VIREO @ Video Browser Showdown 2020

Phuong Anh Nguyen[1(✉)], Jiaxin Wu[1], Chong-Wah Ngo[1], Danny Francis[2], and Benoit Huet[2]

[1] Computer Science Department, City University of Hong Kong, Hong Kong, China
{panguyen2-c,jiaxin.wu}@my.cityu.edu.hk, cscwngo@cityu.edu.hk
[2] Data Science Department, EURECOM, Biot, France
{danny.francis,benoit.huet}@eurecom.fr

Abstract. In this paper, we present the features implemented in the 4th version of the VIREO Video Search System (VIREO-VSS). In this version, we propose a sketch-based retrieval model, which allows the user to specify a video scene with objects and their basic properties, including color, size, and location. We further utilize the temporal relation between video frames to strengthen this retrieval model. For text-based retrieval module, we supply speech and on-screen text for free-text search and upgrade the concept bank for concept search. The search interface is also re-designed targeting the novice user. With the introduced system, we expect that the VIREO-VSS can be a competitive participant in the Video Browser Showdown (VBS) 2020.

Keywords: Sketch-based retrieval · Query-by-object-sketch · Video retrieval · Video Browser Showdown

1 Introduction

The VIREO-VSS has participated in the Video Browser Showdown [1] for three consecutive years since 2017. The latest version of the VIREO-VSS [2] is composed of three retrieval modules: query-by-text, query-by-sketch, and query-by-example. Given a query, a user firstly makes an input into one of these search modules. Then the retrieved ranked list is represented in a user interface, which allows the user to browse and judge the result. This process is repeated in a loop until the user finds a satisfactory answer.

In VBS 2019, the VIREO-VSS ranked 3th over 6 teams participated despite the system flaws exposed in the benchmark. The system has proved its capability for the new large-scale video dataset V3C1 and its performance in the visual known-item search and the ad-hoc search tasks. The retrieval modality contributions are as follows:

- The simplified query-by-color-sketch model proposed in [3] plays the leading role in solving the visual known-item search task.
- The query-by-concept model [3], together with the nearest neighbor search and quick submission [2], enhance the efficiency of submission in the ad-hoc search task.

W.-H. Cheng et al. (Eds.): MMM 2020, LNCS 11962, pp. 772–777, 2020.
https://doi.org/10.1007/978-3-030-37734-2_68

However, the system still can not solve the textual known-item search task efficiently. The reason is that we did not utilize the temporal relation between video segments, which leads to inefficient retrieval result with the correct answer stays pretty low in the rank list. As a consequence, the user needs to browse tediously to find the correct answer. Besides, the user interface is not friendly to the novice user with various functions and configurations.

Facing different shortcomings of the system, we propose the 4^{th} version of the VIREO-VSS with three revisions. First, the color-sketch-based retrieval models from the SIRET [4], VIREO [3]; the object-sketch-based retrieval model from the VISIONE [5]; and the semantic-sketch-based retrieval model from the vitrivr [6] are critical approaches to solve the visual KIS task. Motivated by these approaches, we initiate the idea of utilizing the object color for object-sketch-based retrieval targeting both visual and textual KIS task. As learned from VBS 2019, the temporal information is the deciding factor to solve the textual KIS confirmed by the result of VIRET team [7]. Hence, we utilize the temporal information for both sketch-based and concept-based retrieval model. Second, we supply more data for text-based search including the video speech, the on-screen text, and 1200 additional concepts. Third, the simple and friendly user interface from vitrivr [6] has proved its advantages in the novice session of VBS 2019. Therefore, we update the user interface to make it more compact and user-friendly. The setup details of these revisions are described in the next section.

2 Object-Sketch-Based Retrieval Model

2.1 Feature Extraction

Given a set of video key-frames, we use the Faster-RCNN architecture [8] with Inception ResNetV2 feature extractor [9] pre-trained on Open Images V4. The network can detect up to 600 objects together with the bounding boxes after performing non-maxima suppression. For each key-frame, we keep the objects that have the detection score larger than 0.1 and their bounding boxes. To extract the color attribute, we first define a list of 15 colors, including 12 primary colors from the RGB color wheel and three neutral colors (black, grey, and white). From each bounding box, we map each pixel color to the nearest color in the list; then we calculate the percentage of each color. In the end, from an image O extracted from the video, we have a set of tuples $\{\langle n_i^o, s_i^o, B_i^o, C_i^o \rangle\}_{i=1}^n$ with each tuple representing an object o_i. Each object o_i contains an object name n_i^o, a detection score s_i^o, a bounding box B_i^o, and a 15 dimensions color feature C_i^o.

2.2 Retrieval Model

The user can draw their memo using a canvas where he determines rectangles depicting objects. For each box created by clicking on a clear area of the canvas, the user can add an object with its name and its color by clicking on the rectangle's top and bottom bar. It is noted that the user can leave the object's

name and color empty. The user also can drag and resize the box to specify the location and size of the object. An example of the query canvas and the object detection result can be seen in Fig. 1.

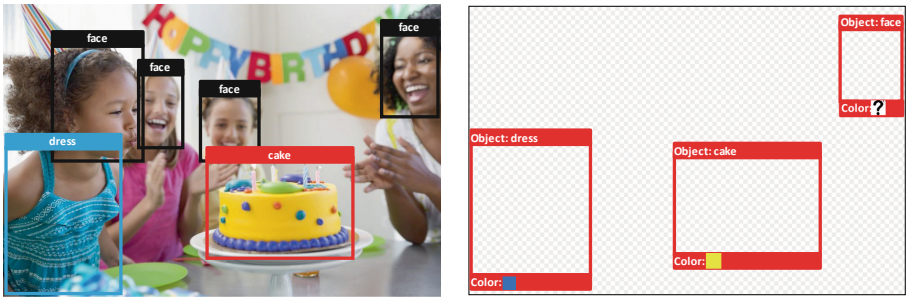

Fig. 1. An example of the object detection result (left) and a possible query to retrieve the image (right). Note that the user can choose a question mark option to specify the color, which represents no specific color. (Color figure online)

Formally, the user's query Q is a set of tuples $\{\langle n_j^q, B_j^q, c_j^q \rangle\}_{j=1}^m$ with each tuple representing an object query q_j. The scoring function to match between a query object q_j to a detected object o_i in the detected is defined as:

$$S(q_j, o_i) = \alpha * match\_obj(n_j^q, n_i^o) + \beta * match\_clr(c_j^q, C_i^o) + \gamma * IoU(B_j^q, B_i^o)$$

with $match\_obj(n_j^q, n_i^o) = s_i^o$ if $n_j^q = n_i^o$, and 0 otherwise; $match\_clr(c_j^q, C_i^o)$ returns the percentage of the color c_j^q from the color feature C_i^o; $IoU(B_j^q, B_i^o)$ return the intersection over union of two bounding boxes; α, β, and γ are the weights of the properties. In this trial, we intuitively select $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$.

Then, the similarity of a query Q and a key-frame O in the dataset is defined as:

$$Sim(Q, O) = \underset{\forall q_j \in Q}{\text{avg}} \left(\max_{\forall o_i \in O} S(q_j, o_i) \right)$$

We calculate the similarities from the query to all key-frames in the dataset to get the rank list of key-frames. In addition, we perform $MinMax$ normalization to map these similarities to the range $[0, 1]$ to enable fusing with other search modalities.

2.3 Temporal Query for Object-Sketch-Based Retrieval

We provide two canvases enabling the user to input two object-sketch queries at the timestamp t and t' with $t < t'$. We follow these steps to define the similarity of a video $V = \{O_1, \ldots, O_k\}$ and two query Q_t, $Q_{t'}$:

1. Calculate two sets of similarities $\{Sim(Q_t, O_1), \ldots, Sim(Q_t, O_k)\}$ and $\{Sim(Q_{t'}, O_1), \ldots, Sim(Q_{t'}, O_k)\}$
2. Construct an array $MaxSim(Q_t, V) = [s_1^t, \ldots, s_k^t]$ with $s_1^t = Sim(Q_t, O_1)$ and $s_i^t = max(s_{i-1}^t, Sim(Q_t, O_i))$ for $i > 1$; the order from 1 to k representing the temporal order of the key-frames. This is an increasing array. An element s_i^t of this array represents the maximum similarity of the query Q_t to the video segment starting from the first key-frame to the key-frame i.
3. Construct an array $MaxSim(Q_{t'}, V) = [s_1^{t'}, \ldots, s_k^{t'}]$ with $s_k^{t'} = Sim(Q_{t'}, O_k)$ and $s_i^{t'} = max(s_{i+1}^{t'}, Sim(Q_{t'}, O_i))$ for $i < k$; the order from 1 to k representing the temporal order of the key-frames. This is a decreasing array. An element $s_i^{t'}$ of this array represents the maximum similarity of the query $Q_{t'}$ to the video segment starting from the key-frame i to the last key-frame.
4. Calculate the $Sim((Q_t, Q_{t'}), V) = \max_{i=1}^{k-1}(s_i^t + s_{i+1}^{t'})$

Each step described above has the complexity as $O(k)$ with k is the number of key-frames in a video. To process the whole video dataset, this approach has the complexity as $O(l)$ with l is the total number of key-frames. With this approach, the user needs to remember the order of the key-frames that ensure $t < t'$ rather than the exact interval between two queries.

3 Video Retrieval Tool Description

At first, we integrate the query-by-object-sketch module proposed in Sect. 2 to the VIREO-VSS. Next, we focus on improving the tool at three points:

- *Query-by-text.* For free text search, we add in the video speech shared by vitrivr and the on-screen text detected by Tesseract-OCR. For concept-based search, we update the concept bank with an addition of 600 object detectors trained on Open Images V4 using FasterRCNN and 600 activity detectors trained on Kinetics using I3D [10]. In the latest version of the VIREO-VSS, the query-by-concept module requires the user to initiate the query from scratch with exact concept selection. As our observation in the novice session of VBS 2019, this constraint creates difficulties for the user who has no prior understanding of the concept bank. To overcome this shortcoming, we let the user input free-text and kick off the concept selection step by automatically picking related concepts. This step is done by using Universal Sentence Embedding to match between the free-text query and the concept definitions. Moreover, the temporal query model presented in Sect. 2.3 is employed for concept-based search targeting the textual KIS task.
- *The rank list for AVS task.* From VBS 2018, the scoring function of AVS task favors the diversity than the quantity of submitted video segments. To compromise with this scoring function, we push the video segments that come from different videos up to the top of the rank list. This rank list allows the user to look at the video segments coming from different videos and leads him to submit these video segments, which are in favor of diversity.

- *The user interface.* As we enable free text search for concept-based retrieval, we provide only one text box for text-based search. The user can select the sources of searching by checking in the corresponding checkboxes. The temporal queries using in concept-based and object-sketch-based retrieval module are presented under different tabs (see Fig. 2).

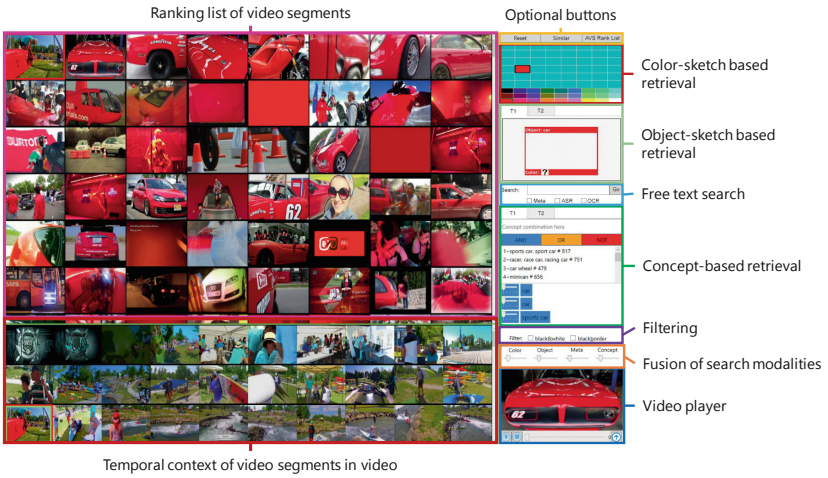

Fig. 2. The user interface of the VIREO-VSS.

We maintain existing modules without further modification, including: the query by color-sketch model, the query-by-example model, and the filtering functions. The detail setup of these models and functions can be found in [2].

4 Conclusion

In this 4th version of the VIREO-VSS, we focus on solving textual KIS task and improving the submission efficiency for AVS task. The proposed query-by-object-sketch model supports both visual and textual KIS task. The temporal query for object-sketch-based retrieval and concept-based retrieval concretely favors the textual KIS task. The modification on the user interface and the rank list of AVS result enhances the submission speed and the diversity of the AVS answer. With these improvements, we expect that the VIREO-VSS can tackle all types of tasks in VBS 2020 efficiently.

Acknowledgments. The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Reference No.: CityU 11250716), and a grant from the PROCORE-France/Hong Kong Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the Consulate General of France in Hong Kong (Reference No.: F-CityU104/17).

References

1. Lokoč, J., et al.: Interactive search or sequential browsing? A detailed analysis of the video browser showdown 2018. ACM Trans. Multimedia Comput. Commun. Appl. **15**(1), 29:1–29:18 (2019)
2. Nguyen, P.A., Ngo, C.-W., Francis, D., Huet, B.: VIREO @ video browser showdown 2019. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W.-H., Vrochidis, S. (eds.) MMM 2019. LNCS, vol. 11296, pp. 609–615. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05716-9_54
3. Nguyen, P.A., Lu, Y.-J., Zhang, H., Ngo, C.-W.: Enhanced VIREO KIS at VBS 2018. In: Schoeffmann, K., et al. (eds.) MMM 2018. LNCS, vol. 10705, pp. 407–412. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73600-6_42
4. Lokoč, J., Kovalčík, G., Souček, T.: Revisiting SIRET video retrieval tool. In: Schoeffmann, K., et al. (eds.) MMM 2018. LNCS, vol. 10705, pp. 419–424. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73600-6_44
5. Amato, G., et al.: VISIONE at VBS2019. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W.-H., Vrochidis, S. (eds.) MMM 2019. LNCS, vol. 11296, pp. 591–596. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05716-9_51
6. Rossetto, L., Amiri Parian, M., Gasser, R., Giangreco, I., Heller, S., Schuldt, H.: Deep learning-based concept detection in vitrivr. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W.-H., Vrochidis, S. (eds.) MMM 2019. LNCS, vol. 11296, pp. 616–621. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05716-9_55
7. Lokoč, J., Kovalčík, G., Souček, T., Moravec, J., Bodnár, J., Čech, P.: VIRET tool meets NasNet. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W.-H., Vrochidis, S. (eds.) MMM 2019. LNCS, vol. 11296, pp. 597–601. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05716-9_52
8. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137–1149 (2015)
9. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI 2017, pp. 4278–4284. AAAI Press, San Francisco (2017)
10. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733 (2017)