CrowdNote

Crowdsourcing Environment for Complex Video Annotations

Marcello N. de Amorim
Federal University of Espírito Santo
novaes@inf.ufes.br

Ricardo M. C. Segundo
Federal University of Espírito Santo
rmcs87@gmail.com

Celso A. S. Santos
Federal University of Espírito Santo
saibel@inf.ufes.br

Orivaldo de L. Tavares
Federal University of Espírito Santo
tavares@inf.ufes.br

ABSTRACT

This paper introduces CrowdNote, a crowdsourcing environment for complex video annotations without the need for trained workers or specialists. CrowdNote is based on a cascading microtasks approach to achieve complex video annotation by aggregating and processing multiple simple annotations collected from the crowd. The approach consists of dividing complex annotation tasks into simpler and smaller microtasks and cascading them to generate a final result. Moreover, this approach allows using simple annotation tools rather than complex and expensive annotation systems. Also, it tends to avoid activities that may be tedious and time-consuming for contributors, that are the workers in crowdsourcing scenarios. The CrowdNote instance presented in this paper produces enriched videos in which all extra content added is provided, selected and positioned by the crowd.

KEYWORDS

Crowdsourcing, Video Annotation, Human Computation, Microtasks, Multimedia Systems, Video Enrichment, Amazon Mechanical Turk, Crowdflower, Microworkers

1 INTRODUCTION

Video is a very effective information container and is a highly expressive type of media, capable of providing a large semantic load by presenting different audiovisual components coherently [2]. However, video can be considerably more useful when carrying metadata that can be used by video applications and is often represented as video annotations.

Video annotation involves inserting tags into video objects to describe their content and context, also to describe media characteristics such as quality, coding, among other features [15]. In other words, they are used to make easier the work of users and systems that can handle annotated items. These annotations facilitate the creation of video applications for content-based distribution [17], indexing [18], summarization [7], synchronization [11],navigation [9], composition [16], among many others, by both automatic and manual means [14].

In this paper, video annotations are categorized as simple and complex ones, considering that simple annotations are those that can be acquired with a simple interaction of the workers in a microtask. Complementarily, a complex annotation is one that requires that the worker execute a more tedious, hard or time-consuming task, in which he needs to perform multiple interactions.

A frequent problem of using a crowdsourcing approach to video annotation is to balance the relationship between task complexity and cost. Simple annotation tasks, such as clicking an object on a video, can be done in a few seconds by anyone. Otherwise, more complex tasks such as providing complementary content and positioning it in the right position on a video, require some expertise of contributors and are more costly to them. In a crowdsourcing context, microtask is a ubiquitous designation for simple tasks that can be performed by any contributor quick and easily [6].

CrowdNote is a crowdsourcing environment based on a microtask cascading approach [3], and capable of achieving complex video annotation without the need for specialized or trained workers, and it can be used as a template to build different crowdsourcing applications based on video annotation. This environment allows collecting contributions from webpages, systems and even platforms such as Amazon Mechanical Turk, Crowdflower, Microworkers [6], and offers a collection of templates for microtasks, including annotation tools, persistence models, and aggregation algorithms.

The system presented in this paper is a CrowdNote instance that produces enriched versions of videos by adding extra content such as images, text boxes, Wikipedia content and Youtube videos. The application presented can be freely downloaded and used in academic context.

The remaining of this paper is structured as follows. Section 2 presents related projects. Section 3 presents the CrowdNote architecture. Section 4 presents the CrowdNote instance for video enrichment. Finally, section 5 concludes the paper with final considerations.

2 RELATED WORK

Crowdsourcing video annotation approaches are used in various scenarios to gather information of various types, such as temporal synchronization [11], events [10], scene objects [12], actions[4], geo-tagging [1] and captions [5].

Studying these related projects have shown that crowdsourcing solutions to achieve complex annotations, such as [12, 13], are plentiful but require elaborate tools, costly tasks, and skilled workers.

In: XVI Workshop de Ferramentas e Aplicações (WFA 2017), Gramado, Brasil. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres. Porto Alegre: Sociedade Brasileira de Computação, 2017.

ISBN 978-85-7669-380-2.

On the other hand, works that use simple annotation tools and unskilled crowd usually achieve only simple annotations.

The differential of CrowdNote is the capability to use simple tools to generate complex annotations from a non-expert crowd. By dividing a complex task into a microtasks sequence, is possible to reduce its complexity to the point where each activity can be performed by any member of the crowd using a simple tool.

3 ARCHITECTURE

CrowdNote was developed as a classic Web system. To facilitate the sharing of all produced software, only technologies that do not require complex infrastructure were adopted. The Server was fully developed in NodeJS for easy deployment, the Client was developed in HTML 5 to improve compatibility, and the Database uses MongoDB as No-SQL database for flexible persistence.

The architecture of the CrowdNote is illustrated in Figure 1 in which is possible to observe the 3 main components: Server, Database, and Clients.

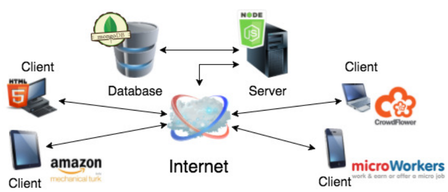

Figure 1: CrowdNote Architecture

3.1 The Server Component

The server system, illustrated in Figure 2, is composed of 3 modules: Collector, Aggregator and Player Provider.

- **Collector:** The Collector sends the jobs to the workers, receives the annotations from them, and stores the annotations into the Database. Information is exchanged between the Collector and the Client as JSON messages through HTTP requests for cross-platform compatibility.
- **Aggregator:** The Aggregator verifies, filters, groups, and processes the collected annotations of the crowd according to the rules defined for each task, and then stores the result in the Database.
- **Player Provider:** The Player Provider sends to the client the annotations, the extra content, and the original video. Thus, the player on the client can play the enriched video synchronously.

3.2 The Database Component

The persistence was addressed using MongoDB, which delivers a very attractive solution to build No-SQL databases with some characteristics that meet the crowdsourcing requirements such as high write load, high availability in an unreliable environment, easy scaling and partition, heterogeneous data into the same collection.

In this model, JSON document collections are used instead of tables, and the documents in each collection may have a different structure to store different attributes. This feature allowed the modeling of a very simple database structure, composed of 3 collections of documents. It was possible because documents in the Input and Output collections can contain different fields according to the task that consumes or generates the entries.

The Video collection stores entries related to the video segments dataset, the Input collection stores the input entries to the tasks, and the Output collection stores the contributions collected from the crowd. The result of the aggregation for each task is stored in the Input collection to be used by the next task, supporting the cascading tasks approach.

3.3 The Client Component

The client consists of simple forms-based annotation tools and a player capable of playing video and extra content synchronously. The client has been fully developed in HTML5, in the simplest way possible. For each task, a simple annotation tool was created to collect contributions.

The Client communicates with the Server through JSON messages and HTTP requests so that they can be deployed on different systems and sites or even on crowdsourcing platforms such as Amazon Mechanical Turk, Crowdflower and Microworkers [6], as long as the JSON structure is respected. By using these platforms, the search and reward of workers are delegated to them, however, there is a financial cost involved in doing so.

3.4 Workflow

The 3 main components of CrowdNote communicate through data flows from **A** to **G**, as can be seen in the workflow in Figure 2.

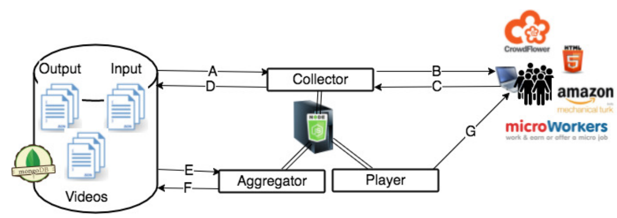

Figure 2: CrowdNote Workflow

- **A:** To generate each job to be sent to a worker, the Collector receives an entry from the Input Collection and the corresponding entry from the Videos Collection.
- **B:** The Collector sends a job to an instance of the Client, to be executed by a worker.
- **C:** The Client sends to the Collector the annotation made by the worker for the job received.
- **D:** The Collector stores in the Output collection the annotation collected from a worker.
- **E:** The Aggregator receives from the Output collection all annotations collected for a given task.
- **F:** The Aggregator stores the resulting entries from the aggregation process in the Input collection so that they are supplied as input to the next task.
- **G:** The outcome of the cascade of microtasks is sent by the Player Provider to the client so that it can play the video synchronously with the extra content.

4 VIDEO ENRICHMENT INSTANCE

CrowdNote is an environment that provides a collection of annotation tools, aggregation methods, and persistence models that can be selected, sequenced, and modified to generate different types of crowdsourcing applications based on video annotations. In order to create a system based on this environment, it is necessary to define the required microtasks sequence, and then select and specialize the resources provided by CrowdNote.

To demonstrate its working was built an instance of CrowdNote which consists of a system for video enrichment by adding extra content provided by the crowd. In this system, contributors are responsible for identifying the points of interest in the video, suggesting what content should be associated with each one, deciding the best suggestion for each point of interest, and finally deciding the best position in the video to present each content.

The extra content suggested by the crowd are images, text boxes, Wikipedia content, and Youtube videos, and the result delivered by this system is an enriched video, that consists of the original video presented synchronized with the extra content provided and selected by the crowd.

The approach taken to achieve the complex annotation needed to enrich the videos is to cascade microtasks that collect simple annotations, instead of collecting complex annotations for each contribution. In this way, people without specialization or training can contribute to the process.

- **Task 1 - Identify the points of interest** in the video that should be associated with the extra content. The first microtask is to send video segments to the worker and ask him to identify in this segment something that he believes deserves to be highlighted or supplemented. The aggregation rules for this microtask are to temporarily group the annotations with a tolerance of 0.5 sec, to count and to merge similar annotations in each group, and to determine for each time group which is the predominant point of interest in the annotations.
- **Task 2: Provide extra content suggestions** for each point of interest. In the second task, the worker receives a point of interest and should suggest extra content related to it. This content can be a text, an image, a YouTube video or a Wikipedia page. The aggregation of the second task consists in grouping the contributions by a point of interest and joining similar contributions to avoid duplicity.
- **Task 3: Ranking the suggested content** provided by each point of interest. In the third microtask, the worker receives a point of interest and the content suggestions for it. The contributor should choose the most appropriate content for the point presented. The aggregation rule for this task is to select the most popular content for each point of interest.
- **Task 4: Determine the positions** to display the extra content associated with each point of interest. In this task, the worker receives an item that represents a point of interest and chooses the position in the video most suitable to display it. The aggregation method for this task calculates the average coordinate for each item to be displayed in the video.

4.1 Cascading Microtasks

The adopted approach consists of dividing the complex annotation into simple annotations that can be collected by a set of simple annotation tools. Each of these simple annotations is collected by a microtask.

As is illustrated in Figure 3, the input for each task is generated by the Aggregator after the previous task, except for the task 1. For this task is provided a bootstrap Input that is a list of video segments provided by the owner, that is who initiate the process. Each entry of the bootstrap input can represent a semantic block of the video.

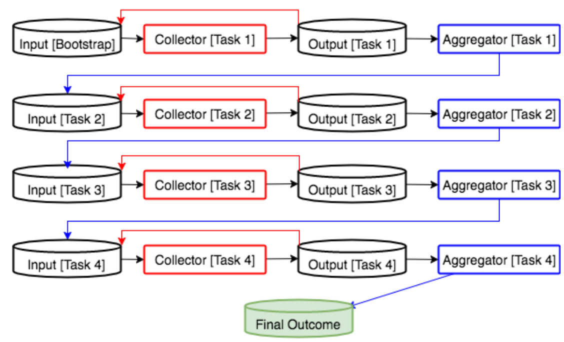

Figure 3: Cascading Microtasks

Other applications that use CrowdNote may use different strategies to segment videos such as fixed time-length, SRT files, or even add a microtask to segment videos.

4.2 Task 1

Identify Points of Interest: The first annotation microtask is supported by the tool represented in Figure 4, collecting identification for points of interest. In this task, the contributor receives a segment of video that should be watched, and if was found any point of interest, it should be marked and briefly described. These points of interest can be gestures, words, expressions, facts, concept, characters, events or anything that can be related to extra content.

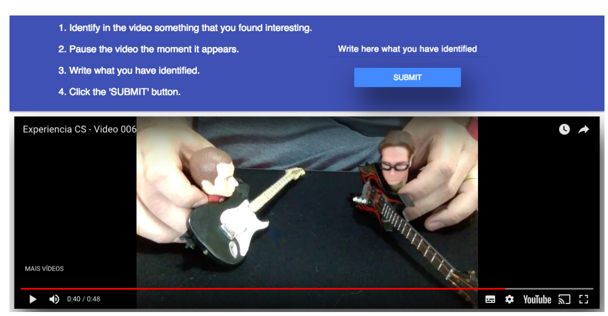

Figure 4: Annotation Tool for Task 1

4.3 Task 2

Provide extra content suggestions: The second task took as input the aggregated result from the task 1 that is a list of points of interest identified by the workers. This microtask is supported by the annotation tool represented in Figure 5. This tool presents to the worker a point of interest and the video segment positioned at the moment it occurs. This way, is possible to use the video for reference and contextualization.

Through this tool, the worker can contribute by writing a text related to the point of interest, sending an image or sending a link to a YouTube video or a Wikipedia page.

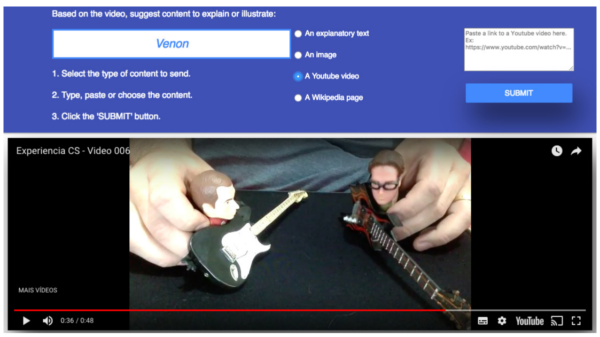

Figure 5: Annotation Tool for Task 2

When the collection of contributions for this task is done, the Aggregator groups the content of the sender by a point of interest, and then joins the similar suggestions. In this way, a list of points of interest with a set of content suggestions for each is added to the next task, without repeated suggestions.

4.4 Task 3

Ranking Suggestions: The third task receives as input the list of points of interest, with the content suggestions for each of them. For each job, the annotation tool illustrated in Figure 6 shows the worker a point of interest and the video positioned at the time that point occurs. The annotation tool displays the content suggestions for that point of interest below the video, so is possible to browse through the content to choose the most appropriate one.

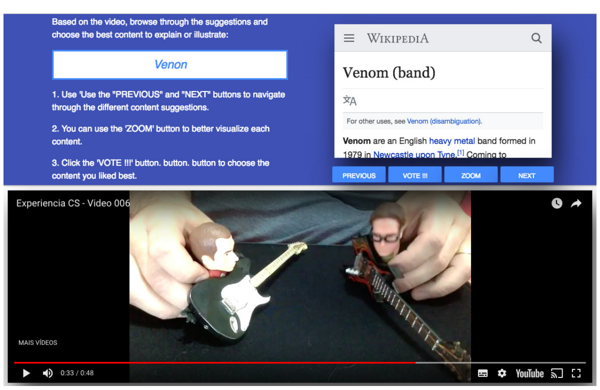

Figure 6: Annotation Tool for Task 3

The worker can enlarge each content to see it better, how can be seen in Figure 7. In addition to playing the videos as a suggestion of content.

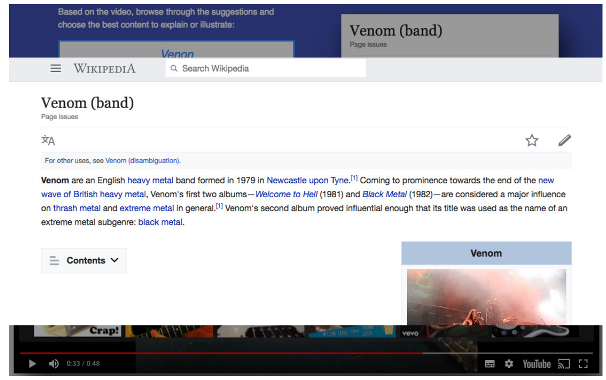

Figure 7: Annotation Tool for Task 3 - Zoom

The aggregation process for this task counts the votes for each content suggestion and chooses the most popular content for each point of interest.

4.5 Task 4

Determine the positions: The last task receives as input the list of points of interest and the content chosen to associate with it. For each job, the tool shown in Figure 8 shows the worker the video that is positioned at the time the point of interest occurs and the reference item for the content selected in the video.

The contribution to this task is to suggest the best position to present the extra content, using the annotation tool to determine this position. The tool allows the worker to change the position of the items in the video by clicking the desired point. Among the 4 microtasks, this is the fastest and easiest to perform.

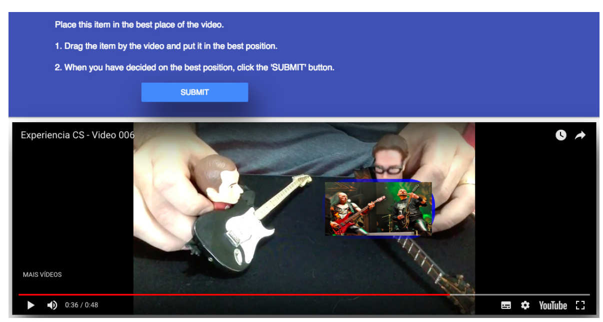

Figure 8: Annotation Tool for Task 4

Following the studies about the wisdom of the crowd, the strategy to determine the correct position is to calculate the average coordinate of the contribution for each content [8]. In this way, the aggregation process calculates the average coordinate of the items, based on the contributions of the crowd. The result of this process is the position where each item related to a point of interest will appear in the video.

4.6 Player

The presentation system, shown in Figure 9, receives the video, extra content, and necessary metadata from the Player Provider. This system is capable of reproducing the original video synchronized with the extra content, that is displayed every time a point of interest happens in the video. Is important to remind that all extra content displayed with the video was provided, selected and positioned by the crowd.

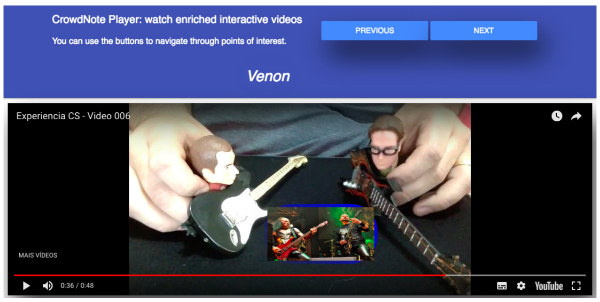

Figure 9: Displaying an extra content item over the video

When the user clicks on some extra content displayed in the video, the presentation is paused and a larger preview of the selected content is displayed in a zoom box. This system features navigation by extra-content instead of the traditional timeline navigation, making available a button-bar with buttons to navigate among the extra contents.

5 FINAL REMARKS

This paper presented CrowdNote, a crowdsourcing environment that can achieve complex video annotation from a crowd of untrained and nonspecializing contributors. CrowdNote can be used as a template for different kinds of crowdsourcing applications based on video annotation.

To demonstrate how CrowdNote works, was created an instance of it that consists of a video enrichment application. This application used the crowd to annotate the points of interest present in the video, collect extra content to associate with them, select the collected content, and to position the extra content in the video when each point of interest happens.

However, various types of applications can be generated as instances of CrowdNote. It is possible to create applications to annotate multiple aspects of scenes, generate transcriptions, human translations, and so on.

The most interesting aspect of CrowdNote is that it offers a way to make complex video annotations without the work of experts, without the need to create expensive and sophisticated annotation systems or ask employees to perform difficult and laborious tasks.

Traditional crowdsourcing approaches are struggling to achieve the complex annotations task, but the strategy of dividing the problem into microtasks that collect simple annotations and cascades them to generate complex annotations proved to be functional.

Future versions of CrowdNote will incorporate features to assist the owner in generating bootstrap input, as well as selecting aggregation methods and annotation tools from a sample library. Another topic to be studied is how to effectively integrate CrowdNote with systems that use Deep Learning.

The presented application can be freely used, modified and downloaded from https://github.com/marcellonovaes/crowdnote, and a demo video can be found on https://youtu.be/FqGbkSoeB2U.

ACKNOWLEDGMENTS

The authors would like to thank FAPES, CAPES and CNPq for financial support of this research.

REFERENCES

[1] Si Chen, Muyuan Li, Kui Ren, and Chunming Qiao. 2015. Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th*. IEEE, 1–10.

[2] M. N. de Amorim, C. A. S. Santos, and O. L. Tavares. 2016. ExCAM - Uma metodologia Crowsourcing para a autoria de conteudo extra para videos. In *WebMedia 2016 WTD*. Teresina - PI, Brazil.

[3] M. N. de Amorim, R. M. C. Segundo, C. A. S. Santos, and O. L. Tavares. 2017. Video Annotation by Cascading Microtasks: a Crowdsourcing Approach. In *WebMedia 2017 - Full and Short papers*. Gramado, RS. https://doi.org/10.1145/3126858.3126897

[4] Travis Desell, Kyle Goehner, Alicia Andes, Rebecca Eckroad, and Susan Ellis-Felege. 2015. On the effectiveness of crowd sourcing avian nesting video analysis at Wildlife@ Home. *Procedia Computer Science* 51 (2015), 384–393.

[5] Rucha Deshpande, Tayfun Tuna, Jaspal Subhlok, and Lecia Barker. 2014. A crowdsourcing caption editor for educational videos. In *Frontiers in Education Conference (FIE), 2014 IEEE*. IEEE, 1–8.

[6] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. 2015. The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk. In *Proceedings of the 24th ICWWW (WWW '15)*. ACM, New York, NY, USA, 238–247. https://doi.org/10.1145/2736277.2741685

[7] Guilherme Fião, Teresa Romão, Nuno Correia, Pedro Centieiro, and A. Eduardo Dias. 2016. Automatic Generation of Sport Video Highlights Based on Fan's Emotions and Content. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology (ACE2016)*. ACM, New York, NY, USA, Article 29, 6 pages. https://doi.org/10.1145/3001773.3001802

[8] FRANCIS GALTON. 1907. Vox Populi (The Wisdom of Crowds). *Nature* 75, 1949 (1907), 450–451. https://doi.org/10.1038/075509f0

[9] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. 2008. Video Object Annotation, Navigation, and Composition. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 10. https://doi.org/10.1145/1449715.1449719

[10] Gunhee Kim, Leonid Sigal, and Eric P. Xing. 2014. Joint Summarization of Large-Scale Collections of Web Images and Videos for Storyline Reconstruction. In *Proceedings of the 2014 IEEE CCVPR (CVPR '14)*. IEEE Computer Society, Washington, DC, USA, 4225–4232. https://doi.org/10.1109/CVPR.2014.538

[11] R. M. C. Segundo, M. N. de Amorim, and C. A. S. Santos. 2017. CrowdSync: User generated videos synchronization using crowdsourcing. In *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*. Poznan, Poland.

[12] Bill Thies, Ed Cutrell, and Andrew Cross. 2014. VidWiki: Enabling the Crowd to Improve the Legibility of Online Educational Videos. ACM Conference on Computer Supported Cooperative Work, 1167–1175.

[13] Carl Vondrick, Donald Patterson, and Deva Ramanan. 2013. Efficiently Scaling Up Crowdsourced Video Annotation. *Int. J. Comput. Vision* 101, 1 (Jan. 2013), 184–204. https://doi.org/10.1007/s11263-012-0564-1

[14] Meng Wang and Xian-Sheng Hua. 2011. Active Learning in Multimedia Annotation and Retrieval: A Survey. *ACM Trans. Intell. Syst. Technol.* 2, 2, Article 10 (Feb. 2011), 21 pages. https://doi.org/10.1145/1899412.1899414

[15] Meng Wang, Xian-Sheng Hua, Jinhui Tang, and Richang Hong. 2009. Beyond Distance Measurement: Constructing Neighborhood Similarity for Video Annotation. *Trans. Multi.* 11, 3 (2009), 12. http://dx.doi.org/10.1109/TMM.2009.2012919

[16] Stefan Wilk, Stephan Kopf, and Wolfgang Effelsberg. 2015. Video Composition by the Crowd: A System to Compose User-generated Videos in Near Real-time. In *Proceedings of the 6th ACM MSC (MMSys '15)*. ACM, New York, NY, USA, 13–24. https://doi.org/10.1145/2713168.2713178

[17] Jun Zhang, Xiaoming Fan, Jianyong Wang, and Lizhu Zhou. 2012. Keyword-propagation-based Information Enriching and Noise Removal for Web News Videos. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 561–569. https://doi.org/10.1145/2339530.2339620

[18] Yifan Zhang, Xiaoyu Zhang, Changsheng Xu, and Hanqing Lu. 2007. Personalized Retrieval of Sports Video. In *Proceedings of the IWMIR (MIR '07)*. ACM, New York, NY, USA, 313–322. https://doi.org/10.1145/1290082.1290126