Keyword based Keyframe Extraction in Online Video Collections

Edoardo Ardizzone, Marco La Cascia and Giuseppe Mazzola
Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica (DICGIM), Università degli Studi di Palermo, Viale delle Scienze, bd. 6, Palermo, Italy

Keywords: Video Summarization, Keyframe Extraction, Automatic Speech Recognition, YouTube, Multimedia Collections.

Abstract: Keyframe extraction methods aim to find in a video sequence the most significant frames, according to specific criteria. In this paper we propose a new method to search, in a video database, for frames that are related to a given keyword, and to extract the best ones, according to a proposed quality factor. We first exploit a speech to text algorithm to extract automatic captions from all the video in a specific domain database. Then we select only those sequences (clips), whose captions include a given keyword, thus discarding a lot of information that is useless for our purposes. Each retrieved clip is then divided into shots, using a video segmentation method, that is based on the SURF descriptors and keypoints. The sentence of the caption is projected onto the segmented clip, and we select the shot that includes the input keyword. The selected shot is further inspected to find good quality and stable parts, and the frame which maximizes a quality metric is selected as the best and the most significant frame. We compare the proposed algorithm with another keyframe extraction method based on local features, in terms of Significance and Quality.

1 INTRODUCTION AND RELATED WORKS

The widespread diffusion of multimedia online collections has increased the need of software tools to index and to annotate the content of these multimedia data. Each month more than 1 billion unique users visit YouTube and over 6 billion hours of video are watched by users. Furthermore 100 hours of video are uploaded every minute, then YouTube is the world's largest video database, and it makes available to the users contents about any type of topic. According to a 2010 survey (Sysomos, 2010), Music is the most popular category with 30.7% of all analyzed videos, followed by Entertainment (14,6%) and People & Blogs (10,8%). Other popular categories are: News and Politics (6,7%), Sports (6%), Comedy (5,2%), Education (4,1%), Movies (3,6%), Animation (3,2%), HowTo (3,1%) and Science and Technology (2,9%).

In the very last years, YouTube users have the possibility to turn on automatic subtitles in the online videos they are watching. Not all the videos have this option but, for some specific domains, a lot of subtitled videos are available. In our work we are interested in studying these kinds of videos. In particular, our goal is to extract information within a specific domain of knowledge, and to summarize the retrieved content, according to the user input constraints.

Keyframe extraction is a technique which aims to extract the most significant frames from a video, in order to have a short summary that contains all the important information needed to understand the video content. The most common approach is to segment a video into sub-sequences (shots) and to select the most representative frame for each sub-sequence. A shot is defined as an uninterrupted sequence of frames acquired from a single camera, without cuts. A keyframe is the frame which can summarize, as best, the content of the shot.

A video sequence typically includes a lot of scene changes, and segmenting a video means finding the boundaries between different shots. Since there is a lot of literature about video segmentation (Hu et al., 2011), in this work we focus onto keyframe extraction techniques. Several features have been used in literature for the problem of keyframe extraction: color histogram in specific color spaces (D'Avila et al., 2011), object and camera motion based features (Yue et a., 2008), and edge information (Chan et al., 2011). Some newest

Ardizzone E., La Cascia M. and Mazzola G..
Keyword based Keyframe Extraction in Online Video Collections.
DOI: 10.5220/0005190001700177
In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2015), pages 170-177
ISBN: 978-989-758-077-2

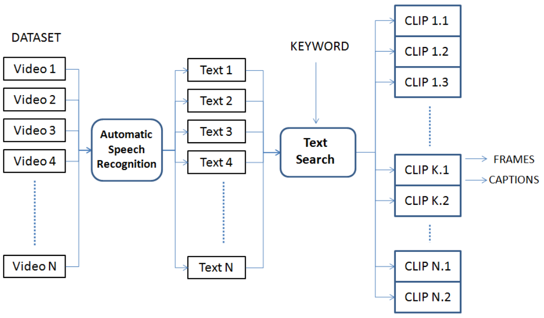


Figure 1: Keyword-Based Clip Selection. The videos in the domain dataset are processed by the ASR module. Automatic captions are extracted from the videos and a list of Clips, whose captions include the input keyword, are selected. Each Video may contain several Clips whose captions match the input keyword.

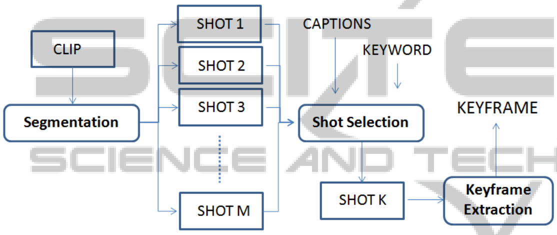


Figure 2: Clip Segmentation and Keyframe Extraction. Each clip is segmented to find the cuts and the shots. The most significant shot of the Clip is selected by considering the keyword position in the caption. The “best” keyframe, according to a quality criterion, is extracted from the most significant shot.

methods use local feature matching to segment videos and extract keyframes (Liu et al., 2009; Guan et al., 2013). Many approaches have been proposed for the problem. Threshold based methods (Wang and Luo, 2008) compare the differences between frames with a static or dynamic threshold. Methods based on statistical models (Jiang et al., 2008) build mathematical models to represent the boundaries of the shots. Clustering based methods (Chasanis et al., 2007) search for similar features in a shot. Some new methods (Wei at al. 2011) exploit also audio information.

In this paper we present a keyframe extraction method to analyze the content of online video collections. We are interested in finding in all the videos of a database the keyframes which are related to a given keyword, and that satisfy quality constraints. The paper is organized as follows: section 2 describes the proposed method; in section 3 we present our experimental results; in the conclusive section we discuss the possible applications and the future works.

2 PROPOSED METHOD

The goal of the proposed method is to search, in a large collection of videos, for frames that are significant to a specific concept, and which respect a visual quality constraint. The process may be subdivided into 4 steps: Clip Selection (fig.1), Clip Segmentation, Shot Selection and Keyframe Extraction (fig.2).

2.1 Clip Selection

Given a collection of videos regarding a specific domain (see experimental section for more details), and an input keyword, the first step is to select from the video database all the parts of the videos (clips) to which this keyword is somewhat related. For our purpose we considered only a single input keyword. We use an automatic transcription algorithm (the YouTube automatic captioning module) to extract the speech from the videos in the database, and to assign subtitles to each clip. We select the clips in which the given keyword is pronounced, namely those whose captions include the input keyword. A sequence, associated to a single line of a caption, is typically 2-5 seconds long. In this way we discard a lot of useless information, and we focus only on those parts of the videos which are significant for the keyword. A text based search is much faster than a visual search, and drastically reduces the execution time to retrieve the desired information. In this work we are not interested in the semantic analysis of the text, as captions are very noisy, and they will need to be processed to recover the underlying syntactic structure, before being able to study the semantic content. We adopted a simple binary search, with the purpose to verify the presence of the keyword in the captions. The output of this step is a list of clips, from different videos, and the related captions, which include the input keyword.

2.2 Clip Segmentation

The selected clips may include many scene changes, so the next step is to divide the clips into shots, by using a boundary frame detection algorithm. For this purpose we decided to use a local features based algorithm, which it is suitable for our goals and gives good results, even if it is slower than other state of the art methods. Note that our segmentation algorithm does not need to process the whole video, but only the selected clips, which are usually no more than 150 frames, and the process takes only few seconds per clip. We extract from each frame of

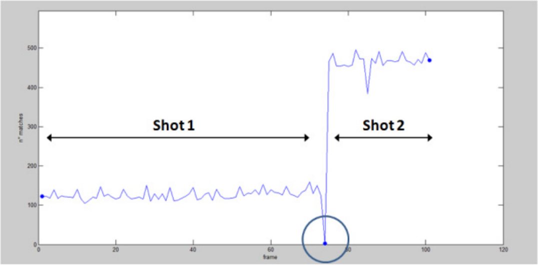


Figure 3: The video segmentation process. If there is a large displacement in the number of matches, the video sequence is split into two shots.

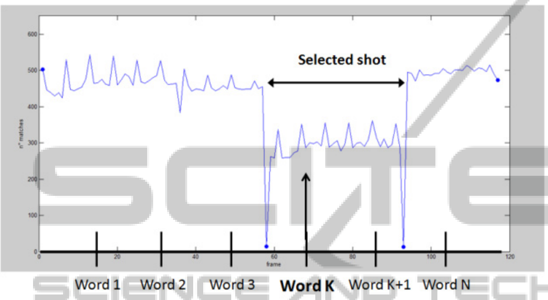


Figure 4: Shot Selection. Words are projected onto the video timeline, and the shot which includes the keyword (Word K), is selected.

the clip the SURF (Bay et al. 2006) points and descriptors, and compute the matches between two consecutive frames, by considering the minimum Euclidean distance between descriptors. We eliminate ambiguous matches if the ratio between the distances of a feature vector to its first and second nearest neighbours is above a threshold. For efficiency reasons, we decided to use the SURF algorithm instead of the SIFT one. In fact, as known, SURF is much faster than SIFT and, in case of very little differences between two images (as in the case of two consecutive frames in a video), has comparable results in many applications. Our clip is then represented by the vector of the number of matches between each frame and the next frame forward. To segment the clip we look for the frames that verify the condition:

$$\hat{F}(i) - F(i) > \alpha \cdot \hat{F}(i) \tag{1}$$

$$\hat{F}(i) = \frac{1}{k} \sum_{j=i-k}^{i.1} F(i) \tag{2}$$

where F(i) is the vector of features of the i-th frame, $\hat{F}(i)$ is a linear prediction of the actual value F, which uses the k previous frames to estimate the actual value, and α is a constant value less than 1 (α is experimentally set to 0.5). In practice, if there is a great variation of the actual value of the feature vector, with respect to the estimated one, we consider the frame as a boundary frame, that is an indication of a scene change (see fig. 3). The segmentation condition is imposed onto the displacement between the prediction and the actual value, instead of the absolute displacement, as we suppose that a change of scene is when there is a sudden "fall" in the number of matches. An abrupt increase is typically due to noise, and not to a change of scene (note that we compute the forward match between two consecutive frames). Finally each clip is segmented into different shots.

2.3 Shot Selection

Video segmentation techniques usually splits a video sequence into shots, to identify its cuts, and each shot is processed separately. In our study, once a clip (a part of the video whose subtitles include the input keyword) is split into shots, we need to use a criterion to select the most suitable shot, i.e. that is most relevant to the input keyword. For this purpose we made an assumption: we suppose that the word of interest is pronounced during the shot that is the most significant for that word. Therefore we equally subdivide the clip according to the number of words in the caption (fig. 4). We "project" the list of the words in the caption into the clip timeline, and we select as the most significant shot the one that includes the word of interest, that is the part of the clip in which the word is likely to have been said. For our purposes there is no need to further process the text of the subtitle, e.g. discarding stop-words, as we are interested in "when" the word is said, and not in its semantic meaning. The drawback of this assumption is that it does not take in account the syntactic structure of the captions. The word of interest can be the subject of the phrase, or the direct object, and this will influence its position in the caption. We plan to further investigate this problem in future works.

2.4 Keyframe Extraction

Once the shot is selected, in the last step of our method we extract a significant frame, the "keyframe", from the sequence of frames that are included in the shot. Many state of the art techniques select the first, the last, or the central frame as the most significant for a shot. Other local feature based methods (Liu et al., 2009) select the frame which has the highest number of keyponts, as it is the one which includes the most of the information. In our work we decided to consider another criterion: we search for the frame which has the highest "quality" between those into the inspected shot, that is the "maximally stable" frame. In particular we search for the frame which at the same time maximizes the average and minimizes the standard deviation of the number of matches with the other frames in a neighbourhood:

$$k = \arg\max_i \left[E_w(F(i)) - \beta \cdot \sigma_w(F(i)) \right] \quad (3)$$

where E_w and σ_w are, respectively, the average value and the standard deviation of the number of matches between frames in a window of size w, centred in the i-th frame, and β is a constant weight, experimentally set to 0.5. The selected keyframe will be extracted from an area in which there is "good" information (high number of matches) and in which there is a small variation of information (small standard deviation). In this way we avoid to extract motion blurred keyframes, which typically have less matches with the other neighbour frames and an high variability in the number of matches, or frames which are part of gradual transition. Note that we did not investigate separately gradual transition, mainly as our method do not select frames that are part of a transition, as they are "not good" quality frames. We observed in our tests that this hypothesis is verified in almost all the cases.

3 EXPERIMENTATION AND EVALUATION

To test our method, we needed to work with videos which have some particular features. First of all, we needed videos with automatic captions, as discussed in section 1. The second requirement is a high correlation between visual and audio information. The purpose of our algorithm is to find frames that are significant for a specific concept. When a speaker is talking about an issue that is not shown during the talk (or if it is shown before or after the talk), our method is not suited.

3.1 Dataset

We selected two specific domains that meet these requirements. The first domain dataset ('recipes') is made of instructional videos (or how-to videos) about cooking, a category of videos that is very popular on the Web. In these videos there is typically a speaker showing the ingredients, the tools, how to cook, etc. Instructional videos usually contain a lot of scene changes, going from a zoom on the objects in the table to a wide angle shot to focus on the speaker.

The second domain dataset ('wildlife') is made of the naturalistic documentaries about wildlife, that may be included in the "educational" category of videos. They are much longer than instructional videos, they have many static shots, and show an alternation of parts with people speaking and long sequences with no description. Therefore, naturalistic videos have very different features with respect to the "how-to" videos. We also tested our algorithm on sports videos but, as they do not meet the audio-visual correlation constraint, they are not suitable for our goals. Furthermore, audio in sports videos is typically very noisy, then the automatic transcription is often unreliable.

For each domain, we downloaded 100 videos from YouTube, in two languages: English and Italian. Not all the videos contain automatic subtitles. In particular we found 100 subtitled videos about "recipes" in English (69 in Italian) and 53 videos about "wildlife" in English (40 in Italian). Within the "recipes" domain, we selected several input keywords and we grouped them into three Classes: Ingredients ('water', 'sugar', 'butter', 'salt', 'oil'), Actions ('cut', 'mix'), and Tools ('pan', 'knife'). The equivalent Italian versions of these words have been used for the Italian language queries. We choose these words as the most frequent ones, within each Class, in the corpus of the words extracted from all the videos of the domain. For the 'wildlife' domain, using the same approach, we selected Animals ('fish', 'tiger', 'snake', 'birds'), Action ('kill'), and Environment ('forest', 'sea') words. Also in this case, we tested both the English and the Italian versions of these words.

3.2 Experimental Setup

For each query several clips, related to the input keyword, are extracted from the videos within the inspected dataset. For a clip of interest, we extract the central frame (in the rest of the paper indicated as "central"), the keyframe with our method (indicated as "proposed") and the keyframe extracted using a slightly modified version of the algorithm described in (Liu et al., 2009) (the "reference"). The reference method has been chosen as it use local features for the video segmentation and the keyframe extraction, as our method does. In particular, the method in (Liu et al., 2009) extracts SIFT keypoints and segments a video by thresholding a feature vector that is a combination of the number of keypoints of a frame, and the number matches between two consecutive frames. The keyframe of a shot is the one that maximizes the number of keypoints as, according to the authors, it is the one that contains the maximum information. In this paper we considered our implementation of (Liu et al., 2009) with two difference with respect to the original algorithm :

- we used SURF algorithm, rather than SIFT, for computing convenience;
- we introduced our shot selection step (see par. 2.3), to select the most significant shot of the clip (w.r.t. the keyword), in between the video segmentation and keyframe extraction steps of the reference algorithm, as it has no counterpart in the original algorithm.

Furthermore we took in account for comparison the central frame of the selected shot, but we decided not to show the results as we observed that they are very similar to those obtained with the reference method, in terms of the metrics described in section 3.3. We also studied the algorithm proposed in (Guan et al., 2013), but it is extremely slower than the chosen reference method, then we have not considered it for efficiency reasons.

3.3 Evaluation Metrics

In our tests we were not interested in evaluating separately the performance of the video segmentation part of the algorithm, but of the whole keyframe extraction process. Since it is impossible to define an objective metric to evaluate the performance of a keyframe extraction method, we adopted a subjective comparative approach. We asked to 5 testers to evaluate the "proposed" keyframe in comparison, separately, with the "central" and the "reference" keyframes, in terms of Significance and of Quality. A keyframe is more significant than another if its visual content is more representative for the input keyword. The Quality concept is highly subjective and involves many aspects, but a blurred or a motion blurred frame typically is considered a poor quality frame. With regard to the Significance evaluation, the testers have three options:

1. frame F1 is more significant than frame F2;
2. frame F2 is more significant than frame F1;
3. frames F1 and F2 are equally significant;

and the additional option:

4. none of the frames is significant.

If more than a half of the people select this last option, both the keyframes are labeled as insignificant. With regard to the Quality evaluation, the testers have three options:

1. frame F1 has better quality than frame F2;
2. frame F2 has better quality than frame F1;
3. frames F1 and F2 have the same quality.

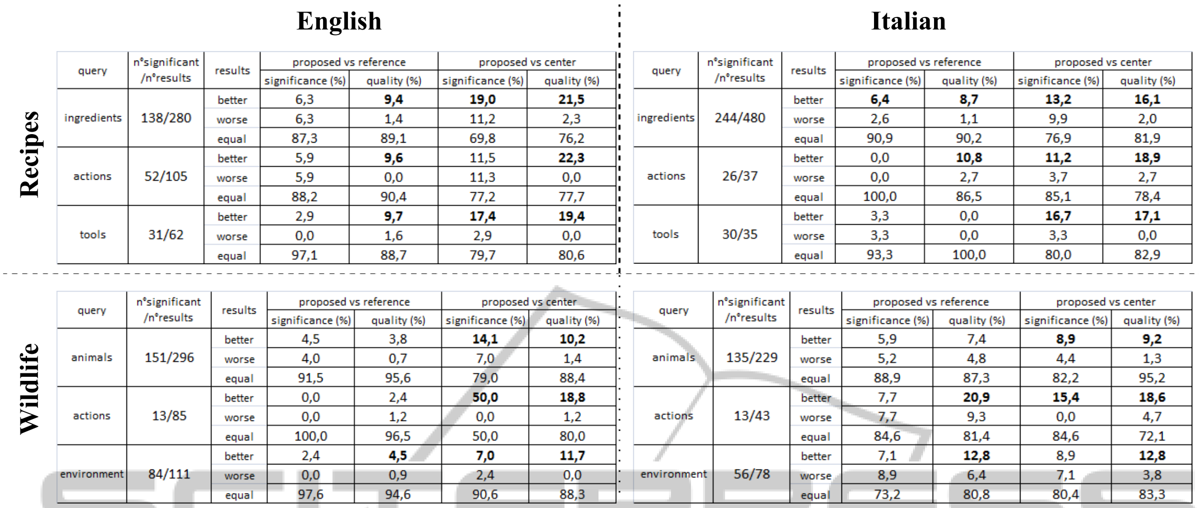
For each test the decisions are taken at majority. In case of draw between the options 1 and 2, the two frames are considered equally significant (or of the same quality). In case of draw between the options 1 (or 2) with option 3, the option 1 (2) wins.

3.4 Experimental Results

Table 1 shows the results obtained for the different domains and the different languages. The first result is that a lot of retrieved clips (about 50%) contain information that have been evaluated by the testers as not significant to the input keyword. This is typically the case of a person speaking of "something", without showing "something" and, in this case, the extracted frame is not relevant for the input keyword. In our tests we measured the Significance metric comparing only frames that are part of significant clips, while we compared all the retrieved frames in terms of Quality. Analyzing only the significant clips, we observed that in many cases all the methods give the same results. In fact, when the retrieved caption is related to a single shot sequence (the most frequent case), all the frames of the sequence have the same visual content and it is not very different, mainly in terms of Significance, to select a frame or another one. Moreover, our method and the reference one use a similar video segmentation algorithm, and the method the shot selection step is the same in both the algorithms. Thus the two methods select, in almost all cases, the same shot.

In terms of Quality, our method achieves better results, above all for the "recipes" domain. This means that the selection of the frame which have the highest informative content (the largest number of interest points) do not ensure the choice of a good quality frame, that is, to our opinion, a very important feature to better understand the content of an image. With respect to the "central" method, the improvement of our method are more evident, both in terms of Significance and Quality. In fact the selection of the "central" frame is almost a random choice, and there is no guarantee that the selected frame is correlated to the input keyword, nor that the frame is a "good quality" frame. We observed no relevant differences in the results obtained using Italian and English languages.

Table 1: Experimental results within the two domains in the two languages. The Significance metric is computed only on significant frames. Quality is compared between all the retrieved frames. We wrote in bold the most interesting results.



English

Recipes

query	n°significant /n°results	results	proposed vs reference		proposed vs center	
			significance (%)	quality (%)	significance (%)	quality (%)
ingredients	138/280	better	6,3	**9,4**	**19,0**	**21,5**
		worse	6,3	1,4	11,2	2,3
		equal	87,3	89,1	69,8	76,2
actions	52/105	better	5,9	**9,6**	11,5	**22,3**
		worse	5,9	0,0	11,3	0,0
		equal	88,2	90,4	77,2	77,7
tools	31/62	better	2,9	**9,7**	**17,4**	**19,4**
		worse	0,0	1,6	2,9	0,0
		equal	97,1	88,7	79,7	80,6

Wildlife

query	n°significant /n°results	results	proposed vs reference		proposed vs center	
			significance (%)	quality (%)	significance (%)	quality (%)
animals	151/296	better	4,5	3,8	**14,1**	**10,2**
		worse	4,0	0,7	7,0	1,4
		equal	91,5	95,6	79,0	88,4
actions	13/85	better	0,0	2,4	**50,0**	**18,8**
		worse	0,0	1,2	0,0	1,2
		equal	100,0	96,5	50,0	80,0
environment	84/111	better	2,4	**4,5**	**7,0**	**11,7**
		worse	0,0	0,9	2,4	0,0
		equal	97,6	94,6	90,6	88,3

Italian

Recipes

query	n°significant /n°results	results	proposed vs reference		proposed vs center	
			significance (%)	quality (%)	significance (%)	quality (%)
ingredients	244/480	better	**6,4**	**8,7**	**13,2**	**16,1**
		worse	2,6	1,1	9,9	2,0
		equal	90,9	90,2	76,9	81,9
actions	26/37	better	0,0	**10,8**	**11,2**	**18,9**
		worse	0,0	2,7	3,7	2,7
		equal	100,0	86,5	85,1	78,4
tools	30/35	better	3,3	0,0	**16,7**	**17,1**
		worse	3,3	0,0	3,3	0,0
		equal	93,3	100,0	80,0	82,9

Wildlife

query	n°significant /n°results	results	proposed vs reference		proposed vs center	
			significance (%)	quality (%)	significance (%)	quality (%)
animals	135/229	better	5,9	7,4	**8,9**	**9,2**
		worse	5,2	4,8	4,4	1,3
		equal	88,9	87,3	82,2	95,2
actions	13/43	better	7,7	**20,9**	**15,4**	**18,6**
		worse	7,7	9,3	0,0	4,7
		equal	84,6	81,4	84,6	72,1
environment	56/78	better	7,1	**12,8**	8,9	**12,8**
		worse	8,9	6,4	7,1	3,8
		equal	73,2	80,8	80,4	83,3

In terms of the efficiency, our algorithm takes few seconds per clip, spent mainly for the video segmentation step. In fact, only the clips of the videos in the dataset that are related to the input concept are analyzed, instead of processing hours and hours of videos, drastically reducing the execution time.

Fig. 5 shows some visual examples of the obtained results within the "recipes" (first three rows) and wildlife (last two rows) domains. The first row shows the results with the keyword "milk" within the "recipes" domain. In this case the "proposed" keyframe is the only one which contains something related to word "milk" and moreover the "reference" one is affected by motion blurring. In the second row the three frames have more or less the same Significance with respect to the word "butter", but the "proposed" keyframe is the best in terms of Quality. Regarding the keyword "cut" (third row), the three extracted keyframes have more or less the same Quality (the "reference" is slightly affected by blurring), but the "proposed" frame is the most significant for the input keyword. For the "wildlife" domain and the keyword "snake", the "proposed" keyframe is the only one which represents a snake, and all three frames are slightly affected by blurring. In the last case ("wildlife" domain and "kill" keyword) the "reference" and the "proposed" methods extract almost the same keyframe, which is significant for the input keyword, while the "central" keyframe is an irrelevant and poor quality image.

4 CONCLUSIONS

Extracting keyframes that are related to a given subject, from large video databases, may be a useful tool for many applications. First of all, it can help users in retrieving those parts of the videos that are related to a desired subject, without directly inspecting all the videos, saving a lot of work and time. Furthermore, the presentation of "good quality" frames may help the users in understanding the content of the retrieved videos, as poor quality frames, e.g. motion blurred ones, may hide important information. If the goal is to find visual examples of a desired subject, the main drawback of the proposed method is that several retrieved clips, whose caption contains the desired word, do not include frames that visually represent the subject. This is sometimes due to incorrect automatic transcriptions, but in most cases the clips contains people "speaking about" a subject, without "showing" it.

To further filter out the retrieved information, we plan in future works to exploit visual information, e.g. visual models, of the given subject to be compared with the extracted keyframes. We also plan to exploit our algorithm, which aim to extract visual examples of a specific subject, to find the same subject into not annotated videos, within a specific domain, using also temporal information. The combination of textual and visual information may be also used to extract the semantic structure of the video, exploiting, for example, domain specific ontologies.

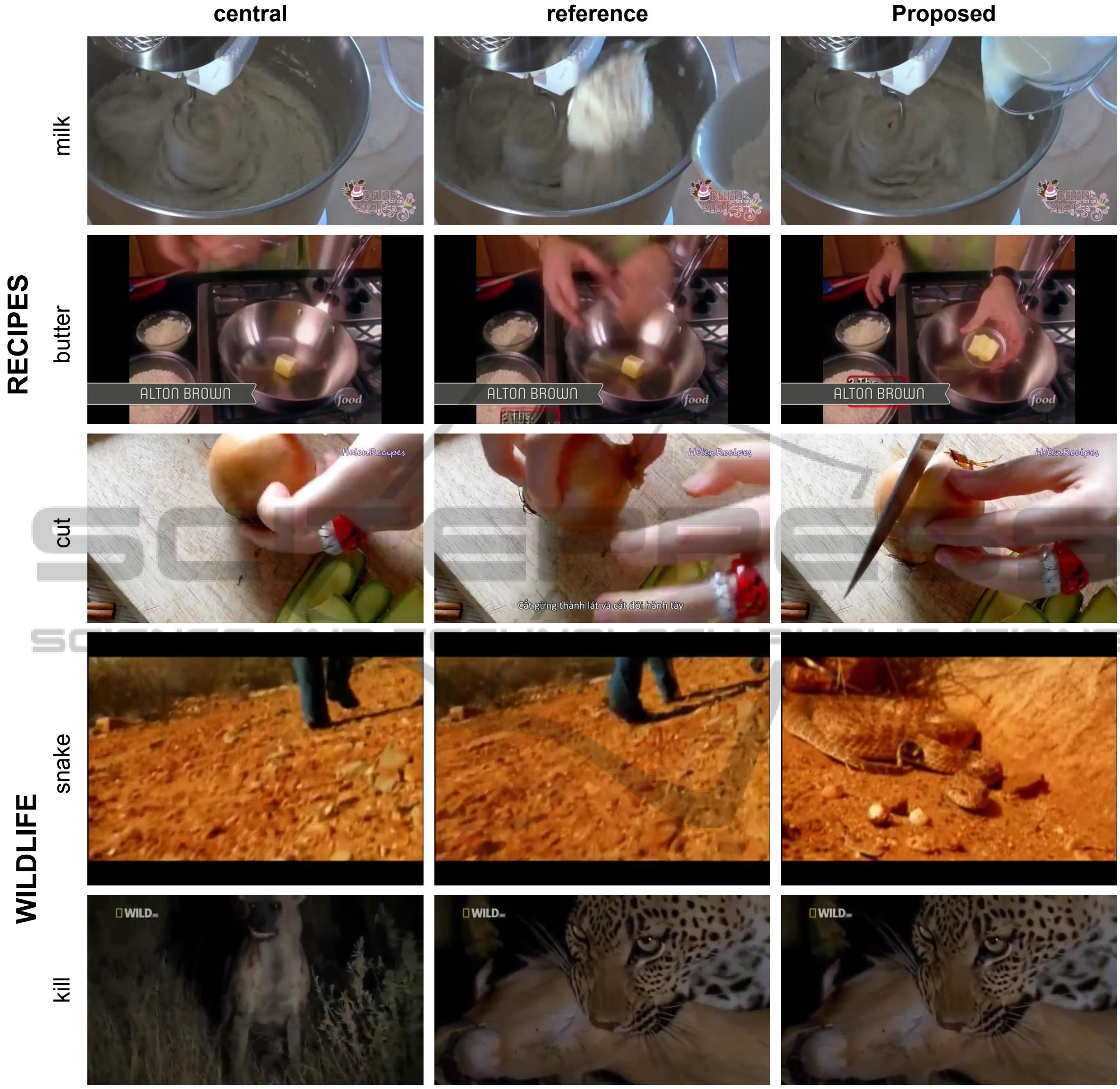


Figure 5: Some visual examples of the obtained results within the different domains. Keyframes extracted with the “central” (left column), “reference” (central), and “proposed” (right) methods.

REFERENCES

Bay, H., Tuytelaars, T., Van Gool, L. 2006. Surf: Speeded Up Robust Features. In *Proceedings of European Conference on Computer Vision* ECCV, 404-417.

Chan, P. P. K., et al., 2011. A Novel Method to Reduce Redundancy in Adaptive Threshold Clustering Keyframe Extraction Systems. *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, Guilin, vol. 4, pp.1637–1642.

Chasanis, V., Likas, A., and Galatsanos, N., 2007. Scene Detection in Videos Using Shot Clustering and Symbolic Sequence Segmentation. *IEEE 9th Workshop on Multimedia Signal Processing*, pp. 187-190.

D’Avila, S. E., et al., 2011. VSUMM: a Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern Recognition Letters* 32 (1) () 56–68.Guan, G., Wang, Z., Lu, S., Deng, J. D., and, D. D., 2013. Keypoint-Based Keyframe Selection. *Circuits and Systems for Video Technology, IEEE Transactions on*. 23(4), 729-734.

Hu, W., et al.., 2011. A Survey on Visual Content-Based Video Indexing and Retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol.41, no.6, pp.797-819.

Jiang, X. H., Sun, T. F., Li, J. H., and Chen, X., 2008. A Novel Shot Edge Detection Algorithm Based on Chi-

Square Histogram and Macro-Block Statistics. *Proc. of International Symposium on Information Science and Engineering*, vol. 2, pp. 604-607.

Liu, G., Wen, X., Zheng, W., and He, P., 2009. Shot Boundary Detection and Keyframe Extraction Based on Scale Invariant Feature Transform. In *Proceedings of the 2009 Eigth IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*, pp. 1126-1130.

Sysomos, 2010. http://www.sysomos.com/reports/ YouTube.

Wang, J. Y., and Luo. W., 2008. A Self-Adapting Dual-Threshold Method for Video Shot Transition Detection. *IEEE International Conference on Networking, Sensing and Control*, pp. 704-707.

Wei, J., Cotton, C., and Loui, A.C., 2011. Automatic consumer video summarization by audio and visual analysis. *Multimedia and Expo (ICME), 2011* I*EEE International Conference on*, pp (1-6).

Yue, G., Wei-Bo, W., Jun-Hai, Y., 2008. A Video Summarization Tool using Two-Level Redundancy Detection for Personal Video Recorders. *IEEE Transactions on Consumer Electronics*, 54, 2, 521–526.