Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

VERGE IN VBS 2018

Anastasia Moumtzidou[1], Stelios Andreadis[1] Foteini Markatopoulou[1,2], Damianos Galanopoulos[1], Ilias Gialampoukidis[1], Stefanos Vrochidis[1], Vasileios Mezaris[1], Ioannis Kompatsiaris[1], and Ioannis Patras[2]

[1] Information Technologies Institute/Centre for Research & Technology Hellas, Thessaloniki, Greece
{moumtzid, andreadisst, markatopoulou, dgalanop, heliasgj, stefanos, bmezaris, ikom}@iti.gr
[2] School of Electronic Engineering and Computer Science, QMUL, UK
i.patras@qmul.ac.uk

Abstract. This paper presents VERGE interactive video retrieval engine, which is capable of browsing and searching into video content. The system integrates several content-based analysis and retrieval modules including concept detection, clustering, visual and textual similarity search, query analysis and reranking, as well as multimodal fusion.

1 Introduction

VERGE interactive video search engine integrates several multimodal indexing and retrieval modules that allow for efficient browsing, and retrieval of video collections. VERGE has participated in several video retrieval related conferences and showcases such as TRECVID [1], and Video Browser Showdown (VBS) [2] and has evolved in order to support Known Item Search (KIS), Instance Search (INS) and Ad-Hoc Video Search tasks (AVS). The VERGE system participating in VBS 2018 incorporates several changes related to the user interface and the functionalities supported compared to the previous version. Section 2 presents the updated modules of the VERGE system and Section 3 describes the new VERGE user interface.

2 Video Retrieval System

VERGE combines advanced browsing and retrieval functionalities with a user-friendly interface, and supports the submission of queries, the fusion and the filtering of relevant results. The following indexing and retrieval modules are integrated in the developed search application: a) Visual Similarity Search; b) High Level Concepts Retrieval; c) Automatic Query Formulation and Expansion; d) ColorMap and Video Clustering; e) Text Based Search; and f) Multimodal Fusion for Multimedia Retrieval. The above modules allow the user to search through a collection of images and/or video keyframes and the user is presented with the results of each module through the graphical user interface. Moreover,

Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

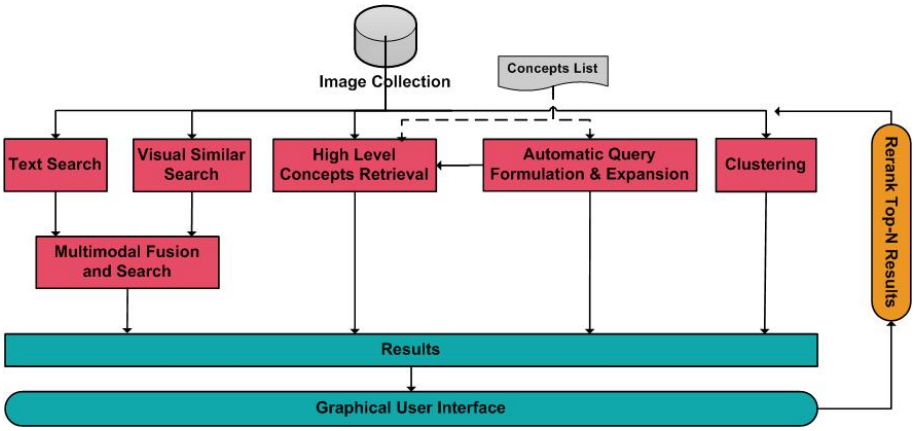

Fig. 1. VERGE Framework.

the system allows the reranking of the top-N results returned, combining also multiple modalities to deliver the final reranked list of retrieved results. Figure 1 depicts the general framework of the VERGE system.

2.1 Visual Similarity Search

The visual similarity search module performs content-based retrieval using deep convolutional neural networks (DCNNs). First, a GoogleNet [3] is trained on 5055 ImageNet concepts, and then, the output of the last pooling layer, with dimension 1024, is used as a global keyframe representation. Fast retrieval of visual similar images is achieved through an IVFADC index database vectors. Then, the K-Nearest Neighbours are computed for the query image [4].

2.2 High Level Concepts Retrieval

This module indexes the video shots based on 1000 ImageNet concepts, 345 TRECVID SIN concepts, 500 event-related concepts, and 205 place-related concepts [5]. To obtain scores regarding the 1000 ImageNet concepts, we applied five pre-trained ImageNet deep convolutional neural networks (DCNNs) on the AVS test keyframes [5]. The output of these networks was averaged in terms of arithmetic mean to obtain a single score for each of the 1000 concepts. To obtain the scores regarding the 345 TRECVID SIN concepts we fine-tuned (FT) the ResNet pretrained ImageNet network on the 345 concepts using the TRECVID AVS development dataset and the extension strategy proposed in [6]. We applied the fine-tuned network on the AVS development dataset and we used as a feature (i.e., a global keyframe representation) the output of the last hidden layer to train one Support Vector Machine (SVM) per concept. Subsequently, we applied this FT network on the AVS test keyframes to extract features, and used them as input to the trained SVM classifiers in order to gather scores for each

Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

of the 345 concepts. Finally, to obtain scores for the event- and place-related concepts we applied the publicly available DCNNs that have been fine-tuned on the EventNet [7] and Places [8] dataset, respectively.

2.3 Automatic Query Formulation and Expansion using High-Level Concepts

This module formulates and expands an input query in order to translate it into a set of high-level concepts C_Q, as proposed in [9]. First, we search for one or more high-level concepts that are semantically similar to the entire query, using the Explicit Semantic Analysis (ESA) measure [10]. If such concepts are found (according to a threshold θ) we assume that the query is well described by them, the selected concept(s) is (are) added in the set C_Q (which is initially an empty set), and no further action is taken. If this is not the case, we examine if any of the concepts in our concept pool appears in any part of the query by string matching, and if so these concepts are added in C_Q. Then, we transform the original query to a set of elementary "subqueries", using Part-of-Speech tagging and a task-specific set of NLP rules. For example, if the original query contains a sequence in the form "Noun Verb Noun", this triad is considered to be a "subquery"; the motivation is that such a sequence is much more characteristic of the original query than any of these three words alone would be, and moreover it is easier to find correspondences between this and the concepts in our pool, compared to doing so for a very long and complex query. Subsequently, we check if any of the "subqueries" exists in our concept pool by calculating the ESA relatedness between each "subquery" and each of the concepts in the pool. If there are concepts that exceed the threshold θ, these are added into the set C_Q. Otherwise, the original query and all the subqueries are used as input to the zero-example event detection pipeline [11], which jointly considers all this input and attempts to find the concepts that are most closely related to it. The outcome of this pipeline is a set of concepts C_Q that describe the input query.

2.4 Clustering

Two clustering approaches are applied for the effective visualization of the dataset:
ColorMap clustering: Video keyframes are clustered by color using Self Organizing Maps into color classes, updating the corresponding previous version [12]. Three MPEG-7 descriptors related to color (i.e. Color Layout, Color Structure, Scalable Color) are extracted from each video frame, and each color cluster provides a representative image for visualization in the VERGE GUI.
Video clustering: This method clusters videos by using the visual features of their keyframes. Specifically, for each video, we retrieve the top-N most similar keyframes of each video keyframe. Then, a simple majority vote algorithm is applied, which counts the frequencies of the returned videos. In the sequel, the frequency values are normalized per video, and we consider as similar to a video the top-M videos that are linked with a similarity value exceeding a pre-defined value. The videos and the links among them are visualized as a network.

Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

2.5 Text Based Search

The text based search module allows for text retrieval in the metadata text describing each video, in the text extracted from the ASR (provided with the video data), and finally in captions extracted from the keyframes using DenseCap [13] using Apache Lucene. It should be noted that textual concepts are also extracted using the DBpedia Spotlight annotation tool, which annotates automatically DBpedia entities in natural language text [14].

2.6 Multimodal Fusion and Search

The fusion module combines high-level concepts and low-level features from visual content with video textual metadata and ASR output, by fusing the similarities per modality in a non-linear graph-based approach [15]. The best performing modality among visual descriptors (Section 2.1), visual concepts (Section 2.2) and textual concepts (Section 2.5) is initially used to filter-out irrelevant retrieved results, keeping only the top-l keyframes, which are then reranked. The $l \times l$ similarity matrices and the $l \times 1$ similarity vectors are combined to obtain the final fused similarity score to rank the l keyframes for the final output.

3 VERGE User Interface and Interaction Modes

This year we introduce a novel graphical user interface (Fig. 2) with an alternative look and feel, always designed to offer the end users an intuitive experience while searching for an image shot or a video. The main difference from previous versions is the absence of links and navigation through pages, since now all retrieval utilities are displayed in a dashboard-like manner and results are shown in the same page.

The VERGE user interface [a] consists of three components: a vertical dashboard menu on the left, a results panel that covers most of the screen and a film-strip on the bottom right. Details for each component follow.

Describing the menu from top to bottom, it contains the applications logo, a countdown that applies to the contest showing the remaining time for submission, a slider to adjust the size of the shots and an undo button to restore the previous results. A novel widget is a switch that toggles between two states, *New* and *Rerank*; in the first option any search module will retrieve fresh results, while in the second option resulted shots will be reranked by a selected retrieval approach. Next, all different search capabilities follow as sliding boxes. Namely, *Text Search* is a one-line text input field that looks for given keywords in the videos metadata, e.g. description of content, (Section 2.5), *Search for Concepts* converts a sentence to recommended concepts (Section 2.2), while *Concepts* present the full list offering auto-complete suggestion and multiple selection (Section 2.3). Furthermore, *Events* refer to queries that combine persons, objects,

[a] http://mklab-services.iti.gr/vbs2017/

Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

Fig. 2. Screenshot of VERGE video retrieval engine.

locations, and activities, and *Videos* provide a video-based representation of results. Finally, *Clusters* give a visual grouping of shots that are more similar, and *Colors* include a palette with some basic shades that results can be mapped to (Section 2.4).

The main component of the user interface displays results as images in a grid view, ranked according to retrieval scores. Hovering on a single shot allows users to run the *Visual Similarity* modality (Section 2.1) or submit it to the contest. Clicking on a shot fills the film-strip on the bottom with every frame of the video where this specific frame belongs to, in a chronological order.

To illustrate the capabilities of the VERGE system, follows a simple usage scenario where we are interested in finding *a woman working on the computer*. We can initiate the search procedure by selecting the concepts "Adult Female Human" and "Computers" or the relevant event "Sitting with a laptop". Alternatively, we can type the query in natural language and receive proposed concepts. After the first results are retrieved, we are able to perform visual similarity to the most relative shot or rerank the images based on a specific color, e.g. to detect shots with a blue background. If the above results are not satisfying, we can always refer to the other modalities, such as clustering or text search in the metadata. Anytime, the full video of a shot is available in the film-strip, just with a click.

4 Future Work

Future work includes developing a color sketching module that provides the user the ability to draw a sketch using colors. Another feature would be to minimize the time response of the system in order to improve the user experience.

Proc. 24th International Conference on Multimedia Modeling (MMM 2018). Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-319-73600-6_48

Acknowledgements This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-645012 KRISTINA, H2020-779962 V4Design, H2020-732665 EMMA, and H2020-687786 InVID.

References

1. G. Awad, A. Butt, J. Fiscus, et al. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *Proceedings of TRECVID 2017*. NIST, USA, 2017.
2. C. Cobârzan, K. Schoeffmann, W. Bailer, et al. Interactive video search tools: a detailed analysis of the video browser showdown 2015. *Multimedia Tools and Applications*, 76(4):5539–5571, Feb 2017.
3. C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 00:1–9, 2015.
4. H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, January 2011.
5. F. Markatopoulou, A. Moumtzidou, and D. Galanopoulos et others. ITI-CERTH participation in TRECVID 2016. In *TRECVID 2016 Workshop*, USA, 2016.
6. N. Pittaras, F. Markatopoulou, V. Mezaris, and I. Patras. Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In *MMM (1)*, volume 10132 of *Lecture Notes in Computer Science*, pages 102–114, 2017.
7. Y. Guangnan, Yitong L., Hongliang X., et al. Eventnet: A large scale structured concept library for complex event detection in video. In *ACM MM*, 2015.
8. B. Zhou, A. Lapedriza, J. Xiao, et al. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.
9. F. Markatopoulou, D. Galanopoulos, V. Mezaris, et al. Query and keyframe representations for ad-hoc video search. In *ICMR*, pages 407–411, 2017.
10. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611. Morgan Kaufmann Publishers Inc., 2007.
11. D. Galanopoulos, F. Markatopoulou, V. Mezaris, and I. Patras. Concept language models and event-based concept number selection for zero-example event detection. In *ICMR*, pages 397–401. ACM, 2017.
12. A. Moumtzidou, Th. Mironidis, F. Markatopoulou, et al. VERGE in VBS 2017. In *International Conference on Multimedia Modeling*, pages 486–492. Springer, 2017.
13. J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574, 2016.
14. J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA, 2013. ACM.
15. I. Gialampoukidis, A. Moumtzidou, D. Liparas, et al. Multimedia retrieval based on non-linear graph-based fusion and partial least squares regression. *Multimedia Tools and Applications*, pages 1–21, 2017.