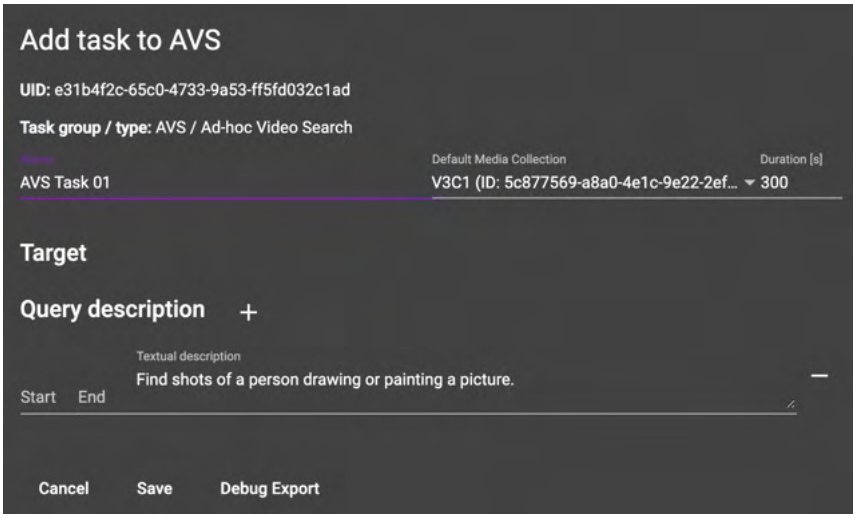


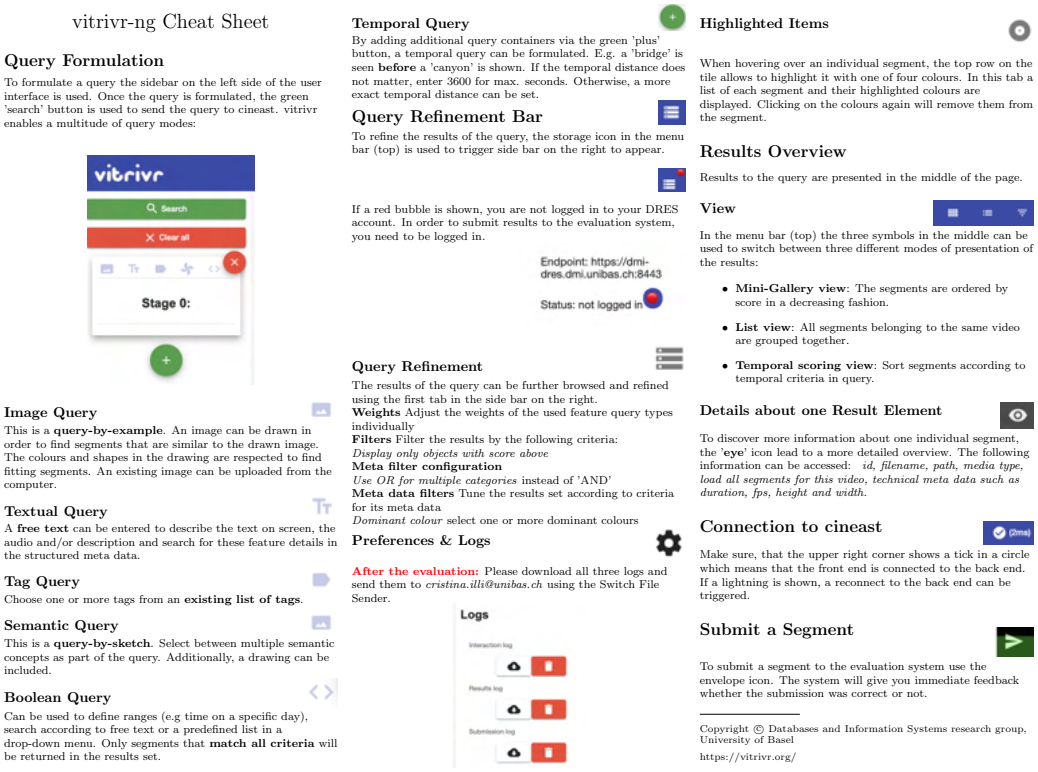
Figure A.12: The definition of a ad-hoc video search task in DRES.

Task	Description
AVS 01	Find shots of a person drawing or painting a picture.
AVS 02	Find shots of an animal chasing another animal.
AVS 03	Find shots of a car race where the cars are crossing the finish line.
AVS 04	Find shots of a group singing a song together.
AVS 05	Find shots of people drinking something in a bar.
AVS 06	Find shots of a person crossing the finish line.
AVS 07	Find shots of people (one or more than one person) having a BBQ.
AVS 08	Find shots of a footballer scoring a touch down.

Table A.2: The descriptions of the eight AVS tasks that were posed.

A.2 Cheat sheet vitrivr as-is

Figure A.13 shows the cheat sheet that was handed out to the participants who used the as-is version of vitrivr.



vitrivr-ng Cheat Sheet

Query Formulation

To formulate a query the sidebar on the left side of the user interface is used. Once the query is formulated, the green 'search' button is used to send the query to cineast. vitrivr enables a multitude of query modes:

vitrivr
Search
Clear all
Stage 0:

Image Query

This is a **query-by-example**. An image can be drawn in order to find segments that are similar to the drawn image. The colours and shapes in the drawing are respected to find fitting segments. An existing image can be uploaded from the computer.

Textual Query

A **free text** can be entered to describe the text on screen, the audio and/or description and search for these feature details in the structured meta data.

Tag Query

Choose one or more tags from an **existing list of tags**.

Semantic Query

This is a **query-by-sketch**. Select between multiple semantic concepts as part of the query. Additionally, a drawing can be included.

Boolean Query

Can be used to define ranges (e.g time on a specific day), search according to free text or a predefined list in a drop-down menu. Only segments that **match all criteria** will be returned in the results set.

Temporal Query

By adding additional query containers via the green 'plus' button, a temporal query can be formulated. E.g. a 'bridge' is seen **before** a 'canyon' is shown. If the temporal distance does not matter, enter 3600 for max. seconds. Otherwise, a more exact temporal distance can be set.

Query Refinement Bar

To refine the results of the query, the storage icon in the menu bar (top) is used to trigger side bar on the right to appear.

If a red bubble is shown, you are not logged in to your DRES account. In order to submit results to the evaluation system, you need to be logged in.

Endpoint: https://dmi-dres.dmi.unibas.ch:8443
Status: not logged in

Query Refinement

The results of the query can be further browsed and refined using the first tab in the side bar on the right.
Weights Adjust the weights of the used feature query types individually
Filters Filter the results by the following criteria:
Display only objects with score above
Meta filter configuration
Use OR for multiple categories instead of 'AND'
Meta data filters Tune the results set according to criteria for its meta data
Dominant colour select one or more dominant colours

Preferences & Logs

After the evaluation: Please download all three logs and send them to *cristina.illi@unibas.ch* using the Switch File Sender.

Logs
Interaction log
Results log
Submission log

Highlighted Items

When hovering over an individual segment, the top row on the tile allows to highlight it with one of four colours. In this tab a list of each segment and their highlighted colours are displayed. Clicking on the colours again will remove them from the segment.

Results Overview

Results to the query are presented in the middle of the page.

View

In the menu bar (top) the three symbols in the middle can be used to switch between three different modes of presentation of the results:

- **Mini-Gallery view**: The segments are ordered by score in a decreasing fashion.
- **List view**: All segments belonging to the same video are grouped together.
- **Temporal scoring view**: Sort segments according to temporal criteria in query.

Details about one Result Element

To discover more information about one individual segment, the '**eye**' icon lead to a more detailed overview. The following information can be accessed: *id, filename, path, media type, load all segments for this video, technical meta data such as duration, fps, height and width.*

Connection to cineast

Make sure, that the upper right corner shows a tick in a circle which means that the front end is connected to the back end. If a lightning is shown, a reconnect to the back end can be triggered.

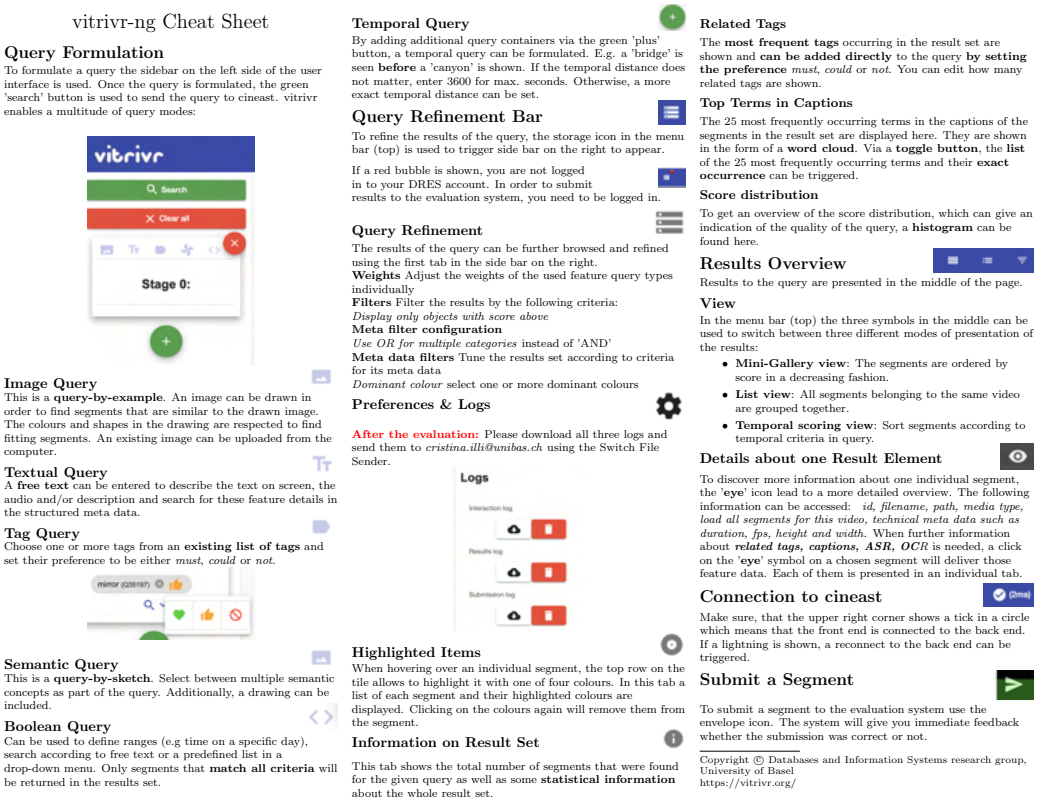
Submit a Segment

To submit a segment to the evaluation system use the envelope icon. The system will give you immediate feedback whether the submission was correct or not.

Figure A.13: A cheat sheet for the as-is version of vitrivr.

A.3 Cheat sheet vitrivr new version

Figure A.14 shows the cheat sheet that was handed out to the participants who used the version of vitrivr that contained new additions.



vitrivr-ng Cheat Sheet

Query Formulation
To formulate a query the sidebar on the left side of the user interface is used. Once the query is formulated, the green 'search' button is used to send the query to cineast. vitrivr enables a multitude of query modes:

vitrivr
Search
Clear all
Stage 0:

Image Query
This is a **query-by-example**. An image can be drawn in order to find segments that are similar to the drawn image. The colours and shapes in the drawing are respected to find fitting segments. An existing image can be uploaded from the computer.

Textual Query
A **free text** can be entered to describe the text on screen, the audio and/or description and search for these feature details in the structured meta data.

Tag Query
Choose one or more tags from an **existing list of tags** and set their preference to be either *must*, *could* or *not*.

Semantic Query
This is a **query-by-sketch**. Select between multiple semantic concepts as part of the query. Additionally, a drawing can be included.

Boolean Query
Can be used to define ranges (e.g time on a specific day), search according to free text or a predefined list in a drop-down menu. Only segments that **match all criteria** will be returned in the results set.

Temporal Query
By adding additional query containers via the green 'plus' button, a temporal query can be formulated. E.g. a 'bridge' is seen **before** a 'canyon' is shown. If the temporal distance does not matter, enter 3600 for max. seconds. Otherwise, a more exact temporal distance can be set.

Query Refinement Bar
To refine the results of the query, the storage icon in the menu bar (top) is used to trigger side bar on the right to appear.

If a red bubble is shown, you are not logged in to your DRES account. In order to submit results to the evaluation system, you need to be logged in.

Query Refinement
The results of the query can be further browsed and refined using the first tab in the side bar on the right.
Weights Adjust the weights of the used feature query types individually
Filters Filter the results by the following criteria:
Display only objects with score above
Meta filter configuration
Use OR for multiple categories instead of 'AND'
Meta data filters Tune the results set according to criteria for its meta data
Dominant colour select one or more dominant colours

Preferences & Logs
After the evaluation: Please download all three logs and send them to *cristina.illi@unibas.ch* using the Switch File Sender.

Logs
Interaction log
Results log
Submission log

Highlighted Items
When hovering over an individual segment, the top row on the tile allows to highlight it with one of four colours. In this tab a list of each segment and their highlighted colours are displayed. Clicking on the colours again will remove them from the segment.

Information on Result Set
This tab shows the total number of segments that were found for the given query as well as some **statistical information** about the whole result set.

Related Tags
The **most frequent tags** occurring in the result set are shown and **can be added directly** to the query **by setting the preference** *must*, *could* or *not*. You can edit how many related tags are shown.

Top Terms in Captions
The 25 most frequently occurring terms in the captions of the segments in the result set are displayed here. They are shown in the form of a **word cloud**. Via a **toggle button**, the **list** of the 25 most frequently occurring terms and their **exact occurrence** can be triggered.

Score distribution
To get an overview of the score distribution, which can give an indication of the quality of the query, a **histogram** can be found here.

Results Overview
Results to the query are presented in the middle of the page.

View
In the menu bar (top) the three symbols in the middle can be used to switch between three different modes of presentation of the results:

- **Mini-Gallery view**: The segments are ordered by score in a decreasing fashion.
- **List view**: All segments belonging to the same video are grouped together.
- **Temporal scoring view**: Sort segments according to temporal criteria in query.

Details about one Result Element
To discover more information about one individual segment, the '**eye**' icon lead to a more detailed overview. The following information can be accessed: *id, filename, path, media type, load all segments for this video, technical meta data such as duration, fps, height and width.* When further information about ***related tags, captions, ASR, OCR*** is needed, a click on the '**eye**' symbol on a chosen segment will deliver those feature data. Each of them is presented in an individual tab.

Connection to cineast
Make sure, that the upper right corner shows a tick in a circle which means that the front end is connected to the back end. If a lightning is shown, a reconnect to the back end can be triggered.

Submit a Segment
To submit a segment to the evaluation system use the envelope icon. The system will give you immediate feedback whether the submission was correct or not.

Figure A.14: A cheat sheet for the new version of vitrivr.

A.4 Qualitative Feedback

To the question: „If you have any further comments, suggestions, wishes or input: please feel free to let us know!" the participants gave the following feedback:

- A refinement option to only display those videos that also contain audio.
- For the three buttons in the middle: show which one is active. It would be helpful to weight the different parts of the query.
- Sometimes the system was very slow which was frustrating.
- See complete tag in drop-down list.
- Select complete text by double click.
- The new vitrivr version would benefit of fine grained formulation of queries: It should be possible to say tag t1 is preferred, but t1 & t2 more, however t2 alone never.
- I like the "must" option for the tags. The "should" option doesn't seem very useful at this point because it still doesn't feel as though results were boosted due to the option. Frankly, I have not used the "must not" option.
- Tags and OCR seem to be the way to go. The other features seem to be not that helpful in a query.
- Maybe would be great if we can add a feature to select few videos, and search within them for the text in screen or audio. there are some tasks for example the fireflight one, it was obvious there is a set of videos, but the other results were clouding the results.
- After submitting a few queries, there seemed to be frequent deadlocks which could only be resolved by the administrator restarting my client - this caused all the current queries to be lost, which was admittedly very frustrating at times.