Is the Reign of Interactive Search Eternal? Findings from the Video Browser Showdown 2020

JAKUB LOKOČ, PATRIK VESELÝ, FRANTIŠEK MEJZLÍK, GREGOR KOVALČÍK, and TOMÁŠ SOUČEK, Charles University, Czech Republic
LUCA ROSSETTO, University of Zurich, Switzerland
KLAUS SCHOEFFMANN, Klagenfurt University, Austria
WERNER BAILER, Joanneum Research, Austria
CATHAL GURRIN, Dublin City University, Ireland
LORIS SAUTER, University of Basel, Switzerland
JAEYUB SONG, Korea Advanced Institute of Science and Technology, South Korea
STEFANOS VROCHIDIS, Information Technologies Institute, CERTH, Greece
JIAXIN WU, City University of Hong Kong, China
BJÖRN ÞÓR JÓNSSON, IT University of Copenhagen, Denmark

Comprehensive and fair performance evaluation of information retrieval systems represents an essential task for the current information age. Whereas Cranfield-based evaluations with benchmark datasets support development of retrieval models, significant evaluation efforts are required also for user-oriented systems that try to boost performance with an interactive search approach. This article presents findings from the 9th Video Browser Showdown, a competition that focuses on a legitimate comparison of interactive search systems designed for challenging known-item search tasks over a large video collection. During previous installments of the competition, the interactive nature of participating systems was a key feature to satisfy known-item search needs, and this article continues to support this hypothesis. Despite the fact that

This article has been supported by Czech Science Foundation (GAČR) project 19-22071Y. Part of this work has received funding from European Union's Horizon 2020 research and innovation programmes under Grant No. 761802 MARCONI and 779962 V4Design.
Authors' addresses: J. Lokoč, P. Veselý, F. Mejzlík, G. Kovalčík, and T. Souček, Faculty of Mathematics and Physics, Charles University, Malostranske nam. 25, Prague, 118 00, Czech Republic; emails: lokoc@ksi.mff.cuni.cz, prtrikvesely@gmail.com, frankmejzlik@gmail.com, gregor.kovalcik@gmail.com, tomas.soucek1@gmail.com; L. Rossetto, University of Zurich, Zurich, Switzerland; email: rossetto@ifi.uzh.ch; K. Schoeffmann, Klagenfurt University, Klagenfurt, Austria; email: ks@itec.aau.at; W. Bailer, Joanneum Research, Steyrergasse 17, 8010 Graz, Austria; email: werner.bailer@joanneum.at; C. Gurrin, Dublin City University, Dublin, Ireland; email: cathal.gurrin@dcu.ie; L. Sauter, University of Basel, Basel, Switzerland; email: loris.sauter@unibas.ch; J. Song, Korea Advanced Institute of Science and Technology, Daejeon, South Korea; email: jsong0327@kaist.ac.kr; S. Vrochidis, Information Technologies Institute, CERTH, 6th Km Charilaou-Thermi Road, Thessaloniki, Greece 57001; email: stefanos@iti.gr; J. Wu, City University of Hong Kong, Hong Kong, China; email: jiaxin.wu@my.cityu.edu.hk; B.Þ. Jónsson, Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark Jaeyub Song: KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea Jiaxin Wu: Run Run Shaw Creative Media Center, City University of Hong Kong, Kowloon Tong, Hong Kong Loris Sauter: University of Basel, Department of Mathematics and Computer Science, Spiegelgasse 1, CH-4051 Basel, Switzerland Luca Rossetto: University of Zurich, Department of Informatics, Binzmühlestrasse 14, 8050 Zurich, Switzerland; email: bjth@itu.dk.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1551-6857/2021/07-ART91 $15.00
https://doi.org/10.1145/3445031

top-performing systems integrate the most recent deep learning models into their retrieval process, interactive searching remains a necessary component of successful strategies for known-item search tasks. Alongside the description of competition settings, evaluated tasks, participating teams, and overall results, this article presents a detailed analysis of query logs collected by the top three performing systems, SOMHunter, VIRET, and vitrivr. The analysis provides a quantitative insight to the observed performance of the systems and constitutes a new baseline methodology for future events. The results reveal that the top two systems mostly relied on temporal queries before a correct frame was identified. An interaction log analysis complements the result log findings and points to the importance of result set and video browsing approaches. Finally, various outlooks are discussed in order to improve the Video Browser Showdown challenge in the future.

CCS Concepts: • **Information systems → Video search**; *Multimedia and multimodal retrieval*;

Additional Key Words and Phrases: Interactive video retrieval, deep learning, interactive search evaluation

ACM Reference format:
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, JÓNSSON-BJÖRN ÞÓR 2021. Is the Reign of Interactive Search Eternal? Findings from the Video Browser Showdown 2020. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 3, Article 91 (July 2021), 26 pages.
https://doi.org/10.1145/3445031

1 INTRODUCTION

The last two decades have brought swift technological progress and there have been many innovations in multimedia data acquisition, storage, management, sharing, and searching. Such innovations have enabled multimedia data to become part of our everyday lives and activities, providing a rich information source for scientific, health, education, industrial, and entertainment application domains. This multimedia data can be used in a variety of applications, from fully automated systems (e.g., self-driving cars) to user-focused systems, such as search or entertainment applications. Regardless of application domain, the ultimate technological challenge is to automatically analyze and understand the data in order to design algorithmic routines that replace monotonous human effort. The focus of this article is on the evaluation of approaches to interactive multimedia content retrieval from large archives. Specifically, we investigate the effectiveness of approaches designed to support an individual when seeking one specific remembered short scene in an entire video collection called *known-item search*. The scenario becomes increasingly relevant to everyday life as the volume of videos watched or stored in personal archives increases beyond what can be considered "sequentially browsable."

This is a challenging research scenario because it is not possible to simply apply proven textual search techniques, producing ranked lists of hundreds or thousands of video shots, to these multimedia archives. Rather, the challenge is to support an individual to quickly and easily find the one particular shot that has been described by an information need. Therefore, advanced multimedia search models and systems are necessary to address this challenge. Advances in machine (deep) learning have substantially increased the effectiveness of visual/audio multimedia data analysis and extraction of semantic concepts. However, search effectiveness with the models is still limited by the size of training sets and uneven distribution of concepts (typically obeying Zipf's law) in various domains. Furthermore, even with human-level annotation models, there are additional constraints for known-item search systems; a user who is interactively searching for a remembered scene usually cannot provide a complete description of the scene due to the fallibility of human memory and a limited capacity to articulate the information need in sufficient detail or accuracy. Therefore, it is more realistic to assume that users provide imprecise and incomplete queries that are not sufficient to find a required scene. Providing a sketch may overcome language problems

if the user has sufficient drawing skills, but the issue of memory fallibility still remains. Another popular approach is searching by an example image, but having an example on hand does not model a realistic scenario for a user. Therefore, there is a need to better understand effective retrieval models and how they can be merged with appropriate interactive user interfaces to enable an individual to refind video content that they have previously seen. This is the challenge that is uniquely addressed by the Video Browser Showdown collaborative retrieval benchmark, which brings together teams from around the world who compete with customized state-of-the-art interactive retrieval engines for video data.

The **Video Browser Showdown (VBS)** results provide a high-level ranking of systems, but, thus far, the competition did not try to explain why the top systems performed well, what are the common trends, and what are the effective unique features. Whereas interaction logging efforts were already tested previously [37, 52], the findings were not conclusive. The goal of this article is to provide the ranking of current systems and also a methodology to analyze and further present the performance of the systems for the community. The contribution of this work can be thus summarized:

- A description of the Video Browser Showdown in 2020, highlighting the operation of the benchmarking challenge and the learnings drawn from the challenge
- A comprehensive review of the state-of-the-art search engine approaches integrated and frequently used in the participating systems
- A performance and interaction analysis of the three top performing systems, using interaction and result logs collected during the competition
- A considered listing of the most important future challenges to be addressed in the domain of interactive known-item and ad hoc search for video archives

The remainder of the article is structured as follows: Section 2 describes the VBS and the tasks performed in VBS 2020. Section 3 then provides a review of the participating systems and their techniques. Section 4 describes in detail how these systems performed in the challenge, partly using an analysis of logs from the top three systems. Finally, Section 5 discusses future plans for VBS and challenges within the domain.

2 VIDEO BROWSER SHOWDOWN

The VBS [8, 36, 37, 52, 59] has become a respected annual evaluation campaign for interactive video search. Since its inception in 2012, this event has been collocated with the **International Conference on Multimedia Modeling (MMM)**. Each VBS event implements a competitive setting to evaluate the participating interactive video search systems in direct live comparison to one another. With the same dataset, tasks, and environment, the setting provides a fair assessment of performance and at the same time enables showcasing systems to the audience. VBS participants perform visual/textual **known-item search (KIS)** tasks and **ad hoc search (AVS)** tasks, during which found items can be submitted to the VBS competition server[1] within a given task time limit. The score of each team is based on the correctness of submitted items and a time to solve a KIS task, where the higher the score, the better the performance of the team. The interactive search setting of VBS complements TRECVID [2, 3], which focuses on automatic ranking approaches.

2.1 Competition Settings

Similar to previous years, VBS 2020 tested *visual known-item search* and *textual known-item search* tasks. Unfortunately, AVS tasks were not considered for this year due to technical issues. The tasks

[1]Source code for the VBS competition server is available at https://github.com/klschoef/vbsserver/.

were selected from the V3C1 [57] dataset, consisting of 1,000 hours of video data. For visual KIS tasks, the teams had to find a random segment (with 20 seconds' duration) within a time limit of 5 minutes, which was repeatedly presented at a projector wall during the task. Even though video distortion of the played scene was suggested before, at the competition, the 20s clips were played without any blurring. The main reason was that effects of query distortion are not yet well understood and need further study [51]. For textual KIS tasks, only a textual description for a 20-second target segment was presented, which additionally required the teams to imagine how the scene could look. Since this is much more challenging than visual KIS, teams get a longer time limit of 8 minutes and additional hints in the form of more textual details after 60 and 120 seconds. It should be noted, though, that one textual KIS task was tested with a 5-minute time limit (during the expert session).

In 2020, 11 systems (10 scoring) competed in the VBS competition. Their submissions are scored by the *VBS Evaluation Server*, which considers search time and the number of wrong submissions for each KIS task. In particular, the base score s for a task i is based on reward r and penalty p, as shown in Equations (1)–(3). The reward is a function of search time t, which results in a linearly decreasing base score from 100 to 50 (the minimum base score s_{min} is set to 50 and the maximum search time for a task t_{max} depends on the task type). The penalty p is a linear function of wrong submissions ws, which are multiplied by 10. Finally, the overall score S for a team for session c is computed as the sum of all base scores for n tasks in the session and normalized to 0-100 final points, by weighting to the best score of the session (out of k teams, as shown in Equation (4)). VBS 2020 had two sessions $c \in \{visual, textual\}$ for aggregation and normalization of session scores.

$$s^i(t, ws) = int(max(0, r^i(t) - p^i(ws))), \tag{1}$$

$$r^i(t) = (100 - s_{min}) \cdot \frac{t_{max} - t}{t_{max}} + s_{min}, \tag{2}$$

$$p^i(ws) = ws \cdot 10, \tag{3}$$

$$S^c = \left(\sum_{i=0}^{n} s^i \right) \cdot \frac{100}{max_{j=1,\ldots,k}(\sum_{i=0}^{n} s_j^i)} \tag{4}$$

$$int(x) = \begin{cases} \lfloor x \rfloor, & \text{if } x - \lfloor x \rfloor \leq 0.5, \\ \lceil x \rceil, & \text{otherwise.} \end{cases} \tag{5}$$

Experts performed both visual KIS and textual KIS tasks. At VBS 2020 the teams had two different expert users (tool instances) trying to solve tasks simultaneously, where the first correct submission was sufficient for each team in each task. In addition, visual KIS tasks were also evaluated with novice users randomly recruited from the audience, in order to give further insight into the usability of a search system. For the first time, VBS in 2020 considered also novice user "rotation" as each novice participated for two different systems after three evaluated tasks.

Outside the competition, for evaluation purposes, some tasks were performed with only one expert and repeated later with the other expert (without any information transfer between the two experts). This should reveal the impact of the user for a system.

2.2 Overview of Evaluated KIS Tasks

The task selection process for VBS 2020 followed the same procedure as in earlier years [37]. Table 4 in the appendix provides a storyboard overview of all tasks completed during the VBS 2020 sessions. Visual KIS tasks are performed by both expert and novice users (the latter is indicated by a superscript N in Table 4). Table 5 in the appendix lists the textual KIS tasks completed during the private and public sessions by the expert users only. Each task description consists of three sentences, of which the first is displayed at the beginning and the other two are delayed and

displayed 60 seconds and 120 seconds into the task. After 120 seconds, the complete description becomes visible.

3 PARTICIPATING SCORING SYSTEMS

In the following, we present an overview of 10 scoring participating systems, starting with a brief summary of the main ideas in Section 3.1, while integrated retrieval and browsing methods are detailed in Section 3.2.

3.1 Brief Introduction of Systems and Their Search Strategies

3.1.1 SOMHunter (Charles University, Czech Republic). SOMHunter [27] relies on intuitive text search and classical browsing of a ranked list, optionally combined with more advanced interactive search approaches based on relevance feedback and **self-organizing maps (SOMs)**. The main workflow begins with a (possibly temporal) text query describing a memorized frame, continues with browsing of top-ranked results, or enables switching to an SOM display. Users can select a few relevant frames as positive examples to update the relevance score maintained for each database frame. This process can be repeated by selecting new examples from updated displays. In addition, users can inspect a video summary, see the temporal context of the given frame or display the most similar frames to a selected frame.

3.1.2 VIRET (Charles University, Czech Republic). The search strategy for the VIRET system [38] consists of iterative query formulation and resulting frame set inspection (using thumbnails) with integrated video preview and summary browsing approaches. Selected frames and their features for text search and query by example image were the same as for SOMHunter (i.e., using the BoW variant of the W2VV++ model [33, 35]). The users usually formulated a (temporal) text query and sometimes combined it with example images or color/semantic sketches [39]. If the searched frame was not found in a result set view, the query was updated.

3.1.3 vitrivr (University of Basel, Switzerland; University of Zurich, Switzerland). vitrivr [58] supports many query formulation modalities [18, 55], of which only a small subset was enabled for the competition, namely Query-by-Sketch (both visual and semantic sketches), Query-by-Example, and various forms of textual query. During the 2020 installment of VBS, the vitrivr team solely used textual queries, mainly tags and captions, both of which were automatically generated [56]. In a few tasks, users additionally used the query mode for text-on-screen. The most common search strategy during the competition was to formulate a tag-based query. One of the major benefits of this approach is that there are a finite number of tags in the system, which are suggested to the user by auto-completion. This enables users to quickly start typing the potential tags, auto-complete them, and issue the query in fast succession—which is crucial in the competitive setup of VBS. Second most, users formulated textual queries for scene captioning. However, they had to guess the proper terminology stored in vitrivr, whereas—due to the auto-completion—tag queries did not have this issue. Furthermore, vitrivr users issued temporal queries to indicate a temporal dependency of the query modalities. Since temporal scoring is a rather new feature in vitrivr, users did not heavily rely on it.

3.1.4 VIREO (City University of Hong Kong, Hong Kong). The VIREO system provides three search modalities for users [47]. They can formulate a query by using color sketch [45], concepts [46], and free text [33]. The result for color sketch retrieval is based on a simple but effective color sketch method [45]. VIREO also extracts multiple kinds of concepts from videos such as objects, actions, and places for users to do concept search. The Universal Sentence Embedding method is used to map the user's input text to the concepts [46]. This system also allows users to

do free text search. The result of this search modality is based on the W2VV++ model [33]. Users can observe the output search results and update the query until they find a satisfying answer. Temporal queries are also supported in the VIREO system. Inspired by the VIRET performance at VBS 2019, the system allows users to define the query at time T and query at time $T + 1$. The system will return the best matches that fit the requirement at time T as well as the requirement at time $T + 1$ [47].

3.1.5 Exquisitor (IT University of Copenhagen, Denmark; University of Amsterdam, Netherlands; Czech Technical University in Prague, Czech Republic). Exquisitor [25] is a research prototype for large-scale interactive learning [26]. The VBS collection is explored using relevance feedback from the user: in each exploration round, positive and negative examples of scene keyframes are used to build a semantic model of the user's intent, and an efficient high-dimensional index is then used to locate the keyframes most likely to be relevant. Once a likely candidate is shown on screen, the user can view the full video using a timeline browser. Text-search functionality is available to support the relevance judgment process, when positive examples of scenes are difficult to find, and filters on metadata allow narrowing the focus of the exploration.

3.1.6 IVIST (Korea Advanced Institute of Science and Technology, South Korea). IVIST [48] provides its users flexibility when finding the target scenes. Four different options are available for searching: object detection, scene-text detection, dominant-color finding, and text retrieval. The user can choose more than one option to search for the target scene and prioritize some of the options so that the system adds more weight to those options. The system will return a ranked list of candidate scenes accordingly. If the system returns too many candidate scenes, the user can narrow down the result by adding more options or by changing the priority of the options.

3.1.7 AAU/ITEC (Klagenfurt University, Austria). AAU and ITEC are two systems from the same research group that build on the same content analysis backend but use different interfaces. While AAU [32] is based on an own shot detection method and a flexible interface with many different search features (search by semantic concepts, recognized objects, color filtering, OCR, and more), the ITEC interface was newly trialed in 2020: instead of shots, it relies on 1-second segments that are indexed with the Lucene Solr server for objects detected by YOLO v3 [50], which is the only provided search feature (in addition to an overview for a video). Unfortunately, the AAU tool provided too many choices for users and was hard to use by novices, while the ITEC tool seemed to be too limited, because it did only provide object search with insufficient possibilities for combinations (not even two objects).

3.1.8 VERGE (Information Technologies Institute, CERTH, Greece). VERGE [1] provides users with all the search capabilities in a compact menu on the left for quickly submitting queries, while the results panel covers most of the screen in order to inspect as many results as possible. A frequent strategy begins with selecting a visual concept to describe what the user is looking for and continues with re-ranking based on a filter (e.g., black-and-white) or a color, if suitable to the case. Once a relevant image appears, visual similarity is applied to retrieve more similar shots. An alternative strategy is to start by searching for relevant keywords inside the transcripts and the video captions.

3.1.9 VNU (University of Science, VNU-HCM, Vietnam). The VNU [30] system integrates models for scene, concept, text, and object color retrieval. The user interface supports various modes enabling expansion of search or result set panels. Since the authors of the system did not join this article, we cannot provide further details of the most recent version of their system used at VBS 2020. Therefore, we also skip a more detailed description of the system in the following section.

3.2 Overview of Methods Integrated to Compared Systems

Before we proceed to the overview of tested approaches, we briefly review current search options. The VBS systems can employ a range of search and browsing modes that together provide means to navigate the video data space. Many systems need to be started with a textual query to obtain an initial result set for further refinement. This initial step is a cross-modal retrieval problem that requires a joint semantic (textual)/visual representation or an appropriate (interactive) interface/method to bridge the semantic gap. Recently, significant advances have been made in learning joint representations (e.g. [33, 68, 69]), but applying these approaches, which typically rely on a set of concept labels, to a topically very broad [4] video collection such as V3C1 is still challenging. Other recently proposed approaches, such as concept-free zero shot retrieval [13], are very promising and might be found in future VBS systems.

In the absence of a good query sample in VBS, similarity search can be initially only performed from sketches. However, once an initial set of related items has been identified, image similarity search is still an important tool to navigate "closer" to searched items. While many descriptors for similarity search have been proposed, for example, in the early MPEG-7 standard [41] and later in the CDVS standard [14], we observe that most systems use learned features instead of handcrafted ones, as also included in the recently completed CDVA standard [15]. One important aspect of VBS, which also sets it apart from other benchmarks, is the focus on the temporal structure of the query (e.g., [40]), which requires fusing the results of often frame-based features along the timeline. Recently, approaches for learning spatiotemporal patterns in video have been proposed (e.g. [66]); however, they do not generalize to cases where a temporal sequence may contain multiple shots that were not continuously recorded but rather manually edited together.

There exist a wide range of methods applicable to solve video search tasks such as those of the VBS. Many of these methods have been discussed extensively in the context of past VBS summaries [8, 37, 52]. Table 1 lists key querying models and browsing methods of nine scoring systems at VBS 2020. Let us note that the competition can be considered as a comparison of these methods in interactive search settings. In the following, we summarize the listed methods based on their category and provide further details.

3.2.1 Text Search. VBS 2020 witnessed various search models based on different text-image matching strategies. The SOMHunter and VIRET systems relied on the same BoW variant of the W2VV++ model [33, 35], a query representation learning approach employing visual features obtained from deep networks trained with a high number of classes [43, 44]. For more details about the employed W2VV++ variant and used similarity for each system, we refer to [35]. Both systems support advanced prompting showing image thumbnails for prompted labels. In addition, the VIRET tool also supported full-text search in ASR data originally generated in [56] and provided with the V3C1 collection to filter videos containing a given text phrase. vitrivr's textual search modalities include tags: a finite set of concepts, suggested via auto-completion [56], free-text input for ASR, OCR, and scene captioning [64]. The AAU tool allowed text search for metadata, concepts detected with GoogLeNet (using batch normalization [23] and trained on ImageNet, Places-365, and Sun397), objects detected with YOLO v3 [50], and OCR [62]. VERGE provided an auto-complete search on predefined visual concepts [42], filters, and activities, as well as a keyword search on transcriptions extracted by a custom ASR technique [1] and video captions [20]. Also, VERGE supported free-text search using a self-attention-based dual encoding network that makes use of multiple encodings' textual content and returns the most correlated keyframes [17]. VIREO allows two ways for users to do text search. The first one is to formulate a query using a free text. The search results are returned based on the W2VV++ model [33] pre-trained on two

Table 1. Selected Search Approaches Integrated and Frequently Used in the Participating Systems, Marked with a Reference to the Paper Describing Features/Method or with ✓ for a Common/Custom Feature

	SOMHunter [27]	VIRET [38]	vitrivr [58]	VIREO [47]	Exquisitor [25]	IVIST [48]	AAU [32]	ITEC [32]	VERGE [1]
Score/Solved Tasks	**91/15**	**87/14**	**79/12**	**61/11**	**59/10**	**50/9**	**47/9**	**45/8**	**35.5/7**
shot detection	[63]	[63]	V3C1	V3C1	V3C1	V3C1	[32]	1 sec	V3C1
text search									
joint embedding	[33, 43]	[33, 43]		[33]		[31]			[17]
concepts			[56]	[46]	[65]		[23]	[23]	[42]
captioning			[64]						[20]
ASR		V3C1	V3C1	V3C1	V3C1				[1]
OCR			[56]	[62]		[11, 61]	[62]		
object detection			[56]			[6]	[50]	[50]	
image search	[33, 43]	[33, 43]	[54]	[45]					[1]
sketch search		[39]	[54]	[45]					
fusion of modalities		[39]	[54]	[45]					[19]
temporal query	[39]	[39]	[58]	[47]					
relevance feedback	[10]				[26]				
top-k from video filter	✓	✓	✓	✓			✓		✓
ranked list	✓	✓	✓	✓	✓	✓	✓	✓	✓
temporal context	✓	✓	✓	✓			✓	✓	
video preview		✓	✓	✓		✓	✓		
video summary	✓	✓					✓	✓	✓
video player			✓	✓	✓	✓	✓	✓	
2D map embedding	✓	✓					✓		

V3C1 means meta-data provided with the V3C1 dataset [57]. The ASR data for V3C1 was provided by [56].

video captioning datasets (MSR-VTT [67] and TGIF [34]). Another way is using concepts to do text search. Many kinds of concepts such as objects, actions, and scenes are extracted from the videos. For the concept extraction, VIREO uses ResNet152 [21] trained on several datasets such as ImageNet [12], ImageNet shuffle [43], OpenImage [29], and Place-365 [70] datasets, and the P3D network [49] trained on the Kinetics dataset [5] to get the concepts [46]. IVIST uses the SCAN model [31] that looks into the latent alignments in images and sentences and predicts the similarity between them. SCAN looks through the V3C1 collection and sorts out the result according to the predicted similarity score. Exquisitor provided a search bar to support the relevance feedback process by facilitating the discovery of positive examples (if a solution was found via search, it could of course be submitted). The text search was implemented using pylucene. By default, the ResNeXt-101 visual concepts for keyframes and their text descriptions [65] were searched, but an option was provided to include the video descriptions and ASR text from the V3C1 collection.

3.2.2 Image and Sketch Search. Currently, a common way to search images by similarity is to rely on deep features from a trained neural network. The SOMHunter and VIRET systems used the same representations for image similarity as for text search (joint space). VIRET also supported

sketch search, where users can place colored circles and two types of semantic objects (face [22], text [72]) with ellipse region boundaries and ALL/ANY specification [39]. VERGE used convolutional neural networks upon a deep hashing architecture to represent images and an IVFADC index database vector for fast binary indexing and retrieval of most visually similar content [24]. The AAU tool used an own HistMap approach [60] for sketch search. The VIREO system computes the similarity between images based on CNN features. VIREO also supports color sketch search, where users can formulate query by placing color in the grid [47]. vitrivr's visual sketch-based search relies on a plethora of features, as described in [54], whereas semantic sketch-based search is based on "*a DeepLab network [7] trained on three image datasets containing concept-instances from different contexts [9, 16, 71]*" [56, Section 3.3]. Issuing visual sketch-based search, users draw a colorized sketch without any restrictions in RGB color space. In contrast, semantic sketch-based search users have to choose from a finite set of concepts and the color-concept mapping is arbitrary.

3.2.3 Fusion Approaches. SOMHunter supports temporal text queries used already by the VIRET system at VBS 2019 [39], where a temporal query can be used to describe a sequence of shots. Since VIRET supports combinations of multiple query modalities, a classical fusion of partial result sets (after possible temporal fusion) is computed as the intersection of the sets and then sorted based on a selected modality [39]. Scoring in vitrivr is a linear combination of the scores of individual feature-modules in a late fusion step [54], based on weights per category. Additionally, vitrivr supports temporal scoring as an alternative fusion function for any query modality [58]. VERGE combined visual concepts and colors using a non-linear graph-based fusion method, where the visual concepts return the top-N relevant shots and the list is re-ranked according to the color [19]. VIREO fuses the normalized scores of each search modality using a linear function and the final result is sorted by the fused score [47].

3.2.4 Relevance Feedback. During VBS 2020, two significantly different approaches incorporating relevance feedback were tested. Exquisitor employs user relevance feedback as its primary user interaction strategy, where the goal of user interaction is to develop a linear SVM model that well captures the information need of the user [26]. At VBS, the goal of this interaction was to identify the most likely candidates to solve each VBS task, thus allowing the user to explore the candidates and identify the correct solution. SOMHunter implemented a Bayesian-like update rule [10] to maintain current relevance scores of frames based on selected positive and implicit negative examples.

3.2.5 Result Set Visualization and Browsing. The most common feature of the systems is to present resulting frames/shots by means of small frame thumbnails. A classical approach is to present a grid with the result set sorted based on the relevance of frames (or shots) with respect to a query (ranked list). For each frame, its temporal context can be presented in a film stripe or as a video preview (e.g., by a mouse wheel). A video summary panel/form with representative frames can provide an overview of the whole video, enabling fast navigation. Some systems directly integrate a video player, where a particular shot can be verified before submission.

In addition to these basic features, systems test several other browsing and visualization approaches. SOMHunter provides a dynamic self-organizing map visualization fitting a result set feature distribution (in the representation space), incorporating also current relevance scores. The VIRET tool enables browsing a smaller static hierarchical image map computed for the whole dataset in advance using self-organizing maps as well. Both systems support a presentation filter for showing just a specified number of top-ranked images from one video and shot. The AAU tool provided several maps with a similarity arrangement (either by color or fingerprints from deep

Table 2. Scoring Teams at VBS 2020

					Solved KIS Tasks				
Rank	System	T-KIS	V-KIS	Score	T	/	V	/	V^N
1.	SOMHunter	82	**100**	**91**	**8**	/	**5**	/	**2**
2.	VIRET	88	86	87	**8**	/	4	/	**2**
3.	vitrivr	**100**	58	79	**8**	/	3	/	1
4.	VIREO	32	90	61	4	/	**5**	/	**2**
5.	Exquisitor	50	68	59	5	/	4	/	1
6.	IVIST	43	57	50	5	/	3	/	1
7.	AAU	70	24	47	7	/	2	/	0
8.	ITEC	53	37	45	5	/	2	/	1
9.	VERGE	25	46	35.5	3	/	3	/	1
10.	VNU	15	15	15	1	/	2	/	0

The last three columns show the number of solved textual expert (T), visual expert (V), and visual novice (V^N) KIS tasks.

neural networks), some of them are filtered by specific concepts (e.g., there is an own map for car, snow, tree, etc.). In the ITEC tool a Lucene-based ranking of x-second segments was used, whereas the user could select $x \in \{1, \ldots, 30\}$. In VERGE the results of each search module were displayed in a grid view, either as single images or groups of images (videos), always sorted by highest relevance. VIREO provides a nearest neighbors preview for each video shot and some filters for browsing such as only showing video shots in grayscale and only showing video shots with a black border.

4 VIDEO BROWSER SHOWDOWN RESULTS

The key contribution of this article lies in fair comparative evaluation efforts of all the complex systems presented in Section 3. Not only must all the experimental systems be designed, implemented, and prepared for the large V3C1 collection, but also they must be fully functional, robust, and user friendly for interactive live evaluations with time constraints. In the first part of this section, we present and discuss overall results of all scoring systems for all 22 known-item search tasks evaluated with the VBS server, where tasks are presented and all submissions are collected. The following subsections then present a more detailed analysis of the top three systems that integrated logging mechanisms for user interactions and query result sets.

4.1 Overall Results and Submission Times

Ten teams solved at least one textual and one visual known-item search task. Table 2 shows the overall results of the competition, where the teams are sorted with respect to the achieved score. The results show a clear correlation between the overall score of teams and the number of solved tasks by the teams. The higher number of solved textual KIS tasks can be attributed to a longer time limit and a higher number of 10 textual KIS tasks evaluated by expert users, compared to six visual KIS tasks evaluated by experts.

To present a more detailed insight of the performance of participating teams/tools, Figure 1 shows search times (cells with a green background) observed by the server when a correct submission was received for a team and task. In addition, a red-font text with white background reveals cases where teams were close to solve a task (correct video ID occurred in an incorrect task submission). The value in the lower index and brackets presents the instance member who solved the task for a team.

	SOMH.	VIRET	vitrivr	VIREO	EXQ.	IVIST	AAU	ITEC	VERGE	VNU	SOLVED
T_1	$412s_{(1)}$	$317s_{(1)}$	$478s_{(2)}$	$186s_{(1)}$			$273s_{(2)}$	$166s_{(2)}$			60%
T_2	$348s_{(2)}$	$118s_{(1)}$	$89s_{(1)}$				$477s_{(2)}$	$478s_{(2)}$			50%
T_3	$84s_{(1)}$	$236s_{(1)}$	$84s_{(2)}$	$377s_{(2)}$		$449s_{(1)}$			$87s_{(2)}$		40%
T_4	$88s_{(2)}$	$22s_{(1)}$	$119s_{(2)}$	$276s_{(1)}$	$413s_{(1)}$		$91s_{(1)}$	$221s_{(1)}$	$171s_{(2)}$		50%
T_5	$59s_{(1)}$		$90s_{(2)}$		$268s_{(1)}$	$192s_{(1)}$		$367s_{(2)}$			50%
T_6	$32s_{(2)}$	$21s_{(1)}$	$110s_{(2)}$	$138s_{(2)}$	$116s_{(1)}$	$164s_{(1)}$	$211s_{(2)}$	$34s_{(2)}$	$344s_{(2)}$	$74s_{(1)}$	100%
T_8	$53s_{(1)}$	$419s_{(1)}$									20%
T_{10}		$131s_{(1)}$			$180s_{(2)}$		$479s_{(2)}$				30%
T_{15}	$254s_{(2)}$	$262s_{(1)}$	$89s_{(2)}$	$257s_{(2)}$	$197s_{(1)}$	$302s_{(1)}$	$296s_{(2)}$		$211s_{(1)}$		80%
T_{18}	$297s_{(1)}$	$266s_{(1)}$	$142s_{(2)}$		$247s_{(2)}$	$169s_{(1)}$	$79s_{(2)}$		$262s_{(1)}$	$170s_{(1)}$	60%
SOLV. T	80%	80%	80%	40%	50%	50%	70%	50%	30%	10%	54%
V_7	$134s_{(1)}$	$52s_{(1)}$	$60s_{(2)}$		$186s_{(2)}$	$136s_{(1)}$				$147s_{(1)}$	60%
V_9	$53s_{(2)}$	$101s_{(1)}$		$87s_{(1)}$	$75s_{(1)}$	$195s_{(1)}$		$203s_{(2)}$	$212s_{(2)}$	$174s_{(1)}$	80%
V_{11}	$125s_{(2)}$			$62s_{(2)}$			$175s_{(1)}$	$299s_{(1)}$		$297s_{(1)}$	30%
V_{16}	$85s_{(1)}$			$102s_{(2)}$				$297s_{(2)}$		$78s_{(1)}$	20%
V_{17}		$219s_{(2)}$	$257s_{(1)}$	$269s_{(1)}$	$85s_{(1)}$		$99s_{(2)}$	$223s_{(2)}$	$103s_{(2)}$	$106s_{(1)}$	70%
V_{19}	$31s_{(2)}$	$86s_{(1)}$	$33s_{(1)}$	$138s_{(1)}$	$207s_{(1)}$	$52s_{(1)}$			$114s_{(2)}$		70%
SOLV. V-E	83%	67%	50%	83%	67%	50%	33%	33%	50%	33%	55%
V_{12}^N				$234s_{(2)}$							10%
V_{13}^N	$218s_{(2)}$						$293s_{(2)}$	$228s_{(1)}$	$298s_{(1)}$		20%
V_{14}^N				$103s_{(2)}$							10%
V_{20}^N		$188s_{(1)}$			$185s_{(1)}$	$261s_{(2)}$					20%
V_{21}^N					$176s_{(1)}$						10%
V_{22}^N	$105s_{(2)}$	$298s_{(1)}$	$188s_{(1)}$	$179s_{(2)}$			$297s_{(2)}$	$170s_{(1)}$			40%
SOLV. V-N	33%	33%	17%	33%	17%	17%	0%	17%	17%	0%	18%
SOLV. ALL	68%	64%	55%	50%	45%	41%	41%	36%	32%	14%	45%

Fig. 1. Search times of correct submissions of each team in VBS search sessions. Red text shows first incorrect submissions with correct video ID in unsolved tasks (the team found the video, but not the scene).

Overall, more than 50% of all possible correct submissions were collected by the VBS server for both expert textual and visual sessions. Considering the top three performing systems, the percentage of correct submissions was impressive, 75% for all expert KIS tasks, where the most successful team, SOMHunter, solved even more than 80% of all expert KIS tasks. The overall percentage of correct submissions in novice sessions was lower, reaching 18%. There are generally two reasons for this. First, the limited number of novice users for 10 systems did not support the use of two instances of each tool, compared to the expert sessions. Second, novice users were not familiar with the systems and observed the expert users just for a few expert tasks. After this short system "presentation," novice users had to solve the task without any help from the experts.

From the task perspective, there appeared tasks that turned out to be either easier (e.g., T_6, T_{15}, V_9) or more difficult (e.g., T_8, V_{16}) for most teams. For example, task V_{16} was challenging as there were many videos with biking in nature. The times to solve a task vary from very fast submissions within 30 seconds up to late submissions close to the task's time limit. This indicates that an ideal query does not have to be always available and so users have to interact with the systems (see Figure 4).

As mentioned earlier, the red text shows for a team the time of the first submission with the correct video ID for unsolved tasks. We may observe that except for two teams (vitrivr and IVIST), all remaining teams submitted an incorrect frame from a correct video in some unsolved KIS tasks. In other words, the teams found the correct video but faced problems in identifying a correct shot within the video. For example, the difference is considerable for the VNU team that officially solved three KIS tasks but found the correct video (at least) for seven KIS tasks. Similarly, the ITEC team found the correct video of six different visual KIS tasks but solved "just" three out of them. This indicates that effective video browsing is an important feature of a known-item search system.

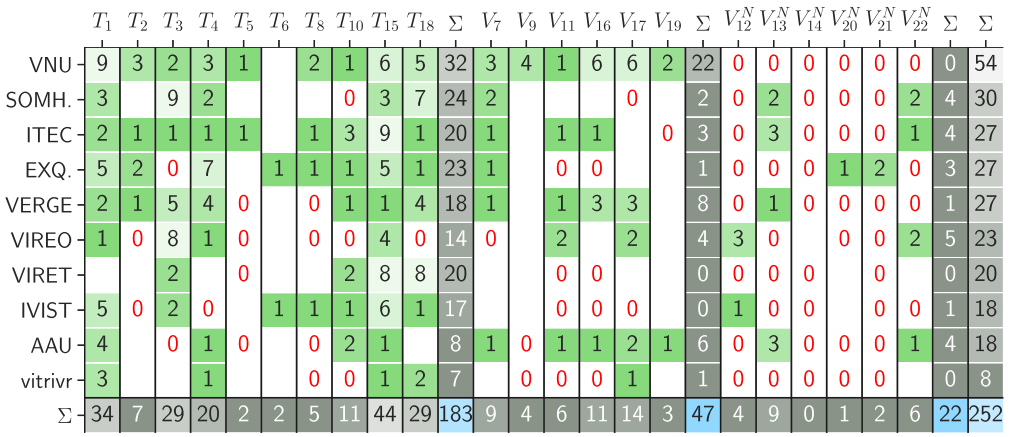


Fig. 2. Wrong submissions for each team and task. Tasks without any submission are denoted by zero.

Figure 2 presents the number of incorrect submissions by a team in a task, which influences the achieved score from that task. The most "careful" team was vitrivr with just eight incorrect submissions. The second team was actually the VIRET team as it turned out that the system prototype sent each submission twice due to an implementation issue. In other words, the real number of incorrect attempts was just 10. It had no penalty effect for correct submissions, but the penalty was twice as high for incorrect submissions. The winning team, SOMHunter, had 30 incorrect submissions, which did not affect its overall VBS rank. From the task perspective, there were several tasks where users faced problems to identify and submit the correct shot. Especially textual KIS tasks are challenging for videos with similar repeating contents due to limited information presented in text form. For example, tasks T_1, T_3, T_4, T_{15}, and T_{18} were challenging for many teams in terms of identification of a correct part of the video (task T_{18} had the time limit of just 5 minutes).

4.2 Result Log Analysis

In addition to submissions collected at the VBS server, the teams were asked to send result logs for each evaluated query during the VBS 2020 competition. The main motivation for this feature was to track the position of the searched scene/video frame in the current result list and better understand the ranking performance of each tool. The log format consisted of the tool/instance identifiers, timestamp, query specification, and a longer list of top-ranked frame identifiers. Though the format is simple, most of the teams faced a lack of time to prepare the tools or forgot to add the timestamp. After the analysis of collected result logs, sufficient data were collected just from the top three performing systems. A similar outcome was also noted for interaction logs analyzed in Section 4.3. Hence, in the following we thoroughly analyze, investigate, and discuss the performance of the SOMHunter, VIRET, and vitrivr systems. Furthermore, we provide additional insight into the top two performing systems that furnished query specifications in the logs. For additional insights, we refer readers to another recent study involving also a performance analysis of SOMHunter and VIRET at VBS 2020 [35].

As in the previous VBS journal report [52], the log record timestamps needed synchronization with the VBS server time and all actions outside the task time frame of each team were filtered out. More specifically, for a team, only logged results and interactions with timestamp

	SOMH. I1				SOMH. I2				VIRET I1				VIRET I2				vitrivr I1				vitrivr I2			
	r_s	r_v	t	t_{cs}	r_s	r_v	t	t_{cs}	r_s	r_v	t	t_{cs}	r_s	r_v	t	t_{cs}	r_s	r_v	t	t_{cs}	r_s	r_v	t	t_{cs}
T_1	-	1	226s	412s	80	37	400s	-	-	2	179s	317s	82	82	60s	-	183	183	426s	-	183	183	19s	478s
T_2	22	22	241s	-	7	7	332s	348s	2	2	71s	118s	403	403	83s	-	1	1	86s	89s	8	8	21s	-
T_3	13	13	37s	-	239	45	131s	-	-	1	96s	236s	49	18	144s	-					-	1	76s	84s
T_4	6	3	71s	-	10	6	20s	88s	4	1	17s	22s					217	162	74s	-	39	38	57s	119s
T_5	4	4	39s	59s	197	197	20s	-	1	1	201s	-	1	1	224s	-	14	14	27s	-	1	1	88s	90s
T_6	13	13	18s	-	13	13	19s	32s	1	1	17s	21s	-	6516	17s	-	-	167	106s	-	3	3	105s	110s
T_8	-	10	29s	53s	239	239	22s	-	10	10	320s	419s	24	13	142s	-								
T_{10}	391	189	452s	-	308	308	236s	-	12	2	46s	131s	11	2	116s	-								
T_{15}	-	14	103s	-	-	4	174s	254s	-	12	30s	262s	3	1	224s	-					424	1	22s	89s
T_{18}	-	3	78s	297s	-	2	238s	-	-	1	226s	-	403	6	69s	-	3	3	140s	-	383	10	39s	142s
V_7	-	48	47s	134s	-	18	61s	-	35	26	14s	52s	-	300	14s	-					131	1	14s	60s
V_9	-	171	39s	-	5	5	46s	53s	1	1	93s	101s	411	411	86s	-					-	6	35s	-
V_{11}	-	48	54s	-	36	16	114s	125s	454	288	151s	-	658	627	236s	-								
V_{16}	-	1	52s	85s	-	16	24s	-	73	73	121s	-	7	7	33s	-	-	166	35s	-	-	28	44s	-
V_{17}	-	773	288s	-	664	439	210s	-	133	108	150s	-	-	5	188s	219s	113	47	146s	257s				
V_{19}					16	16	17s	31s	15	15	40s	86s	693	693	55s	-	60	42	25s	33s				
V_{12}^N	X	X	X	X	-	1	132s	-	39	39	279s	-	X	X	X	X	X	X	X	X				
V_{13}^N	X	X	X	X	57	57	75s	218s	-	7	9s	-	X	X	X	X	X	X	X	X	-	254	122s	-
V_{14}^N	X	X	X	X	587	587	144s	-	356	356	251s	-	X	X	X	X	X	X	X	X				
V_{20}^N	X	X	X	X	-	10	277s	-	-	1	116s	188s	X	X	X	X					X	X	X	X
V_{21}^N	X	X	X	X					4303	1775	135s	-	X	X	X	X					X	X	X	X
V_{22}^N	X	X	X	X	3	3	76s	105s	-	1	180s	298s	X	X	X	X	-	489	48s	188s	X	X	X	X

Fig. 3. Green cells show the best achieved logged rank r_s in time t of a searched scene frame in a task. The best rank r_v of a correct video frame from the same result log is included, while t_{cs} presents the time of the tool's correct submission. Red values are for the best detected ranks of searched video frames if searched scene frames were not present in the logged result sets for a task.

$t \in [t_{TaskStart}, MIN(t_{TaskLimit}, t^{team}_{solved})]$ were considered, where $t^{team}_{solved} < t_{TaskLimit}$ if the team solved the task in time t^{team}_{solved}. We would also like to emphasize that the set of collected logs might be incomplete and so the analysis represents an approximation of all the performed actions. For example, the vitrivr system's result presentation (especially the temporal scoring view) re-orders the query results, without logging this re-ordering, or the logs could be lost due to unreliable transmission. Since SOMHunter and VIRET collected the same logs also locally, we rather used these local logs as the timestamp differences were ascertainable from server submissions. Moreover, thanks to collected interaction logs, we were able to correct several query specification issues that appeared in some result logs.

4.2.1 First-Page-Hit Queries. The first question addressed in this section is whether the users were able to formulate a query to obtain a searched scene frame on a first page of the result list. Based on the available logs for each task, Figure 3 highlights with a green background the minimal detected rank r_s of a frame from the searched KIS scene (from all instance result logs from the task). The rank is accompanied with the elapsed time t of the corresponding query from the task start and the top rank r_v of a searched video frame from the same result log. Let us note that some missing (or too high) numbers of ranks in some cells might be caused by the second tool instance solving a task very quickly (e.g., in tasks T_4, T_6 by VIRET user I1). In many cases, the teams were able to get the searched scene frame on the first page of the result sets. If the frame from the searched scene was not present in the logs, the table shows also red text with the minimal rank of a frame from the searched video. For visually similar video content, it might turn out that the searched shot frames are missing (e.g., due to presentation filters). On the other hand, based on the similar visual content, users could recognize the correct video and enter/inspect a video summary or temporal context or use a video player. The presence of correct video frames with a very low rank for solved tasks indicates that a visually similar video frame may be a sufficient clue to inspect the correct

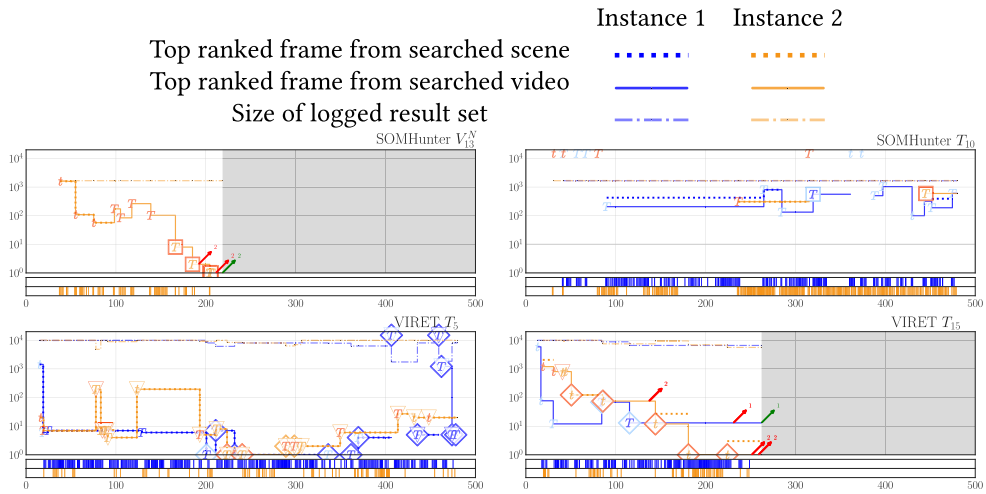


Fig. 4. Search diagrams for selected VBS tasks. Each diagram shows the current top position of a searched shot/video frame from query result logs. Browsing interactions are depicted below at their time of occurrence. For more examples of searches, see [35].

video. For example, both users of the VIRET and SOMHunter instance 1 solved two textual KIS tasks just based on a top-ranked frame from the correct video. The table presents also the correct submission time of a given instance t_{cs} that reveals the time duration between the best achieved rank and the correct submission. In many cases where a searched scene frame was on a top-ranked position, the gap is not too long. The empty cells in the last two columns represent tasks where the vitrivr team was unable to find the searched video, which is reflected by Figure 1. Additionally, vitrivr only logs the top 1,000 results, in order to not clutter the log. It might also happen that some log files were lost due to implementation or technical issues. For example, we cannot guarantee that the last task V_{22}^N was solved by vitrivr based on a first retrieved frame from the searched video at rank 489. Therefore, we repeat that the presented log data should be understood as an approximation of the real performance.

4.2.2 Selected Search Diagrams. A huge benefit of the result logs is that they enable (partial) reconstruction of the search process during a task. To depict such information for a task and team, we construct timeline diagrams from the task start to task end, showing the position of the top-ranked searched frame detected in the result log and browsing interactions below (see Figure 4). Three options are distinguished: a frame from the searched shot is present (dots), at least a frame from the searched video is present (solid lines), or neither of the two (symbol at the top of the graph). Since detailed SOMHunter and VIRET result logs contain query specification information, symbols are displayed at each moment a query was changed. The changed part of the query is highlighted where it is needed. The symbols are defined [35] as follows:

- t represents a text query, while T corresponds to a temporal text query.
- Rhombus represents an example image used.
- Triangle corresponds to semantic sketch search.
- Rectangle shows the usage of a relevance feedback model.
- Red/green arrows depict submission attempts.

The first timeline shows a successful search of a SOMHunter novice user who started with a single text query reformulation, then tried several temporal queries and ended with providing



Fig. 5. Uniformly sampled frames for textual task T_5: "Shot of a harbor crane in front of mountains, then of three harbor cranes on the left, a ship on the right. Ship is seen from the front right, with text 'PROTECT THE ENVIRONMENT' and 'NO SMOKING'. The harbor cranes are blue and white."

positive examples to update the text query scores. After two incorrect submissions, the searched scene frame was submitted. The second SOMHunter timeline presents textual KIS task T_{10} search by expert users who did not manage to get the searched scene frame to a top-ranked position (though different queries and search features were tested). Hence, even with state-of-the-art text search models, KIS tasks might pose a difficult challenge.

During task T_5, both VIRET users had the searched frame on the first page (even at rank 1) for most of the task time. However, in this case the frame was not overlooked, but both users misinterpreted the text description. The users did not find the description matching the correct frames (see Figure 5) and did not want to risk a wrong submission, though both discussed these found frames for some time. Specifically, the "crane in front of mountains" description was not matching at all the idea of mountains for both the users (both fans of hiking in high mountains). On the other hand, the "mountains" specification might be clear for someone from flat plains. The additional information about the text on a ship was not helpful as VIRET uses smaller thumbnails and so the text was not legible. The second timeline for VIRET shows task T_{15}, where the expert user solved the task based on finding a frame from the correct video and subsequent browsing.

4.2.3 Transitions between Query Types. The timelines illustrate the search process; however, due to space restrictions, we cannot present all the searches for both detailed systems. Hence, we aggregate the timelines into transition diagrams in Figure 6, a more detailed version of the diagram presented in [35]. Each diagram shows the start node, nodes for selected query specification types, and node representing a correct submission. The labels of directed edges from the start node indicate which query type was used initially. The numbers in brackets show the number of cases where the result set did not contain a frame from the video (i.e., unsuccessful initialization). The labels of directed edges to the correct submission node indicate what query type was used before the task was solved. The nodes for query types differ for the teams, but the labels show how often the transition info from logs improved/worsened the position of the top-ranked frame from the searched video (i.e., not exclusively the searched scene). In addition, the number of all transitions between nodes is presented as well with gray color. The value counts also cases where the relative outcome of the transition is neutral or unknown.

The SOMHunter diagram shows nodes for simple text query (t) and temporal text query (T), both for the W2VV++ model. Using example image search is presented with node (I), while (O) represents other ranking options (e.g., text-based initial score and relevance feedback-based update of scores). We may observe that users mostly started with a temporal text query and also solved most tasks after a temporal text query. Since SOMHunter did not create a new log entry for potential return from the example image search exploitation display, the arrows from node (I) lead to the query state of the subsequently created result set log. From interaction logs we confirmed that both correct submissions were issued during inspection of the exploitation display. The VIRET diagram has the same nodes (t) and (T) for text search only with the W2VV++ model. The node ($I \in O$) corresponds to any query involving an example image (including a combination with a text query and/or other modalities), while the node ($I \notin O$) represents all remaining query options. In VIRET, users always started with a query for the W2VV++ model except two cases, where textbox for speech data was used. Regarding the remaining options, the sketches were not

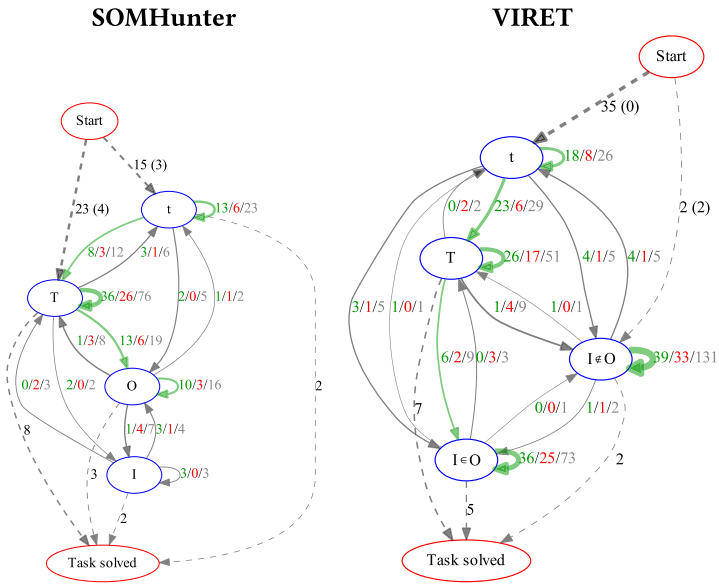


Fig. 6. Transitions between query types. Edge labels between blue ellipses nodes denote how often a transition improved/worsened the logged position of a top frame from searched videos; the third value presents the number of all transitions. Result logs without searched video frames are included. The diagram is an extended version of the diagram presented in [35].

used initially, it is not possible to enter directly a temporal text query (though it might be intended then), and example images are usually selected later from a result set. Similar as for SOMHunter, temporal text queries based on the W2VV++ model were often used to solve a task.

4.2.4 Lessons Learned. The first attempt of result set logging provided valuable experience. Though the format is simple, only three systems provided sufficient logs for some analysis and only two systems furnished sufficient query specification info. This indicates that more effort and specifications are necessary to successfully enforce the logging initiative and thus increase the scientific value of VBS. We also admit that perfect logging by all participating teams might be too ambitious (especially for new teams). Nevertheless, regularly participating experienced teams aiming at more thorough insights of the achieved results should invest more time to collect evidence clarifying their performance. The following list presents selected lessons and ideas that should be considered for future events.

- Given a unified specification, log visualization tools should be available for teams to debug logging implementations before the competition.
- It is necessary to clarify and unify visualization aspects for result logs, as some logs may contain frames added by the visualization component (e.g., temporal context) or local reorganizations. In other words, define whether the ranked set or displayed set should be logged.
- For easier log visualization, result logs should additionally contain information on the type of query. The full query specification should be included in the log as well, including entered values.
- Though query info duplicates interaction logs, we found it useful to keep some basic query representation in the interaction logs as incremental changes can be used to check and

debug result logs. On the other hand, interaction logs are challenging to implement properly and thus benefits of the collected interaction logs were limited so far. Hence, it might be more feasible to insist on result logs at VBS.

4.3 Interaction Log Analysis

Whereas Table 1 presents a list of implemented features by each tool, it is desirable to reveal features that were actually used during the competition. In order to capture this information, a unified interaction log format with predefined vocabulary of interactions was provided in advance, similar as for the VBS 2019 summary article [52]. The log format enables the logging of every action from five different categories and further specifies the action type, timestamp, and other attributes for additional information. All the logged actions were filtered to task time intervals in the same way as result logs. Based on the filtered logs, Table 3 presents chord diagrams for all three sessions, depicting logged interactions (colors on the perimeter) and basic transition statistics between two consecutive interactions (links between perimeter sections). Let us remember that these diagrams remain inappropriate for comparative analysis between tools (see discussion in Section IV in [52]). For example, the logging frequency might be different for teams and even action types logged by one team (typing a whole word/sentence vs. mouse wheel). Hence, the intended purpose of the diagrams is to visualize logged interactions and their transitions.

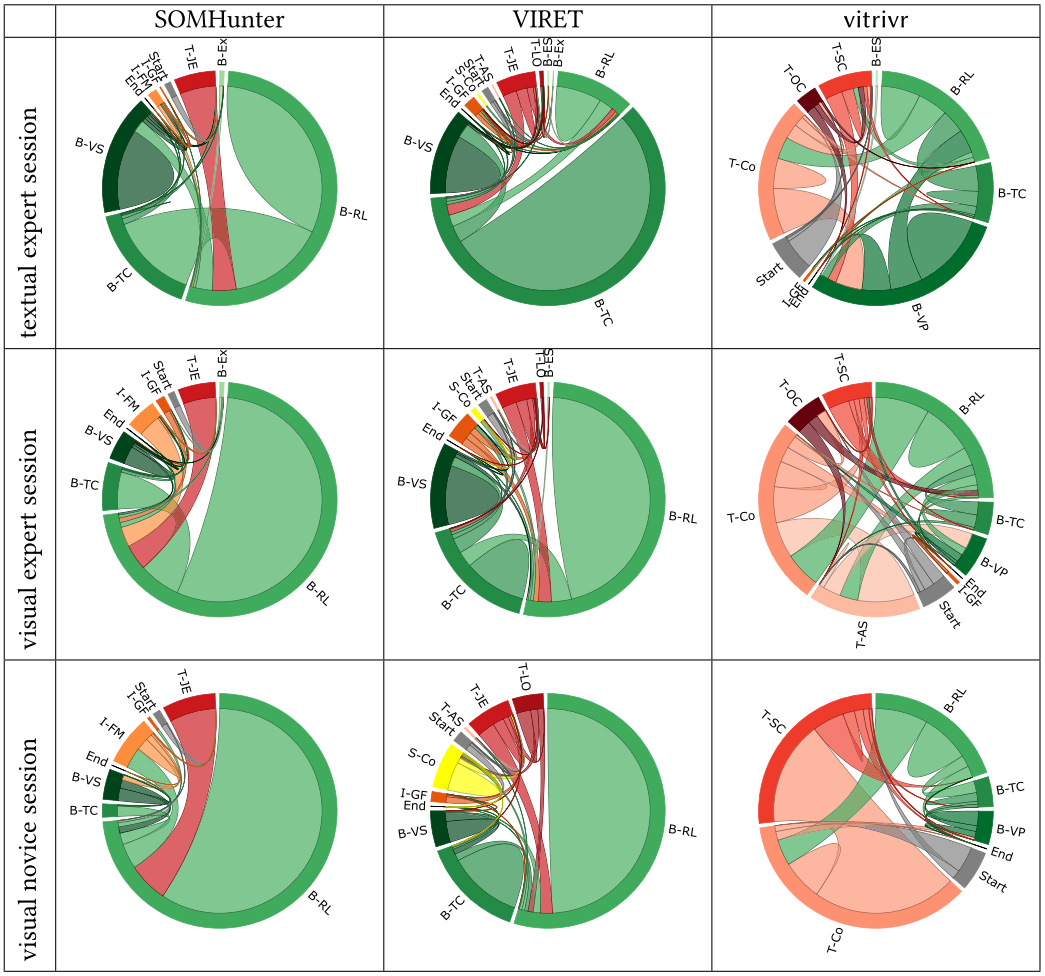
The diagrams reveal used/logged interactions during the competition by the three most successful tools. In general, different types of both querying and browsing actions are used by all three analyzed tools. The higher amount of browsing actions by SOMHunter and VIRET may be caused by the logging methodology employed and frequency of scroll actions, compared to vitrivr, which logs updates of the result cache on the front-end, rather than each wheel action when users scroll. Both SOMHunter and VIRET users often used querying by the W2VV++ text search model (joint-embedding) followed by other tool-specific querying options and browsing. More specifically, the SOMHunter feedback model and k-NN search with global image features were used for querying, while VIRET users relied also on other supported modalities like semantic/color sketches, k-NN search, and speech meta-data provided with the dataset. Logs of both tools report frequent ranked list, temporal context, and video summary inspection. Exploration features were used just rarely by both teams.

While vitrivr offers several sketch-based query options, its users exclusively utilized the text-based query functionality during the competition. Among these, the tag-based queries, where a user selects relevant instances of semantic concepts from a pre-defined set, were the most commonly used, followed by either free-text scene captions or speech transcript search, depending on the task type. The illustrated transitions between these action types indicate that a query commonly consisted of more than one of these modalities, having been fully composed prior to being evaluated by the vitrivr back-end. Please note that the smaller ratio of browsing to query actions might be explained by differing frequencies of log entry generation compared to VIRET and SOMHunter.

5 FUTURE PLANS AND CHALLENGES FOR VBS

Since MMM 2017, the Video Browser Showdown has partly coincided with the conference's welcome reception, making it a highly entertaining spectacle for participants and conference attendees. At MMM 2020, the facilities offered by the conference organizers were magnificent, both in terms of the competition setup and in terms of the ability of conference attendees to follow, and participate in, the competition. Organizing and participating in VBS is a significant undertaking; however, following are some suggestions—in the form of problems and potential solutions—for how to streamline the process and use it to support more scientific progress.

Table 3. Interaction Log Diagrams for Employed Action Types: Start, Browsing: Explicit Sort (B-ES), Browsing: Exploration (B-Ex), Browsing: Ranked List (B-RL), Browsing: Temporal Context (B-TC), Browsing: Video Playback (B-VP), Browsing: Video Summary (B-VS), Sketch: Color (S-Co), Image: Feedback Model (I-FM), Image: Global Features (I-GF), Text: Joint Embedding (T-JE), Text: Localized Object (T-LO), Text: Concept (T-Co), Text: ASR (T-AS), Text: OCR (T-OC), Text: Scene Caption (T-SC), End



Running the competition is a daunting task. Preparing and testing tasks for the competition is a significant undertaking, which is currently done by one or two persons, without access to actual systems for testing tasks. This might be mitigated by appointing judges well before the competition, who do not participate in the competition, and collaborating with them to create and validate tasks. In order to prevent misunderstandings in text KIS tasks, a hand-drawn sketch could be provided with the text. Furthermore, for ad hoc search tasks, judgments made during the validation efforts could then be applied during the competition (even to illustrate positive and negative examples). In contrast to KIS tasks, *Ad hoc Video Search* (AVS) tasks require the teams to submit as many instances for a given topic as possible (e.g., "*Find shots with cars*"). Please note that at VBS

2020 AVS tasks were not included in the final evaluation, due to some technical difficulties with the VBS evaluation server.

Judging segments during AVS tasks is time-consuming. In one of the trial runs of AVS tasks at VBS 2020, more than 2,500 video segments were submitted for judging. Due to this overload of segments, as well as a configuration error on the VBS server, judging took nearly half an hour, even with the collaboration of many participants. And even when all goes well, judging so many segments can be an issue, especially if judging quality is improved by gathering more than one judgment per segment. To solve this issue, several actions can be taken: (1) changing the scoring function to significantly increase the penalty for incorrect submission, thus discouraging hopeful submission of large result sets; (2) allowing conference attendees to participate in judging, especially for tie-breaking purposes; (3) improving the judging software; and (4) when tasks involve a time component, restricting the time interval that the tasks cover.

The barrier to participation remains high. First-time participation, as well as any significant system re-design or re-implementation effort, remains a large challenge, due to the size and complexity of the collection and the tasks to solve. Newcomers sometimes find that due to unforeseen difficulties, often relatively minor, their system can in fact not solve any tasks. In addition to providing template code for result submission and logging, many of these difficulties might be solved by periodically running tasks from a previous VBS competition, prior to the actual current VBS competition taking place. This would be done without any supervision or additional judging, to allow the process to run fully automatically, and could even be done three times daily to suit participants from all continents. Such a process would allow system implementers to test their system using a realistic environment and would also allow them to include some early results in their first VBS paper submissions. Additionally, lightweight VBS systems available as open-source software [28] may help new teams to get started.

The VBS collection is relatively small. Over the years, the video collection used for VBS has grown in steps, to the 1K hours of video currently used. While many KIS tasks remain challenging at this scale, the VBS collection is nevertheless small compared to many of today's collections. As an example, according to one of the MMM 2020 keynote speakers, the VBS collection corresponds to the content newly uploaded to YouTube in about 2.5 minutes. Unfortunately, the size of the collection is part of the participation barrier above, so it would be risky to increase the size of the collection at this point in time. A potential way forward would be to compete in two leagues, using a larger/smaller collection, with KIS tasks defined only from the smaller collection but AVS tasks using both.

The scientific value of the live VBS event is limited. While VBS does offer a venue for comparing systems, the number of data points obtained in small and many factors can contribute to skewing results, such as the experience and attitude of novice users, fit of tasks to the overall approach, and even the form of the expert users—to borrow a phrase from sports—on the day. The VBS approach, however, can be used to generate more data points, by running online evaluations. This could be done in distributed events similar to VBS, where one or two users compete using each system, aimed at comparing the systems and their strategies, keeping the advantages of teams competing simultaneously under the same conditions. These events could be complemented by local events building on the same online infrastructure, where multiple users compete using the same system (or variants of the same system) to provide statistically meaningful user experience data for that particular system. These local events can help teams to gather more experiences, resulting in faster development cycles. Note that the local events could re-use tasks from a previous event, as long as none of the participants has competed in that event, thus re-using competition events multiple times for further data collection. While such distributed evaluation settings have so far not been

possible due to a lack of the required infrastructure, a new evaluation server [53] that will replace the previously used VBS Server starting in 2021 offers all the required functionality.

All these considerations point to a bright future for VBS, both as a live event at upcoming MMM conferences and as a vehicle for significant scientific analysis.

6 CONCLUSIONS

After 9 years of the Video Browser Showdown, a clear winning approach for known-item search scenarios has not yet been identified. The ninth installment hosted many different systems implementing various video retrieval models and browsing approaches, where the initial query specification was usually followed with interactive browsing and query reformulations. Out of 22 evaluated known-item search tasks with expert and novice users, only four systems solved at least half of the tasks during the given time limit. Therefore, we conclude that known-item search still represents a significant challenge for the range of approaches tested at the competition. A significant step forward was the introduction of result set logging, implemented sufficiently by the top three performing systems, SOMHunter, VIRET, and vitrivr. The logs reveal achieved ranks during search sessions and clarify how close a team was to solving a task. With additional query specification available in logs, it is possible to analyze the performance of models (or their combinations) and track transitions between query types. For example, both top-performing systems SOMHunter and VIRET frequently relied on temporal text queries (using W2VV++ model), which represent one of the most common query types before a correct submission of the teams. The interaction log analysis shows that both systems also logged a high volume of browsing interactions. Hence, we conclude that good performance at VBS 2020 was achieved thanks to an effective ranking model, query reformulation based on result set inspection, and support of various result/video browsing features. Still, one of the enduring issues limiting more rapid progress in interactive known-item search is identification of effective models and interactive search strategies. Therefore, we plan to continue our efforts to enforce result/interaction logging and collect more detailed insights of successful searches. We also plan to provide support tools for log analysis and verification as well as simple prototype systems for new teams entering VBS. We believe that all these efforts should help with new advancements in known-item search approaches, where the necessity of interactive search seems to be eternal.

APPENDIX

A TASK DETAILS

Table 4. Preview of All Task Target Sequences

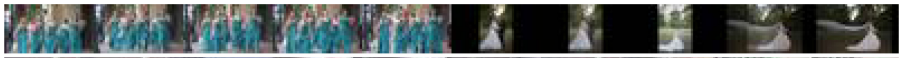
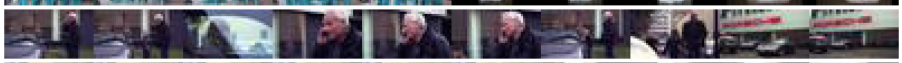
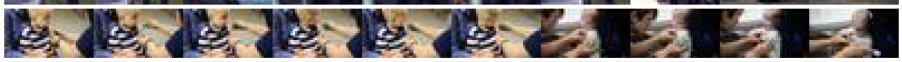
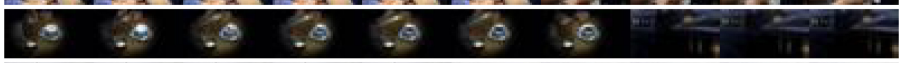
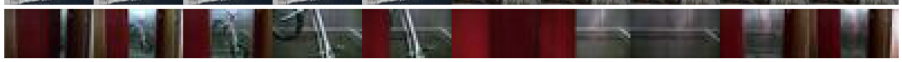
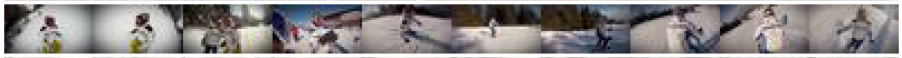
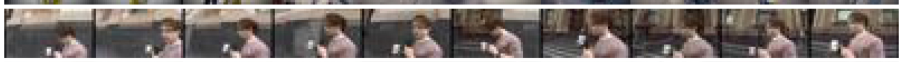
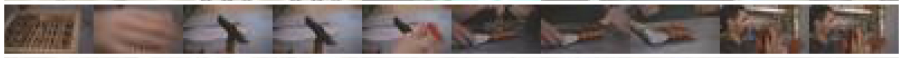
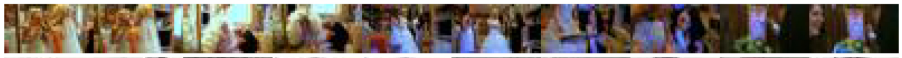
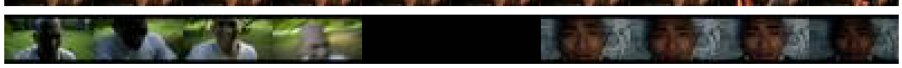
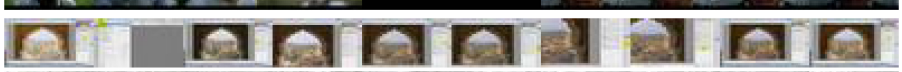
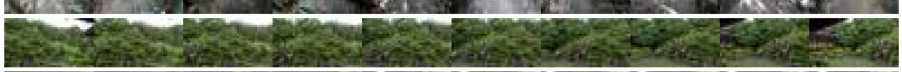
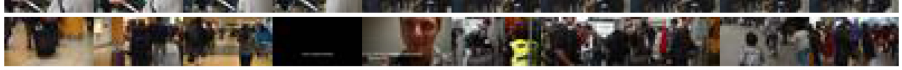
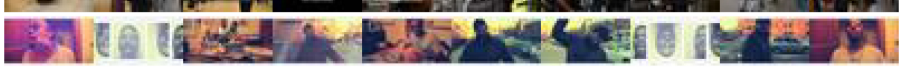
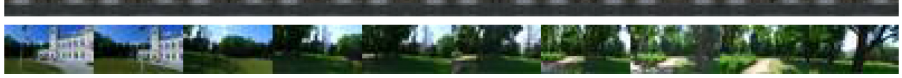
Task	Searched Frame Sequence
T_1	
T_2	
T_3	
T_4	
T_5	
T_6	
V_7	
T_8	
V_9	
T_{10}	
V_{11}	
V_{12}^N	
V_{13}^N	
V_{14}^N	
T_{15}	
V_{16}	
V_{17}	
T_{18}	
V_{19}	
V_{20}^N	
V_{21}^N	
V_{22}^N	

Table 5. Textual KIS Tasks Used in VBS 2020

Task	VBS ID	Query
T_1	Textual2020-16	Seven bridesmaids in turqoise dresses walking down a street, and three still images of the bride and couple. The bridesmaids walk on the sidewalk towards the camera. The photos of the couple and bride are taken in a park.
T_2	Textual2020-17	A man and a young girl walking down the street. In front of a car dealer, the man talks on the phone, the girl looks at a car, they enter the store. The store is a Porsche dealer, a dark gray sports car is parked. The man has short gray hair, both wear dark clothes.
T_3	Textual2020-19	Two shots of nurses measuring the upper arm circumference of kids. They use a green-white tape measure. The woman in the second shot wears a shirt with the text "HOLT International."
T_4	Textual2020-21	Close-up shot of pouring coffee into a white-blue cup, then an outside shot of a house in snow storm. Old-style brass coffee and milk pots. Three story wooden house with white windows.
T_5	Textual2020-22	Shot of a harbor crane in front of mountains, then of three harbor cranes on the left, a ship on the right. Ship is seen from the front right, with text "PROTECT THE ENVIRONMENT" and "NO SMOKING." The harbor cranes are blue and white.
T_6	Textual2020-23	Red elevator doors opening, a bike leans inside, doors closing and reopening, bike is gone. Zoom-in on bike, zoom-out from empty elevator. The bike is silver, the text "ATOMZ" is visible.
T_8	Textual2020-24	A man holding a microphone and coffee cup walks past a building with a dark gray stone wall, and passes the building's dark red door with red fences on the sides. The man has brown hair, wears glasses, and holds a Starbucks cup. The upper part of the building is sandstone, and dark gray stairs lead up to the door.
T_{10}	Textual2020-25	Someone setting letters into a stamp, embossing a name onto an axe cover made of leather, and presenting the axe with the cover. The first shot shows the wooden box with the letters, the last shot a man on the left holding the axe. The man has dark hair and a beard, and wears a sweater with a green/yellow collar.
T_{15}	Textual2020-30	A sequence from a Photoshop tutorial screencast showing how to adjust image settings of a raw image. The image being processed shows a view through an Indian-style window. The image is opened, then lens settings are reviewed and the histogram is adjusted.
T_{18}	Textual2020-31	An African-American musician standing in an NYC subway station and talking to people. He wears a white shirt and a cap, in the second shot one sees he has a drum and a black bag. In the first shot, a sign "EXIT Downtown 6" is visible, in the second "86th" on the wall.

During the competition, the description is being displayed sentence by sentence, at 0, 60, and 120 seconds into the task.

REFERENCES

[1] Stelios Andreadis, Anastasia Moumtzidou, Konstantinos Apostolidis, Konstantinos Gkountakos, Damianos Galanopoulos, Emmanouil Michail, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris. 2020. VERGE in VBS 2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 778–783.

[2] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quénot. 2019. TRECVID 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *TRECVID 2019*. NIST, USA. https://www.nist.gov/publications/trecvid-2019-evaluation-campaign-benchmark-video-activity-detection-video-captioning.

[3] George Awad, Asad Butt, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F. Smeaton, Georges Quénot, Maria Eskevich, Roeland Ordelman, Gareth J. F. Jones, and Benoit Huet. 2017. TRECVID 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *TRECVID 2017*. NIST, USA.

[4] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3c1 dataset: An evaluation of content characteristics. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 334–338.

[5] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. 2018. A short note about kinetics-600. *ArXiv* abs/1808.01340 (2018).

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. Hybrid task cascade for instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 4969–4978.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the 15th European Conference on Computer Vision (ECCV'18)*. Munich, Germany.

[8] Claudiu Cobârzan, Klaus Schoeffmann, Werner Bailer, Wolfgang Hürst, Adam Blažek, Jakub Lokoč, Stefanos Vrochidis, Kai Uwe Barthel, and Luca Rossetto. 2017. Interactive video search tools: A detailed analysis of the video browser showdown 2015. *Multimedia Tools and Applications* 76, 4 (2017), 5539–5571. DOI: https://doi.org/10.1007/s11042-016-3661-2

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. IEEE, Las Vegas, NV, 3213–3223.

[10] Ingemar J. Cox, Matthew L. Miller, Thomas P. Minka, Thomas V. Papathomas, and Peter N. Yianilos. 2000. The bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing* 9, 1 (2000), 20–37.

[11] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. PixelLink: Detecting scene text via instance segmentation. (2018). arxiv:cs.CV/1801.01315

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[13] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9346–9355.

[14] Ling-Yu Duan, Jie Lin, Jie Chen, Tiejun Huang, and Wen Gao. 2014. Compact descriptors for visual search. *IEEE MultiMedia* 21, 3 (2014), 30–40.

[15] Ling-Yu Duan, Yihang Lou, Yan Bai, Tiejun Huang, Wen Gao, Vijay Chandrasekhar, Jie Lin, Shiqi Wang, and Alex Chichung Kot. 2018. Compact descriptors for video analysis: The emerging MPEG standard. *IEEE MultiMedia* 26, 2 (2018), 44–54.

[16] Mark Everingham, S.M. Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)* 111, 1 (2015), 98–136.

[17] Damianos Galanopoulos and Vasileios Mezaris. 2020. Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc videosearch with dual encoding networks. In *Proceedings of the 2020 ACM on International Conference on Multimedia Retrieval (ICMR'20)*. ACM.

[18] Ralph Gasser, Luca Rossetto, and Heiko Schuldt. 2019. Multimodal multimedia retrieval with Vitrivr. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 391–394.

[19] Ilias Gialampoukidis, Anastasia Moumtzidou, Dimitris Liparas, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2016. A hybrid graph-based and non-linear late fusion approach for multimedia retrieval. In *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI'16)*. IEEE, 1–6.

[20] Konstantinos Gkountakos, Anastasios Dimou, Georgios Th Papadopoulos, and Petros Daras. 2019. Incorporating textual similarity in video captioning schemes. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC'19)*. IEEE, 1–6.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[22] Peiyun Hu and Deva Ramanan. 2016. Finding tiny faces. *CoRR* abs/1612.04402 (2016). arxiv:1612.04402

[23] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. 448–456.

[24] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2010), 117–128.

[25] Björn þór Jónsson, Omar Shahbaz Khan, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. 2020. Exquisitor at the video browser showdown 2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 796–802.

[26] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2020. Interactive learning for multimedia at large. In *Proceedings of the European Conference on Information Retrieval (ECIR'20)*. Springer, Lisboa, Portugal, 16.

[27] Miroslav Kratochvíl, Patrik Veselý, František Mejzlík, and Jakub Lokoč. 2020. SOM-Hunter: Video browsing with relevance-to-SOM feedback loop. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 790–795.

[28] Miroslav Kratochvíl, Patrik Veselý, František Mejzlík, and Jakub Lokoč. 2020. SOM-Hunter: Video browsing with relevance-to-SOM feedback loop. In *International Conference on Multimedia Modeling*. Springer, 790–795.

[29] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. 2018. The open images dataset V4. *International Journal of Computer Vision* (2018), 1–26. https://link.springer.com/article/10.1007/s11263-020-01316-z.

[30] Nguyen-Khang Le, Dieu-Hien Nguyen, and Minh-Triet Tran. 2020. An interactive video search platform for multimodal retrieval with advanced concepts. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 766–771.

[31] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. (2018). arxiv:cs.CV/1803.08024

[32] Andreas Leibetseder, Bernd Münzer, Jürgen Primus, Sabrina Kletz, and Klaus Schoeffmann. 2020. diveXplore 4.0: The ITEC deep interactive video exploration system at VBS2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 753–759.

[33] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2VV++: Fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19)*. 1786–1794. DOI: https://doi.org/10.1145/3343031.3350906

[34] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[35] Jakub Lokoč, Tomáš Souček, Patrik Veselý, František Mejzlík, Jiaqi Ji, Chaoxi Xu, and Xirong Li. 2020. A W2VV++ case study with automated and interactive text-to-video retrieval. In *Proceedings of the 28th ACM International Conference on Multimedia (MM'20)*. Association for Computing Machinery, New York, NY.

[36] Jakub Lokoč, Werner Bailer, Klaus Schoeffmann, Bernd Münzer, and George Awad. 2018. On influential trends in interactive video retrieval: Video browser showdown 2015-2017. *IEEE Transactions on Multimedia* 20, 12 (2018), 3361–3376.

[37] Jakub Lokoč, Gregor Kovalčík, Bernd Münzer, Klaus Schöffmann, Werner Bailer, Ralph Gasser, Stefanos Vrochidis, Phuong Anh Nguyen, Sitapa Rujikietgumjorn, and Kai Uwe Barthel. 2019. Interactive search or sequential browsing? A detailed analysis of the video browser showdown 2018. *ACM Transactions on Multimedia Computing, Communications, and Applications* 15, 1, Article 29 (Feb. 2019), 18 pages. DOI: https://doi.org/10.1145/3295663

[38] Jakub Lokoč, Gregor Kovalčík, and Tomáš Souček. 2020. VIRET at video browser showdown 2020. In *MultiMedia Modeling - 26th International Conference (MMM'20), Proceedings, Part II (Lecture Notes in Computer Science)*, Yong

Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.), Vol. 11962. Springer, 784–789. DOI: https://doi.org/10.1007/978-3-030-37734-2_70

[39] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. 2019. A framework for effective known-item search in video. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19)*. ACM, New York, NY, 1777–1785. DOI: https://doi.org/10.1145/3343031.3351046

[40] Jakub Lokoč, Gregor Kovalčík, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. 2019. VIRET: A video retrieval tool for interactive known-item search. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval (ICMR'19)*. ACM, New York, NY, 177–181. DOI: https://doi.org/10.1145/3323873.3325034

[41] Bangalore S. Manjunath, Philippe Salembier, and Thomas Sikora. 2002. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons.

[42] Foteini Markatopoulou, Vasileios Mezaris, and Ioannis Patras. 2018. Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. *IEEE Transactions on Circuits and Systems for Video Technology* (2018). https://ieeexplore.ieee.org/document/8387768.

[43] Pascal Mettes, Dennis C. Koelma, and Cees G.M. Snoek. 2016. The Imagenet shuffle: Reorganized pre-training for video event detection. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR'16)*. ACM, New York, NY, 175–182. DOI: https://doi.org/10.1145/2911996.2912036

[44] Pascal Mettes, Dennis C. Koelma, and Cees G.M. Snoek. 2020. Shuffled Imagenet banks for video event detection and search. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 2 (2020), 1–21.

[45] Phuong Anh Nguyen, Yi-Jie Lu, Hao Zhang, and Chong-Wah Ngo. 2018. Enhanced VIREO KIS at VBS 2018. In *MultiMedia Modeling*. 407–412.

[46] Phuong Anh Nguyen, Jiaxin Wu, Chong-Wah Ngo, Francis Danny, and Huet Benoit. 2019. VIREO-EURECOM @ TRECVID 2019: Ad-hoc video search. In *NIST TRECVID Workshop*.

[47] Phuong Anh Nguyen, Jiaxin Wu, Chong-Wah Ngo, Danny Francis, and Benoit Huet. 2020. VIREO @ Video browser showdown 2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 772–777.

[48] Sungjune Park, Jaeyub Song, Minho Park, and Yong Man Ro. 2020. IVIST: Interactive video search tool in VBS 2020. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 809–814.

[49] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3D residual networks. *CoRR* abs/1711.10305 (2017). arxiv:1711.10305

[50] Joseph Redmon and Ali Farhadi. 2018. YOLO v3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[51] Luca Rossetto, Werner Bailer, and Abraham Bernstein. 2021. Considering human perception and memory in interactive multimedia retrieval evaluations. In *Proceedings of the 27th International Conference on MultiMedia Modeling*.

[52] Luca Rossetto, Ralph Gasser, Jakub Lokoč, Werner Bailer, Klaus Schoeffmann, Bernd Muenzer, Tomáš Souček, Phuong Anh Nguyen, Paolo Bolettieri, Andreas Leibetseder, and Stefanos Vrochidis. 2021. Interactive video retrieval in the age of deep learning - Detailed evaluation of VBS 2019. *IEEE Transactions on Multimedia* 23 (2021), 243–256. DOI: https://doi.org/10.1109/TMM.2020.2980944

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

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

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

[56] Luca Rossetto, Mahnaz Amiri Parian, Ralph Gasser, Ivan Giangreco, Silvan Heller, and Heiko Schuldt. 2019. Deep learning-based concept detection in vitrivr. In *MultiMedia Modeling - 25th International Conference (MMM'19), Proceedings, Part II*. 616–621. DOI: https://doi.org/10.1007/978-3-030-05716-9_55

[57] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A. Butt. 2019. V3C - A research video collection. In *MultiMedia Modeling - 25th International Conference (MMM'19), Proceedings, Part I*. 349–360. DOI: https://doi.org/10.1007/978-3-030-05710-7_29

[58] Loris Sauter, Mahnaz Amiri Parian, Ralph Gasser, Silvan Heller, Luca Rossetto, and Heiko Schuldt. 2020. Combining Boolean and multimedia retrieval in vitrivr for large-scale video search. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 760–765.

[59] Klaus Schoeffmann. 2019. Video browser showdown 2012-2019: A review. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI'19)*. 1–4. DOI: https://doi.org/10.1109/CBMI.2019.8877397

[60] Klaus Schoeffmann, Bernd Münzer, Andreas Leibetseder, Jürgen Primus, and Sabrina Kletz. 2019. Autopiloting feature maps: The deep interactive video exploration (diveXplore) System at VBS2019. In *MultiMedia Modeling*. Springer International Publishing, Cham, 585–590.

[61] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2019. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 9 (2019), 2035–2048.

[62] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *9th International Conference on Document Analysis and Recognition (ICDAR'07)*, Vol. 2. IEEE, 629–633.

[63] Tomáš Souček, Jaroslav Moravec, and Jakub Lokoč. 2019. TransNet: A deep network for fast detection of common shot transitions. *CoRR* abs/1906.03363 (2019). arxiv:1906.03363

[64] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2017. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 652–663.

[65] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE Computer Society, Honolulu, HI, 5987–5995.

[66] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10334–10343.

[67] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

[68] Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. 2020. Tree-augmented cross-modal encoding for complex-query video retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1339–1348.

[69] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. 2019. Deep supervised cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10394–10403.

[70] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. In *Proceedings of the International Conference on Neural Information*. 487–495.

[71] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, Vol. 1. IEEE, Honolulu, HI, 4.

[72] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: An efficient and accurate scene text detector. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 2642–2651. DOI: https://doi.org/10.1109/CVPR.2017.283

Received April 2020; revised October 2020; accepted December 2020