Rapid Model-Driven Annotation and Evaluation for Object Detection in Videos

Marc Ritter[1](✉), Michael Storz[2], Manuel Heinzig[1], and Maximilian Eibl[2]

[1] Junior Professorship Media Computing, Technische Universität Chemnitz,
09107 Chemnitz, Germany
{marc.ritter,manuel.heinzig}@informatik.tu-chemnitz.de

[2] Chair Media Informatics, Technische Universität Chemnitz,
09107 Chemnitz, Germany
{michael.storz,eibl}@informatik.tu-chemnitz.de

Abstract. Nowadays, the annotation of ground truth and the automated localisation and validation of objects in audiovisual media plays an essential role to keep pace with the large data growth. A common approach to train such classifiers is to integrate methods from machine learning that often demand multiple thousands or millions of samples. Therefore, we propose two components. The first constraints the annotation space by predefined models and allows the creation of ground truth data while providing opportunities to annotate and interpolate objects in keyframes or in-between by granting a user-friendly frame-wise access. The graphical user-interface of the second component focuses on the rapid validation of automatically pre-classified object instances in order to alter the assignment of the class label or to remove false-positives to clean-up the result list which has been successfully applied on the task of Instance Search within the TRECVid evaluation campaign.

Keywords: Model-based annotation · Object detection · Instance search · Rapid evaluation · Image and video processing · Big data

1 Introduction

One way to cope with the ever increasing amounts of audiovisual data recorded day by day is the automatic detection and storage of object instances in databases in order to make large archives searchable. In the last decades, scientific research has focused on the detection of specific object classes like faces and pedestrians [1]. However, the creation of robust systems for such unconstrained amounts of data still appears as a very challenging task even in the well-known field of face detection [2]. Frequently, the mere complexity entails the need for hundreds or thousands of intellectually selected positive training samples as well as millions to billions of negative non-object samples while being utilized in appearance-based machine learning algorithms.

Beyond that, the development of algorithms and systems for the automatic detection of various object instances with a small sample size has been pursued by the community of the *Text Retrieval Evaluation Campaign on Videos*

M. Antona and C. Stephanidis (Eds.): UAHCI 2015, Part I, LNCS 9175, pp. 464–474, 2015.
DOI: 10.1007/978-3-319-20678-3_45

(TRECVid) [3] for many years. Ordinarily, it is well-known that the intellectual annotation and localization of target objects [4] is a repetitive, time-consuming, demanding, but yet necessary and critical task when processing large data collections in order to determine the performance of automatic detection algorithms, draw assumptions over possible misfits, or identify areas of improvement.

The previous work of Ritter & Eibl [5] and Storz et al. [6] proposed a strategy to conduct image-based annotations of extracted keyframes after the application of shot boundary detection algorithms by using predefined models. While building on that work, this contribution introduces some handy methods to facilitate the creation of almost arbitrary ground-truth data while providing the means and opportunities for rapid model-driven annotations in videos by restricting the annotation space to specified properties of the underlying domain. Furthermore, a fast selection scheme is introduced to increase the speed in which the assessment and evaluation of object detection algorithms is performed.

The remainder of this paper is organized as follows: Sect. 2 gives an overview about other approaches from the literature concerning model-based annotation as a base methodology. Section 3 describes our approach for video annotation and the validation of outcomes yielded from automated detections of object classifiers in the context of a specified TRECVid use case scenario that is also evaluated shortly in Sect. 4. A brief summary and an outlook to future work in Sect. 5 concludes this contribution.

2 Related Work

In visual media we need algorithms to be able to classify a vast range of concepts. According to Forsyth et al. [7], we can differentiate visual concepts in *stuff* meaning materials (e.g. grass, road) and *things* meaning objects like cars or persons. The concepts can be content-independent (i.e. author name), content-dependent (e.g. texture, shape), and content-descriptive (semantics, shape is a car) [8]. These visual and theoretical differences between annotations open up a vast annotation space that lead to the development of a variety of different annotation tools. Tools and applications vary greatly from e.g. game based web applications like *ESP Game* [9] to fairly complex tools like the *LHI* annotation tool [10] that includes sophisticated methods for graph based segmentation, scene decomposition, and semantic annotation as well.

However, the usage of annotation models allows to cover a broad range of possible annotation types that are often directly related to different use cases. Specific use cases like video annotation or the verification and evaluation of object candidates that were detected and localized by a trained detector are described in the following paragraphs.

2.1 Model Based Annotation

Similarly to *ViPER-GT* [11] we apply annotation models to precisely define the amount and scope of information that needs to be annotated in visual media.

Models serve as a annotation template consisting of different geometric (e.g. bounding boxes or polygons) and semantic components like text called *model elements* [6].

The incorporation of such a model facilitates the workflow within the annotation process while reducing the necessary input of information to constraint properties. With regards to the annotation of objects and their position in an image this might comprise actions like adjusting marker points, determining the area of a bounding box or entering a textual caption. This procedure leads to a specific and dependable structure of the annotated results making them comparable even if the intellectual annotations are created collaboratively by multiple workers with differing experience.

2.2 Video Annotation

The annotation of video sequences can serve different purposes and the creation of a training or validation dataset for training object classifiers is only one of them. Tools like *Anvil* [12], *ELAN* [13], or *VCode* [14] focus on the annotation of speech or the coding of behaviour and interactions of actors. These most frequently used video analysis tools are applied to many different research domains like human-computer interaction, linguistics, and social sciences.

Another category comprises tools like *Advene* [15] and *VideoANT* [16] that facilitate the sharing, communication, and comprehension of video content by providing interfaces to comment, to discuss, and to link other media.

However, the most important category within the context of this contribution focuses on the aforementioned task of video annotation for the creation of training and validation datasets. The subsequent tools differ greatly in presentation style and in the range of functionality, but share the ability to create spatial annotations in videos.

VATIC [17] is a web annotation tool that can be used with *Mechanical Turk*[1] in order to outsource annotation tasks. In comparison to other tools, it offers a very reduced and easy to learn interface. After drawing a bounding box around an object, it offers a brief categorization of the object class (e.g. person or car) and allows to specify certain properties that for instance might be used to mark an object as occluded. Object annotation over time is accomplished by the intellectual masking of a small subset of frames and the automatic interpolation inbetween. To the best of our knowledge, the tool does not differentiate between annotations that were created by hand or result from automated interpolations while lacking frame-wise access, whereas a modification of previous annotations may be challenging.

In contrast the *Semantic Video Annotation Suite (SVAS)* [18] offers a more complex interface with semi-automatic annotation capabilities. *SVAS* uses a automatic video preprocessing to detect shots, extract key frames and capture image features. In the keyframe based annotation process the video is divided into the detected shots. A *SIFT* based object and shot re-detection can be

[1] https://www.mturk.com, 19.02.2015.

applied to an intellectual object annotation in a keyframe to retrieve the specific object in other keyframes as well. An object tracking mechanism is used to track the object in between. The described process can effectively minimize the intellectual annotation effort assuming automatic annotation is accurate enough. According to the authors, accuracy decreases if object has low textural information, is small in size or moving, which is quite a common case in challenging object recognition tasks.

The video annotation tool *ViPER-GT* can either be used for intellectual annotation of video content or to view automatically generated markups. In a similar way to *VATIC*, annotation can be achieved via an interpolation approach that visualizes the intellectual annotations and interpolations likewise in a *timeline view* while allowing the definition of annotation models. Displayed in a spreadsheet view, it shows the current values of the selected frame like the location of a bounding box. Unfortunately the support for different video formats turns out to be minimalistic. Furthermore, interpolation proves to be slightly cumbersome since it requires the navigation of several context menus and the manual typing of frame numbers. Nevertheless, intellectually annotated frames are highlighted in the *timeline view* and can be adjusted easily.

All the aforementioned tools contain similar components since they are used to navigate in a video file, to display, or to employ some sort of annotation. A video player component is often composed of a control bar that allows to play, rewind and fast-forward the video and a graphical editor to create localized annotations like bounding boxes in a displayed frame. Most tools also visualize the occurrence of annotations over time in a *timeline view* by associating a row with an object or coded item. Usually, a colored section in the row of the timeline represents the time span of an occurrence. The linkage of *timeline view* and the video player allows for an annnotation-based video navigation.

Annotating objects on a frame by frame basis is prohibitively time consuming so that assisting mechanisms like linear interpolation, object tracking or even the application of computer vision algorithms for semi-automatic annotation are highly required. The mere availability of assisting mechanisms is not sufficient, they need to be easy to use. If large amounts of objects needs to be annotated, the tools must provide instruments to focus on the current work at hand and by enabling the user to hide annotations or their representations in the interface.

2.3 Evaluation of Detection Results

The aforementioned video annotation tools can be used to create ground truth data that is needed to automatically evaluate trained classifiers. But the creation of accurate ground truth data can be prohibitively time and resource consuming in large datasets on which evaluations are performed nowadays.

A convenient way to measure the quality of a trained classifier is to compare results with annotated ground truth. However, the creation of the ground truth is not always possible. With exemplary application to face labeling and in accordance to *Jain & Learned-Miller* [2, p. 4] it can be stated that "[f]or some image regions, deciding whether or not it presents a 'face' can be challenging.

Several factors such as low resolution, occlusion, and pose of the head may make this determination ambiguous." These findings are especially considered to be meaningful when dealing with large collections of video footage, where validation data might not be available. When considering the exemplary case of detecting frontal faces, the evaluation procedure boils down to a mere supervised reevaluation of all available face detections that were created with a certain classifier repeating the simple but yet not always distinct binary question: "Is a frontal face present in the shown image patch?"

This common situation demands different evaluation mechanisms that do not always require large amounts of ground truth data. Detections need to be scanned for false positives to assess the performance of a trained classifier and enable their improvement. Similarily photo management software like *iPhoto*[2] or *Picasa*[3] require the user to manually accept or decline the assignment of faces to specific persons which is proposed by an integrated face recognition algorithm.

For instance in *iPhoto*, users can select an already defined person whereupon the application retrieves and shows other similar faces from the dataset. Users then may accept or decline proposed faces by clicking on them once or twice respectively. This simple selection scheme allows a very fast evaluation of large amounts of faces. Moreover, this interaction concept can be easily applied to the evaluation of custom classifiers. It could also be regarded as a more concrete implementation of a more generalized scheme that assigns one of several predefined values (here accept or decline) to a specific object (in this case a detection). Hence, the concept can be used for the annotation of relevant position-independent object properties like occlusion or the color of objects. An application that incorporates this kind of functionality should focus on an easy to use interface in order to allow for rapid evaluation or annotation of potentially thousands of detections. Additionally, it should allow the user to customize the number and size of objects shown at the same time on the screen in order to find an adequate representation that is also simple to perceive to account for the large visual variations in classes and properties of different objects.

3 System Description

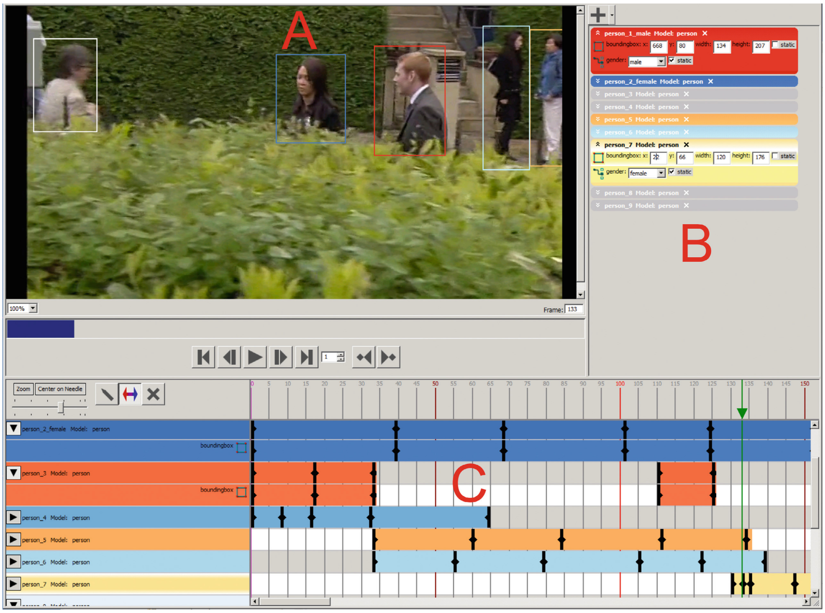
This section investigates the structures of our two main components: The *video annotation component* should enable the user to make use of predefined models in order to grant fast object annotations in videos whereas the main objective of the *evaluation component* consists in the validation of previously classified objects or properties that might be used to improve the performance of machine learning algorithms or to remove false-positives from data sets.

3.1 Video Annotation Component

In a first step we build a paper prototype of the *video annotation component* afterwards the prototype was refined in a usability tests with four students (two

[2] https://www.apple.com/de/mac/iphoto/, 19.02.2015.

[3] http://picasa.google.com/, 20.02.2015.



person_1_male Model: person ✕
boundingbox: x: 668 y: 80 width: 134 height: 207 □ static
gender: male ☑ static

person_2_female Model: person ✕
person_3 Model: person ✕
person_4 Model: person ✕
person_5 Model: person ✕
person_6 Model: person ✕

person_7 Model: person ✕
boundingbox: x: 22 y: 66 width: 120 height: 176 □ static
gender: female ☑ static

person_8 Model: person ✕
person_9 Model: person ✕

person_2_female Model: person
boundingbox
person_3 Model: person
boundingbox
person_4 Model: person
person_5 Model: person
person_6 Model: person
person_7 Model: person

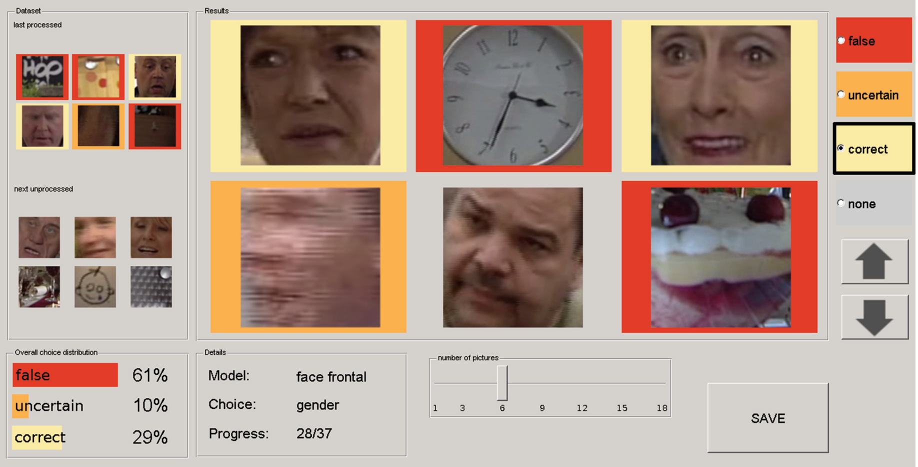
Fig. 1. The *video annotation component* consist of three parts. The *video player* (A) shows the annotations on the current frame. The *model instance list* (B) displays all available objects and their current values. The *timeline view* (C) visualizes all continuous annotations over time, differentiating manual from interpolated annotations.

domain experts). Although representing continuous media in interface mock-ups appears as rather difficult task, the handling of the static interface elements was effectively studied and led the omission of unused components. Most important functionality hidden in context menus was externalised into buttons to facilitate the usability.

Our proposed video annotation component (Fig. 1) consists of the three fundamental parts:

(A) The part of the *video player* shows annotations of the current frame. Colors correspond with object representations in the other parts. The player controls can be used to navigate within video. Besides the regular VCR like functionality (play, jump to start/end) the video can also be navigated frame wise or one can jump to the next intellectual annotation of a selected object.

(B) The *model instance list* on the righthand side represents objects in a row or block of a single color. The list shows all objects in the video file. While colored rows represent objects that are annotated in the current frame, grey rows represent the opposite. All rows can be expanded and collapsed to reveal or hide all model elements. Model element values always correspond to the current frame (like the position of a bounding box). Non changing properties can be marked as static.



false 61%
uncertain 10%
correct 29%

Model: face frontal
Choice: gender
Progress: 28/37

Fig. 2. The *evaluation component* allows the fast validation of detection results. A small subset of the detection results is shown in the center of the application including some statistics about the current state of the evaluation process (bottom left). Users can cycle through and assign the available classification labels to an image with a left mouse click. In uncertain situations, a right mouse click may be used to pop up the original image to display the surrounding context of the image patch.

(C) Similarly to the model instance list the rows in the *timeline view* correspond to the annotated objects. The timeline further indicates the time spans in which the object is annotated. A black line symbolizes an intellectual annotation. The colored space in-between denote linear interpolations between the reference points. The *timeline view* enables the user to switch between annotation modes like the intellectual annotation inside the player component and the interpolation inside the timeline between two or more manual annotations. Only continuous model elements are shown in the timeline. Timeline rows can be collapsed to view only the most relevant objects.

3.2 Evaluation Component

Similarly to the previous component, a paper prototype was developed prior to implementation. It was tested with the same four users from above. Major results showed that detections should be displayed separately and not solely as overlayed annotated rectangles in the original corresponding image. Furthermore, the assignment of the predefined options to the detections should apparently and intuitively made visible. Therefore, we decided to allow switching trough a given set of available options while continuously clicking on a detection result that appears to be slightly similar to interaction scheme within *iPhoto*.

The evaluation tools main workspace (see Fig. 2) is a section in the middle of the screen where the results are displayed in a rectangle. The number of pictures N displayed at a time can be changed by using the slider. The lefthand

Table 1. Results of our preliminary user study for the video annotation component in contrast to ViPER-GT on the task to annotate pedestrians in a video sequence of 500 frames.

Tool	Tester	Time (mm:ss)	Mean (Stddev)
Video annotation component	#1	10:58	12:33 (04:28)
	#2	08:05	
	#3	09:16	
	#4	16:02	
	#5	18:25	
ViPER-GT	#6	11:01	11:18 (01:53)
	#7	09:35	
	#8	13:19	

view contains thumbnails of the previous and next group of N image patches. In order to start the evaluation process, a user has to select a classification group that is derived from the the currently selected model. Consequently, the first N images are shown for annotation within the main window. A simple left click assigns the chosen label to the image patch. A right-mouse click can be used to pop-up the original image yielding the highlighted detection inside. The up and down arrows on the right side allow a group-wise navigation through the data collection. Besides, we decided to add a statistical overview about the distribution of the choices that have already been made on the bottom left side, indicated by the length of horizontal color bars and the numbers shown, respectively. Summarized information depict the selected model, the choice and the progress as well.

4 Preliminary Evaluation

A small preliminary user study with eight participants (three female, five male) was conducted for a performance comparison between our *video annotation component* (five testers) and ViPER-GT (three testers). Participants were given a brief introduction into the application and could familiarize themselves with the interaction by applying several manual annotations and linear interpolations in between, before starting the actual scenario, where they had to annotate three walking persons in a 500 frame video clip that is provided by ViPER-GT.

The mean annotation time of *video annotation component* exceeded that of ViPER-GT and therefore performed slightly weaker, but had the best completion time. However, annotation strategy and annotator motivation influenced the annotation time greatly and are mainly responsible for the diverging completion times (see Table 1). Especially motion turning points of objects or video segments of no, less or irregular motion patterns usually compromise linear interpolations and therefore reduce the accuracy and usability of straight interpolating techniques.

Ritter et al. [19] conducted a study that uses the principles of the second component in the interactive part of last years evaluation of *TRECVid Instance Search*[4] [3], whereas an instance can be roughly denoted as the occurrence of a specific object in a shot in the video footage. This main task consists in retrieving up to 1.000 shots of given instance within the large archive of 464 hours of the British soap opera *BBC East Enders*. This comprises 24 different categories each resembling up to four instances that were given by sample pictures together with a segmented binary image, a short text description and a sample clip file. The automated detectors from the authors retrieved 1.000 classified instance candidates for each category. Eight human annotators were given a period with a maximum of 15 min to validate the specific object instances in each category and eliminate false-positives. This work was achieved by using the proposed approach yielding to an average completion time of 11 min per category.

5 Summary and Future Work

We presented two components to speed up the process of object annotation in videos and the validation of large data collections with multiple thousands of previously classified object candidates. Our preliminary evaluations showed the potential usefulness of these approaches. However, larger evaluations with more participants could be helpful to draw more reliable conclusions for the video annotation component. Moreover, the presented framework could benefit from the integration of semi-automated methods for object tracking like block-matching within automatically detected shots. A combination with any well-known object recognition descriptors like SIFT operators within a bag-of-word approach [20] should prove effective in order to retrieve the object from other shots of the video footage. Both components improve the ground truth or the availability of object training samples that could at least be integrated as feedback components into the machine learning workflow from Storz et al. [6] in order to consecutively optimize already trained classifiers.

Acknowledgments. This work was partially accomplished within the projects *ValidAX – Validation of the AMOPA and XTRIEVAL* framework (VIP0044), and *localizeIT* (03IPT608X) funded by the *Federal Ministry of Education and Research* (BMBF, Germany) in the program of *InnoProfile Entrepreneurial Regions*, and the *Research Training Group CrossWorlds - Connecting Virtual and Real Social Worlds* (GRK1780), funded by the DFG (Deutsche Forschungsgesellschaft), Germany. We would like to thank Gerald Meier and Markus Keller for their contributions.

References

1. Dollár, P., Wojek, C., Schiele, D., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)

[4] http://www-nlpir.nist.gov/projects/tv2014/tv2014.html#ins, 20.02.2015.

2. Jain, V., Learned-Miller, E.: FDDB: a benchmark for face detection in unconstrained settings/University of Massachusetts, Amherst, Technical report (UM-CS-2010-009), 19 February 2015, pp. 11 (2010). http://vis-www.cs.umass.edu/fddb/fddb.pdf
3. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and TRECVid. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 321–330. ACM, New York (2006)
4. Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., Kompatsiaris, Y.: A survey of semantic image and video annotation tools. In: Paliouras, G., Spyropoulos, C.D., Tsatsaronis, G. (eds.) Multimedia Information Extraction. LNCS, vol. 6050, pp. 196–239. Springer, Heidelberg (2011)
5. Ritter, M., Eibl, M.: An extensible tool for the annotation of videos using segmentation and tracking. In: Marcus, A. (ed.) HCII 2011 and DUXU 2011, Part I. LNCS, vol. 6769, pp. 295–304. Springer, Heidelberg (2011)
6. Storz, M., Ritter, M., Manthey, R., Lietz, H., Eibl, M.: Annotate. Train. Evaluate. A unified tool for the analysis and visualization of workflows in machine learning applied to object detection. In: Kurosu, M. (ed.) HCII/HCI 2013, Part V. LNCS, vol. 8008, pp. 196–205. Springer, Heidelberg (2013)
7. Forsyth, D.A., Malik, J., Fleck, M.M., Greenspan, H., Leung, T., Belongie, S., Carson, C., Bregler, C.: Finding pictures of objects in large collections of images. In: Ponce, J., Hebert, M., Zisserman, A. (eds.) ECCV-WS 1996. LNCS, vol. 1144, pp. 335–360. Springer, Heidelberg (1996)
8. Hanbury, A.: A survey of methods for image annotation. J. Vis. Lang. Comput. **19**(5), 617–627 (2006)
9. Ahn, L., von Dabbish, L.: Labeling images with a computer game. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, pp. 319–326 (2004)
10. Yao, B., Yang, X., Zhu, S.-C.: Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 169–183. Springer, Heidelberg (2007)
11. Doermann, D., Mihalcik, D.: Tools and techniques for video performance evaluation. In: Proceedings of the 15th International Conference on Pattern Recognition, vol. 4, pp. 167–170 (2000)
12. Kipp, M.: Spatiotemporal coding in ANVIL. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (2008)
13. Brugman, H., Russel, A.: Annotating multimedia/multi-modal resources with ELAN. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (2004)
14. Hagedorn, J., Hailpern, J., Karahalios, K. G.: VCode and VData: illustrating a new framework for supporting the video annotation workflow. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 317–321. ACM (2008)
15. Aubert, O., Pri, Y.: Advene: active reading through hypervideo. In: Proceedings of the 16th ACM Conference on Hypertext and Hypermedia, pp. 235–244. ACM (2005)
16. Hosack, B.: VideoANT: extending online video annotation beyond content delivery. TechTrends **54**(3), 45–49 (2010)
17. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. Int. J. Comput. Vis. **101**(1), 184–204 (2013)

18. Schallauer, P., Ober, S., Neuschmied, H.: Efficient semantic video annotation by object and shot re-detection. In: Posters and Demos Session, 2nd International Conference on Semantic and Digital Media Technologies, Koblenz (2008)
19. Ritter, M., Heinzig, M., Herms, R., Kahl, S., Richter, D., Manthey, R., Eibl, M.: Technische Universitt Chemnitz at TRECVid Instane Search 2014. In: TRECVid Workshop 2014, 10–12 November 2014, Orlando, Florida, pp. 8 (2014). http://www-nlpir.nist.gov/projects/tvpubs/tv14.papers/tuc_mi.pdf. 01 March 2015
20. Jiang, W., Zhao, Z., Chen, Q., Zhao, J., Huang, Y., Zhao, X., Li, L., Zhao, Y., Su, F., Cai, A.: BUPT-MCPRL at TRECVid 2014 Instance Search Task. In: TRECVid Workshop 2014, 10–12 November 2014, Orlando, Florida, pp. 22 (2014). http://www-nlpir.nist.gov/projects/tvpubs/tv14.slides/bupt-mcprl.tv14.ins.slides.pdf. 01 March 2015