Combining Boolean and Multimedia Retrieval in vitrivr for Large-Scale Video Search

Loris Sauter[1][0000−0001−8046−0362], Mahnaz Amiri Parian[1,3][0000−0001−7063−8585], Ralph Gasser[1][0000−0002−3016−1396], Silvan Heller[1][0000−0001−5386−330X], Luca Rossetto[1,2][0000−0002−5389−9465], and Heiko Schuldt[1][0000−0001−9865−6371]

[1] Department of Mathematics and Computer Science
University of Basel, Basel, Switzerland
{firstname.lastname}@unibas.ch

[2] Department of Informatics, University of Zurich, Zurich, Switzerland
rossetto@ifi.uzh.ch

[3] Numediart Institute, University of Mons, Mons, Belgium

Abstract. This paper presents the most recent additions to the vitrivr multimedia retrieval stack made in preparation for the participation to the 9th Video Browser Showdown (VBS) in 2020. In addition to refining existing functionality and adding support for classical Boolean queries and metadata filters, we also completely replaced our storage engine ADAM$_{pro}$ by a new database called *Cottontail DB*. Furthermore, we have added support for scoring based on the temporal ordering of multiple video segments with respect to a query formulated by the user. Finally, we have also added a new object detection module based on Faster-RCNN and use the generated features for object instance search.

Keywords: Video Browser Showdown · Interactive Video Retrieval

1 Introduction

In this paper, we present the recent improvements made to vitrivr [18], our multimedia retrieval stack capable of processing several different types of media documents [3]. vitrivr (and its predecessor, the IMOTION system [16]) has participated in the Video Browser Showdown (VBS) [9] for several years [17] and recently also made its debut [13] at the Lifelog Search Challenge (LSC) 2019 [6]. Throughout its development history, vitrivr has gained a large amount of content-based retrieval related functionality. Some of these capabilities however have been discontinued due to the replacement or re-implementation of certain components of the stack. Other capabilities have become impractical in a competitive retrieval context due to changing circumstances, such as the introduction of larger datasets, most recently the V3C [21], the first shard of which already contains 1000 hours of video [1]. Our primary focus for this year's participation to the VBS is to consolidate several recent changes made to vitrivr as well as to re-introduce some of these past capabilities, as outlined in Section 3.3.

The remainder of this paper is structured as follows: Section 2 provides an overview of the vitrivr stack and its primary components and illustrates its current capabilities. Section 3 then goes into some detail regarding the newly added as well as the re-introduced functionality with which vitrivr will participate to this iteration of the competition. Finally, Section 4 concludes this paper.

2 System Overview and Existing Capabilities

vitrivr is a content-based multimedia retrieval stack that is able to retrieve results from mixed media collections [3] containing images, audio, 3D data, and video – of which only the latter is relevant to the VBS competition. vitrivr enables a multitude of query modes, such as Query-by-Sketch (QbS) with both visual and semantic representations, Query-by-Example (QbE) using external example documents from all supported media domains, textual and Boolean queries using structured metadata as well as any combination of the above.

For visual QbS and QbE, vitrivr uses several low-level image features, the combination of which is configurable by the user. The semantic sketch capabilities are realized using a DNN pixel-wise semantic annotator as described in [14]. The textual features encompass OCR and ASR data extracted from the videos as well as automatically generated scene-wise descriptions. The structured metadata contains the data that is part of V3C1 itself along with several object annotations generated by various semantic annotators or obtained from publicly available sources [12].

The vitrivr stack is comprised of three primary components: the persistence layer, a retrieval engine called *Cineast* and a browser-based user interface called *Vitrivr NG*. The persistence layer, for which up and until now we have used our own database system ADAM_{pro} [5], manages all the data required for retrieval, i.e., feature vectors, IDs, attributes, and facilitates query execution. The retrieval engine Cineast generates features from the raw input, such as images, sketches or text, orchestrates the execution of different queries through the persistence layer, fuses results and communicates with the user interface. The user interface Vitrivr NG offers different modes of result presentation and provides all the tools required for query formulation. Furthermore, it employs a secondary late-fusion step that gives the end-user some control over how partial results should be merged.

The entire vitrivr stack and its components are open source[4] and publicly available from GitHub[5].

3 New Functionality in vitrivr

This section provides an overview of improvements compared to the system which won the 2019 iteration of the Video Browser Showdown [19]. One major

[4] https://vitrivr.org/

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

focus was the introduction of a new storage layer to address performance issues discussed in [20] which will be summarized in Section 3.5. Other major additions are based on our winning participation at the Lifelog Search Challenge [13], where metadata and Boolean retrieval was important given the content of the dataset [6].

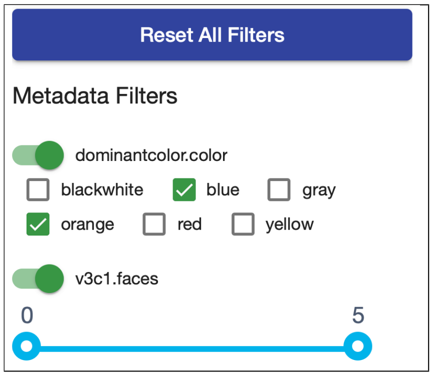
3.1 Boolean Queries

As discussed in [4], unifying the traditional Boolean retrieval model used in structured data with the world of multimedia retrieval remains an open challenge. To briefly summarize our approach introduced in [13], Boolean feature modules return all matching elements. In the result-fusion step, these results are then used as a filter in contrast to results from similarity modules, where non-matching segments simply do not contribute to the score.

Another design option would have been to do early filtering, only considering elements for similarity queries which match the Boolean filter criteria. Our reasoning to go for late filtering is discussed in [13].

3.2 Metadata Filters

In addition to the Boolean queries discussed in Section 3.1, vitrivr offers the possibility to refine results in the user interface. To improve performance and responsiveness, filtering is only done on the client's side, meaning no new queries are executed. The available filters are based on query results and a whitelist of filter keys. This is especially sensible for novices since it hides metadata which is not useful for filtering. This is in line with the focus on usability which has served vitrivr well in competitive settings. Figure 1 shows an example of the metadata filter UI in action.



dominantcolor.color

☐ blackwhite ☑ blue ☐ gray

☑ orange ☐ red ☐ yellow

v3c1.faces

0 5

Fig. 1. Metadata filters with both checkboxes and slider values. Results are dynamically updated on changing filters.

3.3 Temporal Scoring

Inspired by the successful application of temporal queries and scoring by other teams [10], we re-introduced this once available feature [15, 16] into vitrivr. Temporal queries enable the user to specify multiple queries in a given order, which is then used as a relative temporal reference. For scoring the results, we introduce an expansion to our media description model with the notion of *temporal closeness*. Two segments s_i and s_j are *temporally close*, if $s_j.startTime - s_i.endTime < \varepsilon$, where ε is a configurable threshold. Such temporally close segments s_i and s_j are merged to a single *temporal segment*, if the first segment matches the first query condition and the second segment matches the second query condition. This rule can be applied recursively to other segments as well. The configurable scoring function then assures that temporally close segments occurring in the specified order are boosted.

The user interface reflects the re-introduction of temporal scoring by enabling users to re-order their queries as well as by adding adequate visualization of temporal closeness.

3.4 Object Instance Search

To enhance the feature extraction module of vitrivr, we use the idea proposed in [2] to incorporate the feature embedding extracted by an object detection module in a retrieval task. More specifically, the feature embedding acts as a hard attention module and is used to assign scores to the parts of the feature maps which represent these objects.

To perform the object instance search, two steps are taken: First, the Faster-RCNN [11] framework – pre-trained on the Openimages V4 dataset [8] – is used to extract the regions of interests (ROIs). These ROIs are bounded by boxes and localize objects in the keyframe of the video clip. Second, the ROIs are fed to the feature extraction module as hard attention. The feature extraction is performed by ResNet-50 [7] pre-trained on ImageNet for the classification task. The convolutional features prior to the fully connected layer of ResNet-50 are extracted and used for similarity search. The attended features from ResNet-50 increase the relevance of the search results based on the existing objects in the video clips which eventually enhances the performance of the retrieval task.

3.5 Storage Layer

The vitrivr stack generates and operates on a variety of different types of data ranging from primitive data types to feature vectors. In the past, Cineast has delegated persistent storage as well as lookup to an underlying storage engine called ADAM_{pro}.

In preparation for LSC 2019, we have completely replaced that storage engine by a new system called *Cottontail DB*. This step was necessitated by performance considerations. Even though ADAM_{pro} was designed with scalability and distribution in mind, it always under-performed in a single-node setup and on workloads typically found during competitions such as LSC or VBS. The high query times, especially for similarity based queries, severely limited our choices time-critical settings.

Cottontail is a columnar storage engine. Hence, data can be accessed very efficiently if entire columns are read, e.g., for full scans of a particular attribute. *Cottontail* allows to organize such columns into entities, which in turn can be organized into different schemas. Hence, the data model is very similar to the one found in a classical relational database and ADAM_{pro}.

In addition to Boolean queries and full text search (powered by Apache Lucene), *Cottontail* offers support for k nearest neighbour (kNN) lookup, typically used for feature-based similarity search. In that area, it outperforms the ADAM_{pro} system by at least an order of magnitude without relying on any index structures. However, secondary indexes are supported and the addition of index structures for kNN lookups is planned for future versions.

4 Conclusion

In this paper, we presented the additions made to vitrivr for VBS 2020. We expect the transition to the new storage engine to provide us with better performance and thus with more flexibility as to what types of queries we can use in a competitive setting. Additionally, we see VBS as the final test before *Cottontail* can be published and released as the new, official storage subsystem that powers the vitrivr stack.

Acknowledgements

This work was partly supported by the Hasler Foundation in the context of the project City-Stories (contract no. 17055).

References

1. Berns, F., Rossetto, L., Schoeffmann, K., Beecks, C., Awad, G.: V3c1 dataset: An evaluation of content characteristics. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 334–338. ACM (2019)
2. Chen, B.C., Davis, L.S., Lim, S.N.: An analysis of object embeddings for image retrieval. arXiv preprint arXiv:1905.11903 (2019)
3. Gasser, R., Rossetto, L., Schuldt, H.: Towards an all-purpose content-based multimedia information retrieval system. arXiv preprint arXiv:1902.03878 (2019)
4. Giangreco, I.: Database support for large-scale multimedia retrieval. Ph.D. thesis, University of Basel (2018)
5. Giangreco, I., Schuldt, H.: ADAM_{pro}: Database Support for Big Multimedia Retrieval. Datenbank-Spektrum **16**(1), 17–26 (2016)
6. Gurrin, C., Schoeffmann, K., Joho, H., Munzer, B., Albatal, R., Hopfgartner, F., Zhou, L., Dang-Nguyen, D.T.: A test collection for interactive lifelog retrieval. In: International Conference on Multimedia Modeling. pp. 312–324. Springer (2019)

7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
8. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Kamali, S., Malloci, M., Pont-Tuset, J., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., Murphy, K.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://storage.googleapis.com/openimages/web/index.html (2017)
9. Lokoč, J., Kovalčík, G., Münzer, B., Schöffmann, K., Bailer, W., Gasser, R., Vrochidis, S., Nguyen, P.A., Rujikietgumjorn, S., Barthel, K.U.: Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **15**(1), 29 (2019)
10. Lokoč, J., Kovalčík, G., Souček, T., Moravec, J., Čech, P.: Viret: A video retrieval tool for interactive known-item search. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 177–181. ACM (2019)
11. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR **abs/1506.01497** (2015), http://arxiv.org/abs/1506.01497
12. Rossetto, L., Berns, F., Schoeffman, K., Awad, G., Beeks, C.: The v3c1 dataset: Advancing the state of the art in video retrieval. ACM SIGMultimedia Records **11**(2) (2019)
13. Rossetto, L., Gasser, R., Heller, S., Amiri Parian, M., Schuldt, H.: Retrieval of structured and unstructured data with vitrivr. In: Proceedings of the ACM Workshop on Lifelog Search Challenge. pp. 27–31. ACM (2019)
14. Rossetto, L., Gasser, R., Schuldt, H.: Query by semantic sketch. CoRR **abs/1909.12526** (2019), https://arxiv.org/abs/1909.12526
15. Rossetto, L., Giangreco, I., Heller, S., Tănase, C., Schuldt, H.: Searching in video collections using sketches and sample images–the cineast system. In: International Conference on Multimedia Modeling. pp. 336–341. Springer (2016)
16. Rossetto, L., Giangreco, I., Heller, S., Tănase, C., Schuldt, H., Dupont, S., Seddati, O., Sezgin, M., Altıok, O.C., Sahillioğlu, Y.: Imotion–searching for video sequences using multi-shot sketch queries. In: International Conference on Multimedia Modeling. pp. 377–382. Springer (2016)
17. Rossetto, L., Giangreco, I., Schuldt, H., Dupont, S., Seddati, O., Sezgin, M., Sahillioğlu, Y.: Imotiona content-based video retrieval engine. In: International Conference on Multimedia Modeling. pp. 255–260. Springer (2015)
18. Rossetto, L., Giangreco, I., Tanase, C., Schuldt, H.: Vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 1183–1186. ACM (2016)
19. Rossetto, L., Parian, M.A., Gasser, R., Giangreco, I., Heller, S., Schuldt, H.: Deep learning-based concept detection in vitrivr. In: International Conference on Multimedia Modeling. pp. 616–621. Springer (2019)
20. Rossetto, L., Parian, M.A., Gasser, R., Giangreco, I., Heller, S., Schuldt, H.: Deep learning-based concept detection in vitrivr at the video browser showdown 2019-final notes. arXiv preprint arXiv:1902.10647 (2019)
21. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C–A Research Video Collection. In: International Conference on Multimedia Modeling. pp. 349–360. Springer (2019)