Chapter 1
Video fragmentation and reverse search on the Web

Evlampios Apostolidis, Konstantinos Apostolidis, Ioannis Patras, Vasileios Mezaris

Abstract This chapter is focused on methods and tools for video fragmentation and reverse search on the Web. These technologies can assist journalists when they are dealing with fake news - which nowadays are rapidly spread via social media platforms - that rely on the reuse of a previously posted video from a past event with the intention to mislead the viewers about a contemporary event. The fragmentation of a video into visually and temporally coherent parts and the extraction of a representative keyframe for each defined fragment enables the provision of a complete and concise keyframe-based summary of the video. Contrary to straightforward approaches that sample video frames with a constant step, the generated summary through video fragmentation and keyframe extraction is considerably more effective for discovering the video content and performing a fragment-level search for the video on the Web. This chapter starts by explaining the nature and characteristics of this type of reuse-based fake news in its introductory part, and continues with an overview of existing approaches for temporal fragmentation of single-shot videos into sub-shots (the most appropriate level of temporal granularity when dealing with user-generated videos) and tools for performing reverse search of a video on the Web. Subsequently it describes two state-of-the-art methods for video sub-

Evlampios Apostolidis
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece and School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: `apostolid@iti.gr`

Konstantinos Apostolidis
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: `kapost@iti.gr`

Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: `i.patras@qmul.ac.uk`

Vasileios Mezaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: `bmezaris@iti.gr`

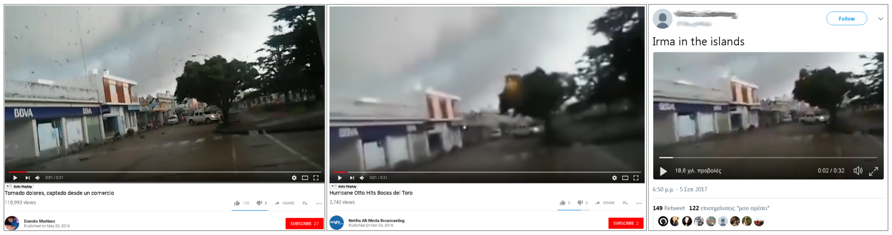
shot fragmentation - one relying on the assessment of the visual coherence over sequences of frames, and another one that is based on the identification of camera activity during the video recording - and presents the InVID web application that enables the fine-grained (at the fragment-level) reverse search for near-duplicates of a given video on the Web. In the sequel the chapter reports the findings of a series of experimental evaluations regarding the efficiency of the above mentioned technologies, which indicate their competence to generate a concise and complete keyframe-based summary of the video content, and the use of this fragment-level representation for fine-grained reverse video search on the Web. Finally, it draws conclusions about the effectiveness of the presented technologies and outlines our future plans for further advancing them.

1.1 Introduction

The recent advances in video capturing technology made possible the embedding of powerful, high-resolution video sensors into portable devices, such as camcorders, digital cameras, tablets and smartphones. Most of these technologies now offer network connectivity and file sharing functionalities. The latter, combined with the rise and widespread use of social networks (such as Facebook, Twitter, Instagram) and video sharing platforms (such as YouTube, Vimeo, DailyMotion) resulted in a enormous increase in the number of videos captured and shared online by amateur users on a daily basis. These user-generated videos (UGVs) can nowadays be recorded at any time and place using smartphones, tablets and a variety of video cameras (such as GoPro action cameras) that can be attached to sticks, body parts or even drones. The ubiquitous use of video capturing devices supported by the convenience of the user to share videos through social networks and video sharing platforms, leads to a wealth of online available UGVs.

Over the last years these online shared UGVs are, in many cases, the only evidence of a breaking or evolving story. The sudden and unexpected appearance of these events make their timely coverage by news or media organization impossible. However, the existence (in most cases) of eyewitnesses capturing the story with their smartphones and instantly sharing the recorded video (even live, i.e. during its recording) via social networks, makes the UGV the only and highly valuable source of information about the breaking event. In this newly formed technological environment that facilitates information diffusion through a variety of social media platforms, journalists and investigators alike are increasingly turning to these platforms to find media recordings of events. Newsrooms in TV stations and online news platforms make use of video to illustrate and report on news events, and since professional journalists are not always at the scene of a breaking or evolving story (as mentioned above), it is the content shared by users that can be used for reporting the story. Nevertheless, the rise of social media as a news source has also seen a rise in fake news, i.e. the spread of deliberate misinformation or disinformation on these platforms. Based on this unfortunate fact, the online shared user-generated content comes into question and people's trust in journalism is severely shaken.

One type of fakes, probably the easiest to do and thus one of most commonly found by journalists, relies on the reuse of a video from an earlier event with the claim that it shows a contemporary event. An example of such a fake is depicted in Fig. 1.1. In this figure, the image on the left is a screenshot of a video showing a hurricane that strikes in Dolores, Uruguay on May 29 2016, the image on the middle is a screenshot of the same video with the claim that is shows Hurricane Otto that strikes in Bocas del Toro, Panama on November 24 2016, and the image on the right is a screenshot of a tweet that uses the same video with the claim that is shows the activity of Hurricane Irma in the islands near the United States on September 9 2017.



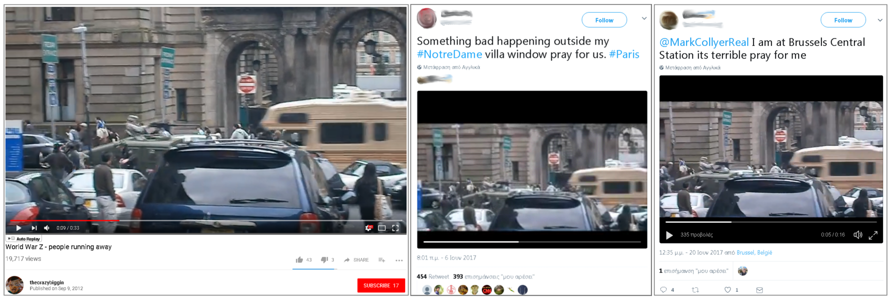
Tornado dolores, captado desde un comercio
118,993 views
Ernesto Martinez
Published on May 29, 2016

Hurricane Otto Hits Bocas del Toro
2,742 views
Net4ia Alt-Media Broadcasting
Published on Nov 24, 2016

Irma in the islands
18,6 χιλ. προβολές
6:50 μ.μ. - 5 Σεπ 2017
149 Retweet 122 επισημάνσεις "μου αρέσει"

Fig. 1.1: Example of a fake news based on the reuse of a video from a hurricane in Uruguay (image on the left) to deliberately mislead people about the strike of hurricane Otto in Panama (image in the middle) and the strike of hurricane Irma in the US islands (image on the right).

The identification and debunking of such fakes requires the detection of the original video through the search for prior occurrences of this video (or parts of it) on the Web. Early approaches for performing this task were based on manually taking screenshots of the video in the player and uploading these images for performing reverse image search using the corresponding functionality of popular Web search engines (e.g. Google search). This process can be highly laborious and time-demanding, while its efficiency depends on a limited set of manually taken screenshots of the video. However, the in-time identification of media posted online, which (claim to) illustrate a (breaking) news event is for many journalists the foremost challenge in order to meet deadlines to publish a news story online or fill a news broadcast with content. The time needed for extensive and effective search regarding the posted video, in combination with the lack of expertise by many journalists and the time-pressure to publish the story, can seriously affect the credibility of the published news item. And the publication or re-publication of fake news can significantly harm the reliability of the entire news organization. An example of miss-verification of a fake video by an Italian news organization is presented in Fig. 1.2. A video from the filming of the "World War Z" movie (left part of Fig. 1.2) was used in a tweet claiming to show a Hummer attack against police in Notre-Dame, Paris, France on June 6 2017 (middle part of Fig. 1.2) and another tweet claiming to show an attack at Gare Centrale, Brussels, Belgium two weeks later (right part of Fig. 1.2). The fake tweet about the Paris attack was used in a new item published by the aforementioned news organization, causing a strong defeat in its trustworthiness.



World War Z - people running away

Something bad happening outside my #NotreDame villa window pray for us. #Paris

@MarkCollyerReal I am at Brussels Central Station its terrible pray for me

Fig. 1.2: Example of a fake news based on the reuse of a video from the filming of the “World War Z” movie (image on the left) to deliberately mislead people about a Hummer attack attack in Notre-Dame, Paris (image in the middle) and at Gare Centrale in Brussels (image on the right).

Several tools that enable the identification of near-duplicates of a video on the Web have been developed over the last years, a fact that indicates the usefulness and applicability of this process by journalists and members of the media verification community. Nevertheless, the existing solutions (presented in details in Section 1.2.2) exhibit several limitations that restrict the effectiveness of the video reverse search task. In particular, some of these solutions rely on a limited set of video thumbnails provided by the video sharing platform (e.g. the YouTube DataViewer of Amnesty International[1] and the Custom Reverse Image Search of IntelTechniques[2]). Other technologies demand the extraction of video frames for performing reverse image search (e.g. the TinEye search engine[3] and the Karma Decay[4] web application). A number of tools enable this reverse search on closed collections of videos, that significantly limit the boundaries of investigation (e.g. the Berify[5], the RevIMG[6] and the Videntifier[7] platforms). Last but not least, a commonality among the aforementioned technologies is that none of them supports the analysis of locally stored videos.

[1] https://citizenevidence.amnestyusa.org/

[2] https://inteltechniques.com/osint/reverse.video.html

[3] https://tineye.com/

[4] http://karmadecay.com/

[5] https://berify.com/

[6] http://www.revimg.com/

[7] http://www.videntifier.com

Aiming to offer a more effective approach for reverse video search on the Web, in InVID we developed: a) an algorithm for temporal fragmentation of (single-shot) UGVs into sub-shots (presented in Section 1.3.1.1), and b) a web application that integrates this algorithm and makes possible the time-efficient and at the fragment-level reverse search for near-duplicates of a given video on the Web (described in Section 1.3.2. The developed algorithm allows the identification of visually and temporally coherent parts of the processed video, and the extraction of a dynamic number of keyframes in a manner that secures a complete and concise representation of the defined - visually discrete - parts of the video. Moreover, the compatibility of the web application with several video sharing platforms and social networks is further extended by the ability to directly process videos that are locally stored in the user's machine. In a nutshell, our complete technology assists users to quickly discover the temporal structure of the video, extract detailed information about the video content and use this data in their reverse video search queries.

In the following, Section 1.2 discusses the current state of the art on methods for video sub-shot fragmentation (Section 1.2.1) and tools for reverse video search on the Web (Section 1.2.2. Then Section 1.3 is dedicated to the presentation of two advanced approaches for video sub-shot fragmentation - the InVID method that relies on the visual resemblance of the video content (see Section 1.3.1.1) and another algorithm that is based on the extraction of motion information (see Section 1.3.1.2) - and the description of the InVID web application for reverse video search on the Web (see Section 1.3.2). Subsequently, Section 1.4 reports the extracted findings regarding the performance of the aforementioned methods (see Section 1.4.1) and tool (see Section 1.4.2), while the last Section 1.5 concludes the document and presents our future plans on this research area.

1.2 Related Work

This part presents the related work, both in terms of methods for temporal fragmentation of uninterruptedly captured (i.e. single-shot) videos into sub-shots (Section 1.2.1) and tools for finding near-duplicates of a given video on the Web (Section 1.2.2).

1.2.1 Video Fragmentation

A variety of methods dealing with the temporal fragmentation of single-shot videos have been proposed over the last couple of decades. Most of them are related to approaches for video summarization and keyframe selection (e.g. [21, 9, 29, 15]), some focus on the analysis of egocentric or wearable videos (e.g. [27, 41, 19]), others aim to address the need for detecting duplicates of videos (e.g. [8]), a number of them is related to the indexing and annotation of personal videos (e.g. [28]), while there is a group of methods that targeted the indexing and summarization of rushes video (e.g. [12, 25, 4, 36]). The majority of the suggested approaches can be grouped in two main classes of methodologies.

The techniques of the first class consider a sub-shot as an uninterrupted sequence of frames within a shot that only have a small variation in visual content. Based on this assumption, they try to define sub-shots by assessing the visual similarity of consecutive or neighboring video frames. A rather straightforward approach that evaluates frames' similarity using colour histograms and the x^2 test was described in [36], while a method that detects sub-shots of a video by assessing the visual dissimilarity of frames lying within a sliding temporal window using 16-bin HSV histograms (denoted as "Eurecom fragmentation") was reported in [12]. Instead of using HSV histograms, the video fragmentation and keyframe selection approach described in [39], represents the visual content of each video frame with the help of the Discrete Cosine Transform (DCT) and assesses the visual similarity of neighboring video frames based on the cosine similarity. The generated frame-level sequence of similarity scores is then post-processed and the sequences of frames that exhibit visual and temporal coherence form the sub-shots of the video. A different approach [4] estimates the grid-level dissimilarity between pairs of frames and fragments a video by observing that the cumulative difference in the visual content of subsequent frames indicates gradual change within a sub-shot; a similar approach was presented in [25]. The method of [34] estimates the brightness, contrast, camera and object motion of each video frame using YUV histograms and optical flow vectors, and defines sub-shot boundaries by analysing the extracted features through a coherence discontinuity detection mechanism on groups of frames within a sliding window.

The methods of the second class fragment a video shot into sub-shots based on the rationale that each sub-shot corresponds to a different action of the camera during the video recording. Hence, these approaches aim to detect different types of camera activity over sequences of frames, and define these frame sequences as the different sub-shots of the video. An early, MPEG-2 compatible, algorithm that detects basic camera operations by fitting the motion vectors of the MPEG stream into a 2D affine model, was presented in [22]. Another approach that exploits the same motion vectors and estimates the camera motion via a multi-resolution scheme was proposed in [13]. More recently, the estimation of the affinity between pairs of frames for motion detection and categorization was a core idea for many other techniques. Some of them use the motion vectors of the MPEG-2 stream (e.g. [29]), while others compute the parameters of a 3×3 affine model by extracting and matching local descriptors [9] or feature points [33]. The dominant motion transformation between a pair of frames is then estimated by comparing the computed parameters against pre-defined models. [10] studies several approaches for optical flow field calculation, that include the matching of local descriptors (i.e. SIFT [26], SURF [5]) based on a variety of block matching algorithms, and the use of the Pyramidal Lucas Kanade (PLK) algorithm [7]. The more recently introduced algorithm of [3] performs a lightweight computation of spatio-temporal optical flow over sequences of frames and compares the frame-level motion distribution against pre-defined motion models. The extracted motion information is then used to detect (and categorize) a number of different video recording actions (which relate to camera movement or focal distance change) and the frame sequences that temporally correlate with each identified action are considered as the video sub-shots. Contrary to the use of experimentally-defined thresholds for categorizing the detected camera motion, [18] describes a generic approach for motion-based video parsing that estimates the affine motion parameters, either based on motion vectors of the MPEG-2 stream or by applying a frame-to-frame image registration process, factorizes their values via Singular Value Decomposition (SVD) and imports them into three multi-class Support Vector Machines (SVMs) to recognize the camera motion type and direction between successive video frames. A variation of this approach [1], identifies changes in the "camera view" by estimating a simplified three-parameter global camera motion model using the Integral Template Matching algorithm [24]. Then, trained SVMs classify the camera motion of each frame, and neighboring frames with the same type of camera motion are grouped together forming a sub-shot. Another threshold-less approach [19] aims to identify specific activities in egocentric videos using hierarchical Hidden Markov Models (HMM), while the algorithm of [15] combines the concept of "camera views" and the use of HMM for performing camera motion-based fragmentation of UGVs. Finally, a study on different approaches for motion estimation was presented in [6].

Further to the aforementioned two general classes of methodologies, other approaches have been also proposed. The early approach from [23] and the more recently proposed algorithm from [21] exploit motion vector information from the compressed video stream at the macro-block level. The methods in [16] and [8] extract several descriptors from the video frames (e.g. color histograms and motion features) and subdivide each shot into sub-shots by clustering its frames into an appropriately determined number of clusters with the help of the c-means and k-means clustering algorithms, respectively. A couple of techniques, presented in [40, 11], utilize data from auxiliary camera sensors (e.g. GPS, gyroscope and accelerometers) to identify the camera motion type for every video sub-shot or a group of events in UGVs. On a slightly different context, algorithms capable to analyze egocentric or wearable videos were discussed in [27] and [41]. Last but not least, the variety of introduced algorithms for video sub-shot fragmentation includes approaches based on the extraction and processing of 3D spatio-temporal slices (e.g. [32, 31]), and statistical analysis (e.g. [30, 35, 17]), while a comparative study evaluating the performance of different approaches for sub-shot fragmentation can be found in [10].

1.2.2 Reverse Video Search on the Web

Nowadays there is a plethora of tools that support the search and retrieval of near duplicates of an image or video on the Web. The latter indicates the popularity and attractiveness of image/video-based search and highlights the usefulness of the visual-content-based searching procedure for performing several media asset management tasks, including the assessment of the originality and authenticity of a given video.

One of the earliest (and most known among journalists) technologies is the YouTube DataViewer of Amnesty International[8] which enables the users to find near duplicates of a YouTube video by performing a reverse image search using the YouTube-extracted video thumbnails. Another web-based application that extends this functionality to additional video sharing platforms is the Custom Reverse Image Search of IntelTechniques[9]. The latter allows the thumbnail-based reverse search of videos coming from Vimeo, Facebook, Vine, Instagram, LiveLeak and Backpage, and exploits the image search functionality of several search engines, containing Google, Tineye, Yandex, Bing, and Baidu. Nevertheless, both of these solutions perform reverse video search based on a limited set of (usually) randomly selected video keyframes/thumbnails that have been associated to the video. This fact introduces the risk of excluding parts of the video that could enhance the reverse search or be of particular interest to the user, or even worse, to base the reverse search on thumbnails that are completely irrelevant to the video and have been deliberately selected for click-bait purposes. In addition, the search is supported only for videos available online, thus making impossible the reverse search of a video stored in the user's machine.

Another (pre-existing) solution that can partially support the retrieval of near duplicates of a video is the TinEye search engine[10], which enables the online search and retrieval of a given image. The advantage of this tool is that it offers a (paid) API to anyone who wishes to perform image search requests in a more automated way instead of providing every time the URL of the image file or uploading a local copy of the file on the TinEye web application. The limitation of this technology when trying to find near duplicates of a given video is that it requires the extraction of video frames that should be used as query images, a process which implies an overhead to the overall procedure. A variation of this platform, with significantly more restricted functionalities though, is the Karma Decay[11] web application which allows to perform reverse image search on Reddit.com. Last but not least, three recently developed platforms that assist the detection and retrieval of images and videos are the Berify, the RevIMG and the Videntifier. Berify[12] is a paid service that, according to its developers, offers functionalities for image-driven search of online available images and videos; updates of the search results are checked and forwarded to its users on a predefined basis. RevIMG[13] is another non-free solution that offers more unique functionalities, enabling the user to specify and use a portion of an image to search. However, the reverse search is performed only within closed

[8] https://citizenevidence.amnestyusa.org/

[9] https://inteltechniques.com/osint/reverse.video.html

[10] https://tineye.com/

[11] http://karmadecay.com/

[12] https://berify.com/

[13] http://www.revimg.com/

collections of images. Videntifier[14] is a visual search engine which can be used for the retrieval of a given image or video stream (even after being modified), but similar to RevIMG, the identification of a near duplicate relies on the matching of the given media item against a closed reference collection of video content.

1.3 State-of-the-art Techniques and Tools

1.3.1 Video Fragmentation

This section describes two different approaches for the fragmentation of single-shot videos into sub-shots; one that relies on the assessment of the visual resemblance between neighboring frames of the video (presented in 1.3.1.1), and another one that is based on the detection of motion which corresponds to different camera activities during the recording of the video (explained in 1.3.1.2). In terms of the utilised strategy for defining the different segments of the video, these methods cover a major portion of the techniques reported in the literature part of the chapter (section 1.2.1). Hence, the following subsections allow the reader to understand how these different approaches tackle the problem of video sub-shot fragmentation, identify the pros and cons of each approach, and get a concrete view about the efficiency of each method based on the evaluation outcomes reported in section 1.4.1.

1.3.1.1 Video Sub-shot Fragmentation based on the Visual Coherence

This algorithm (described in [39]) belongs to the first class of methods presented in Section 1.2.1. It decomposes a single-shot video into sub-shots based on the detection of visually and temporally coherent parts of the video, i.e. sequences of frames having only a small and contiguous variation in their visual content. This detection relies on the comparison and assessment of the visual similarity of neighboring frames of the video. For this purpose, the visual content of each video frame is represented with the help of the Discrete Cosine Transform (DCT), which is similar to the applied transformation when extracting the MPEG-7 Color Layout Descriptor [20]. More specifically, the pipeline for computing the DCT-based representation of a video frame is illustrated in Fig. 1.3 and contains the following steps:

- the video frame is initially resized to $m \times m$ dimensions for increasing the resilience of the analysis against changes in the image aspect ratio and size (step 1 in Fig. 1.3);
- the resized image is represented as a sum of cosine functions oscillating at different frequencies via a two-dimensional DCT (step 2 in Fig. 1.3), which results in an $m \times m$ matrix (for illustration purposes, $m = 8$ in Fig. 1.3) where the top

[14] http://www.videntifier.com

left element corresponds to the DC coefficient (zero-frequency) and every other element moving from left to right and from top to bottom corresponds to an increase in the horizontal and vertical frequency by a half cycle, respectively;

- the top left $r \times r$ part ($r < m$) of the computed matrix (for illustration purposes, $r = 3$ in Fig. 1.3) is kept, thus maintaining the most of the visual information that tends to be concatenated in a few low-frequency components of DCT, while high-frequency coefficients that store information related to the visual details of the image are discarded (step 3 in Fig. 1.3);
- a matrix reshaping process is applied to piece together the rows of the extracted $r \times r$ sub-matrix to a single row vector (step 4 in Fig. 1.3);
- the first element of this vector, which corresponds to the DC coefficient, is removed (step 5 in Fig. 1.3), forming a row vector of size $r^2 - 1$ that represents the image.

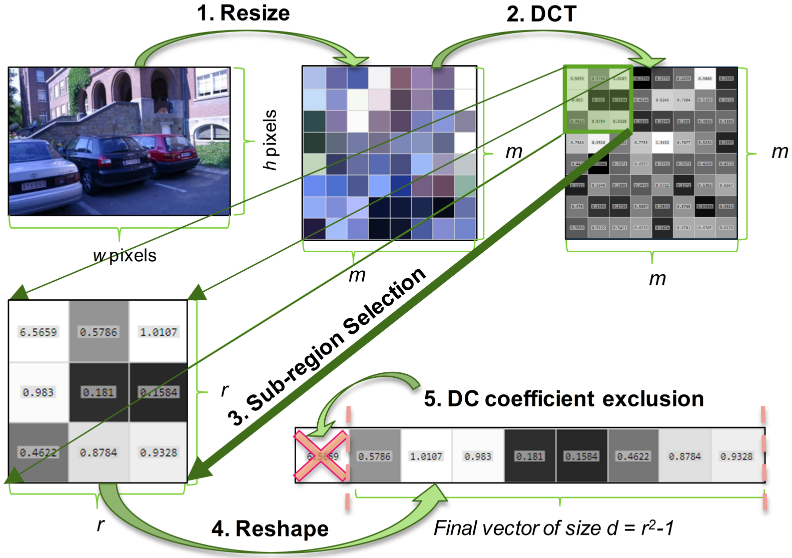


Fig. 1.3: The steps of the applied analysis for extracting the DCT-based representation of the visual content of each processed video frame.

Having extracted the DCT-based representation of the video frames, the visual similarity between a pair of frames is then estimated by computing the cosine similarity of their descriptor vectors. More specifically, given a pair of video frames F_i and F_j with descriptor vectors D_i and D_j respectively, their visual resemblance $V_{i,j}$ is calculated by: $V_{i,j} = \frac{D_i \cdot D_j}{\|D_i\| \|D_j\|}$, where $\cdot$ denotes the dot product of the descriptor vectors and $\|\|$ denotes their Euclidean norm. Nevertheless, a pair of subsequent video frames - even in the case of a video with the typical frame-rate of 30fps - usually exhibits high visual similarity; and this similarity gets more and more significant for videos of higher frame-rates that users are allowed to capture with modern smartphones or other devices (such as GoPro cameras which support video recoding up to 240fps). Driven by this observation, the algorithm does not apply the aforementioned pair-wise similarity estimation on every pair of consecutive video frames, but only for neigboring frames selected via a frame-sampling strategy which keeps 3 equally distant frames per second.

The analysis of the entire set of selected video frames results in a series of similarity scores, which is then post-processed in order to identify visually coherent fragments with gradually changing visual content (that exhibit high visual resemblance), and parts of the video with more drastically altered visual content (which typically show lower visual resemblance). In particular, the computed series of similarity scores undergoes a smoothing procedure with the help of a sliding mean average window of size 3 (see the gray curve in Fig. 1.4). Through this process the algorithm reduces the effect of sudden, short-term changes in the visual content of the video, such as the ones introduced due to camera flash-lights or after a slight hand movement of the camera holder. Following, the turning points of the smoothed series are identified by computing its second derivative (see the yellow vertical lines in Fig. 1.4). Each turning point signifies a change in the similarity tendency and therefore a sub-shot boundary - the latter implies that each video sub-shot is delimited by a pair of subsequent turning points in the smoothed series of similarity scores. Through this process the algorithm indicates both sub-shots having none or small and slowly gradual variation in their visual content, and sub-shots with more drastically changing visual content.

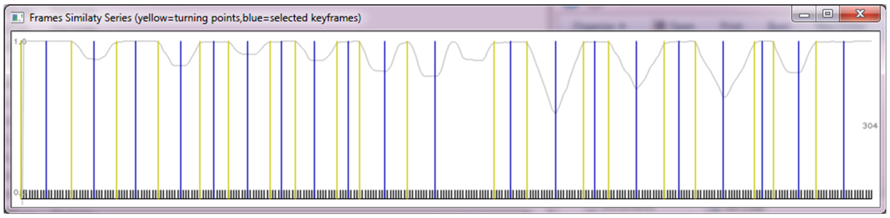


Fig. 1.4: An example of the smoothed series of similarity scores (gray curve), the identified sub-shot boundaries (yellow vertical lines) and the selected representative keyframe (blue vertical lines) for each one of them.

As a final processing (keyframe extraction) step, sub-shots with low variation are represented by their middle frame and sub-shots with more drastically changing visual content are represented by the frame that corresponds to the most pronounced visual change within the sub-shot (see the blue vertical lines in Fig. 1.4). The selected keyframes can be used for supporting fragment-level reverse video search on the Web, as detailed in Section 1.3.2.

1.3.1.2 Video Sub-shot Fragmentation based on Motion Detection

This method (reported in [3]) belongs to the second class of techniques presented in Section 1.2.1. It fragments a single-shot video into sub-shots by identifying self-contained parts which exhibit visual continuity and correspond to individual elementary low-level actions that take place during the video recording. These actions include camera panning and tilting; camera movement in the 3D Euclidean space; camera zoom in/out and minor or no camera movement. The detection of sub-shot boundaries - and the identification of the performed action as an extra feature - is based on the extraction and spatio-temporal analysis of motion information.

Following the reasoning explained in Section 1.3.1.1 regarding the significantly high visual resemblance of successive video frames, the algorithm applies the subsequently described pair-wise motion estimation on neighboring frames selected through a sampling strategy with a fixed-step equal to 10% of the video frame-rate. The conducted motion between a pair of neighboring frames is estimated by computing the region-level optical flow based on the procedure depicted in Fig. 1.5, which consists of the following steps:

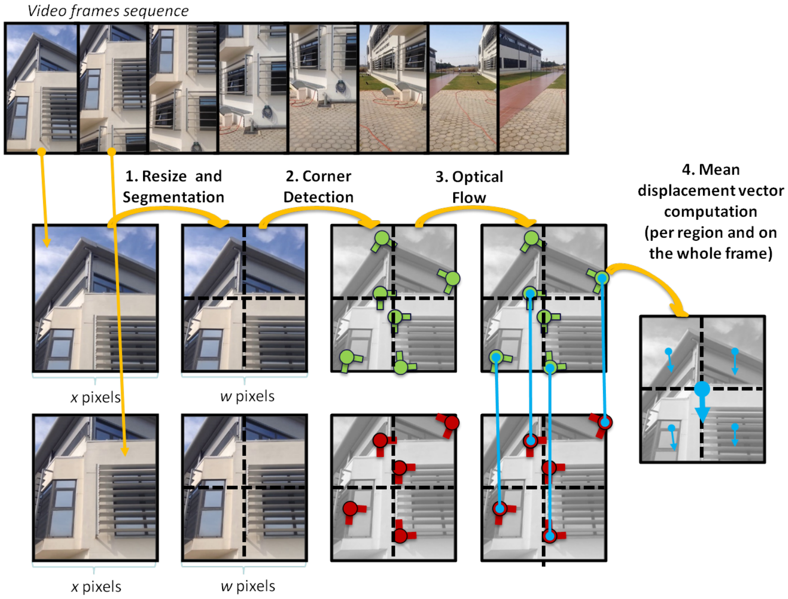

Video frames sequence
1. Resize and Segmentation
2. Corner Detection
3. Optical Flow
4. Mean displacement vector computation (per region and on the whole frame)
x pixels
w pixels
x pixels
w pixels

Fig. 1.5: The steps of the applied procedure for estimating the region-level optical flow between a pair of frames.

- each frame undergoes an image resizing process that maintains the original aspect ratio and makes the frame width equal to w, and then it is spatially fragmented into four quartiles;

- the most prominent corners in each quartile are detected based on the algorithm of [38];
- the detected corners are used for estimating the optical flow at the region-level by utilizing the Pyramidal Lucas Kanade (PLK) method;
- based on the extracted optical flow, a mean displacement vector is computed for each quartile, and the four spatially distributed vectors are treated as a region-level representation of the motion activity between the pair of frames.

To detect (and classify) any displacement of the camera in the 2D space at the frame-level, the algorithm:

- takes the computed region-level mean displacement vectors (left part of Fig. 1.6a, 1.6b, 1.6c);
- averages them producing a frame-level mean displacement vector (middle part of Fig. 1.6a, 1.6b, 1.6c);
- projects the created frame-level mean displacement vector to the horizontal and vertical axis of the Euclidean space (right part of Fig. 1.6a, 1.6b, 1.6c).

Subsequently, a single x-axis vector (Fig. 1.6a) is interpreted as a horizontal-only camera displacement to the vector's direction, a single y-axis vector (Fig. 1.6b) is recognized as a vertical-only camera displacement to the vector's direction, while a pair of x- and y-axis vectors (Fig. 1.6c) is correlated to a diagonal displacement to the frame-level mean displacement vector's direction.

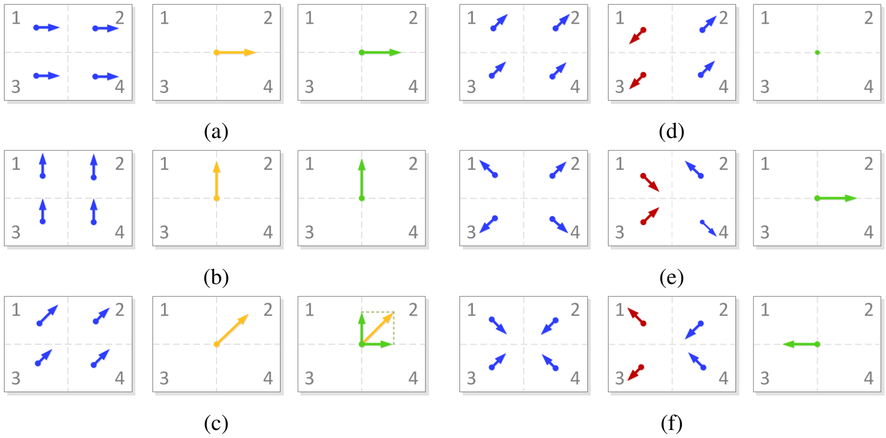
For identifying camera activity at the depth level (i.e. the z-axis of the 3D space) the algorithm:

- takes the computed region-level mean displacement vectors (left part of Fig. 1.6d, 1.6e, 1.6f);
- inverts the direction of the top- and bottom-left vectors (middle part of Fig. 1.6d, 1.6e, 1.6f);
- computes the sum vector and projects it on the x-axis (right part of Fig. 1.6d, 1.6e, 1.6f).

As shown in Fig. 1.6d, the vector inversion process in the case of camera movement at the horizontal and/or vertical axes only, leads to a set of counterbalanced mean displacement vectors and thus, the magnitude of the projection is zero. However, in case of camera activity at the depth axis, the four mean displacement vectors do not maintain the same direction, but point either: to the corners of the frame (Fig. 1.6e), forming a projection vector with positive magnitude, which indicates the existence of forward camera movement or a camera zooming in; or to the centre of the frame (Fig. 1.6f), forming a projection vector with negative magnitude, that denotes the occurrence of backward camera movement or a camera zooming out.

Based on the above the algorithm computes for each pair of frames three values that represent the spatial displacement in x-, y- and z-axis. These values, denoted as V_x, V_y and V_z in the sequel, are normalized in $[-1,+1]$ where:

- V_x $(V_y) = -1$ represents left (downward) displacement of frame pixels equal to 5% of the frame width (height);



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1.6: Motion estimation process for (a) right displacement, (b) upward displacement, (c) diagonal displacement of the camera. Focal distance change estimation process in case of (d) displacement only at horizontal and vertical axes (similar to (c) - thus, no change in the z-axis), (e) forward displacement or camera zoom in, (f) backward displacement or camera zoom out.

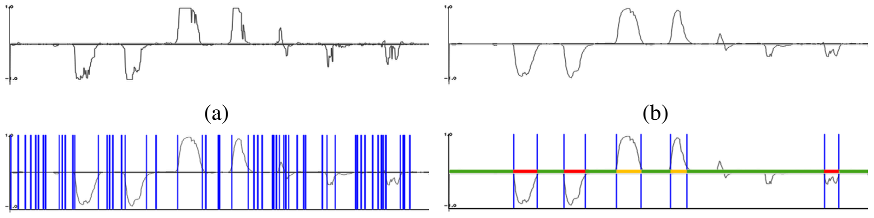
- V_x $(V_y) = +1$ signifies right (upward) displacement of frame pixels equal to 5% of the frame width (height);
- V_x $(V_y) = 0$ denotes no displacement of frame pixels;
- $V_z = -1$ $(+1)$ indicates increment (decrement) of the focal distance that causes inward (outward) spatial displacement of frame pixels equal to 5% of the frame's diagonal;
- $V_z = 0$ indicates no change of the focal distance.

The normalized spatial displacement vectors V_x, V_y and V_z are then post-processed, as described in Algorithm 1, to detect the different sub-shots. Specifically, the values of each vector are initially subjected to low pass filtering in the frequency domain (sample rate equals video frame-rate; cut-off frequency empirically set as 1.0Hz), which excludes sharp peaks related to wrong estimation of the PLK algorithm or quick changes in the light conditions (top row of Fig. 1.7). Each of the filtered vectors V'_x, V'_y and V'_z is then processed for finding its intersection points with the corresponding axis, and the identified intersection points are stored in vectors I_x, I_y and I_z respectively (Fig. 1.7c). These intersection points are candidate sub-shot boundaries, since the video frames between a pair of consecutive intersection points exhibit a contiguous and single-directed camera movement, thus being a potential sub-shot according to the proposed approach.

Driven by the observation that most (single-shot) user-generated videos (UGVs) are captured by amateurs without the use of any professional equipment that ensures camera's stability, the algorithm filters-out fragments depicting minor motion

Algorithm 1 Pseudo code of the proposed technique

```
Input: $V_x$, $V_y$, $V_z$: axes displacement vectors
Output: $O'$: set of sub-shot boundaries
function ProcessVector($V$)
    Low-pass filter $V$. Store in $V'$.
    Detect intersection points in $V'$. Store in $I$.
    Measure the total displacement between intersection points in $I$. Store in $D$.
    Select fragments with displacement $D > t$ as sub-shots. Store in $B$.
end function
$B_x \leftarrow$ ProcessVector($V_x$)
$B_y \leftarrow$ ProcessVector($V_y$)
$B_z \leftarrow$ ProcessVector($V_z$)
Add in $O$ the $B_x$ and $B_y$ fragments.
Extend $O$ by adding $B_z$ fragments that do not coincide with $B_x$ and $B_y$ fragments. Mark remaining parts of the video as fragments with no or minor movement.
Discard fragments less than 1 sec. Store in $O'$.
```



(a) (b)

(c) (d)

Fig. 1.7: Application of Algorithm 1 for a single normalized displacement vector: (a) initial values V_x, (b) low-pass filtered values V'_x, (c) detected candidate sub-shot boundaries in I_x, (d) selected sub-shot boundaries in B_x; red parts denote fragments with left displacement, orange parts denote fragments with right displacement and green parts denote fragments with no or minor movement.

by computing the total displacement over each fragment as the sum of the absolute values of the filtered displacement values V'_x, V'_y and V'_z of each pair of frames in the fragment. This process results in vectors D_x, D_y and D_z, which store the total displacement score of each defined fragment in the x-, y- and z-axis respectively. The video fragments with total displacement score less than an experimentally-defined threshold $t = 12$, are discarded. The determined fragments of each axis are stored in vectors B_x, B_y and B_z (Fig. 1.7d). Following, a simple fusion process is applied, that: i) takes the union O of B_x and B_y fragments, ii) extends it by adding B_z fragments that do not temporally coincide (either completely or partially) with B_x and B_y fragments, and iii) marks the remaining parts of the video as fragments with no or minor movement. The final output of the algorithm (O') is formed by discarding fragments with duration less than 1 sec. through a process that equally dispenses their frames in the previous and the following sub-shot. Each defined video sub-shot is finally represented by its middle frame that is selected as keyframe.

1.3.2 Reverse Video Search on the Web

The InVID project developed a web application for reverse video search on the Web. This web-based tool is directly accessible at http://multimedia3.iti.gr/video_fragmentation/service/start.html, or through the "Keyframes" component of the InVID Verification Plugin[15]. Through its interactive user interface, this technology enables a user to quickly fragment a single-shot video - which is the most common case for UGVs - into visually and temporally coherent parts, using the video sub-shot fragmentation algorithm described in Section 1.3.1.1. The subsequent and automatic selection of a representative keyframe for each defined fragment results in a complete and concise visual summary of the video, that facilitates the time-efficient discovery of the video content and the fragment-level reverse video search on the Web based on the image search functionality of popular Web search engines (e.g. Google search).

Contrary to the technologies presented in Section 1.2.2, that rely on a pre-selected and limited set of video thumbnails (YouTube DataViewer, Custom Reverse Image Search), the manual extraction of video frames for performing reverse image search (TinEye, Karma Decay, Berify), or the creation of collections of (pre-analyzed) video content (RevIMG, Videntifier), this web application extracts a dynamic number of keyframes in a way which ensures that all the visually discrete parts of the video are adequately represented through the extracted set of keyframes. Furthermore, it supports the direct analysis of both online available videos from several platforms and local copies of a video from the user's machine without requiring its prior upload to any video sharing platform. In this way, it assists users to quickly discover the temporal structure of a video, to extract detailed information about the video content and to use this data in their reverse video search queries.

Through the user interface of the InVID web application for reverse video search on the Web, the user is able to submit a video for (video sub-shot fragmentation) analysis, quickly get a rich and representative collection of video keyframes, and perform keyframe-based reverse video search via a "one-click" process. The submitted video can be fetched in two ways: i) either via its URL (in case of an online available video), ii) or by uploading a local copy of it from the user's machine (a typical environment for file browsing and selection is shown to the user through a pop-up window). The provision of the user's e-mail is optional and can be selected in case that the user needs to be automatically notified by e-mail when the analysis results are available.

As stated in the documentation of this tool (accessible after clicking at the "About this tool" button), the supported online video sources include YouTube, Facebook, Twitter, Instagram, Vimeo, DailyMotion, LiveLeak and Dropbox. However the user is being informed that not all videos from these platforms are accessible to the web application, due to platform-specific or user-defined restrictions about the use of each specific video; moreover, the provided URL should always point to a single video, rather than a playlist of videos). Last but not least, the tool can handle videos

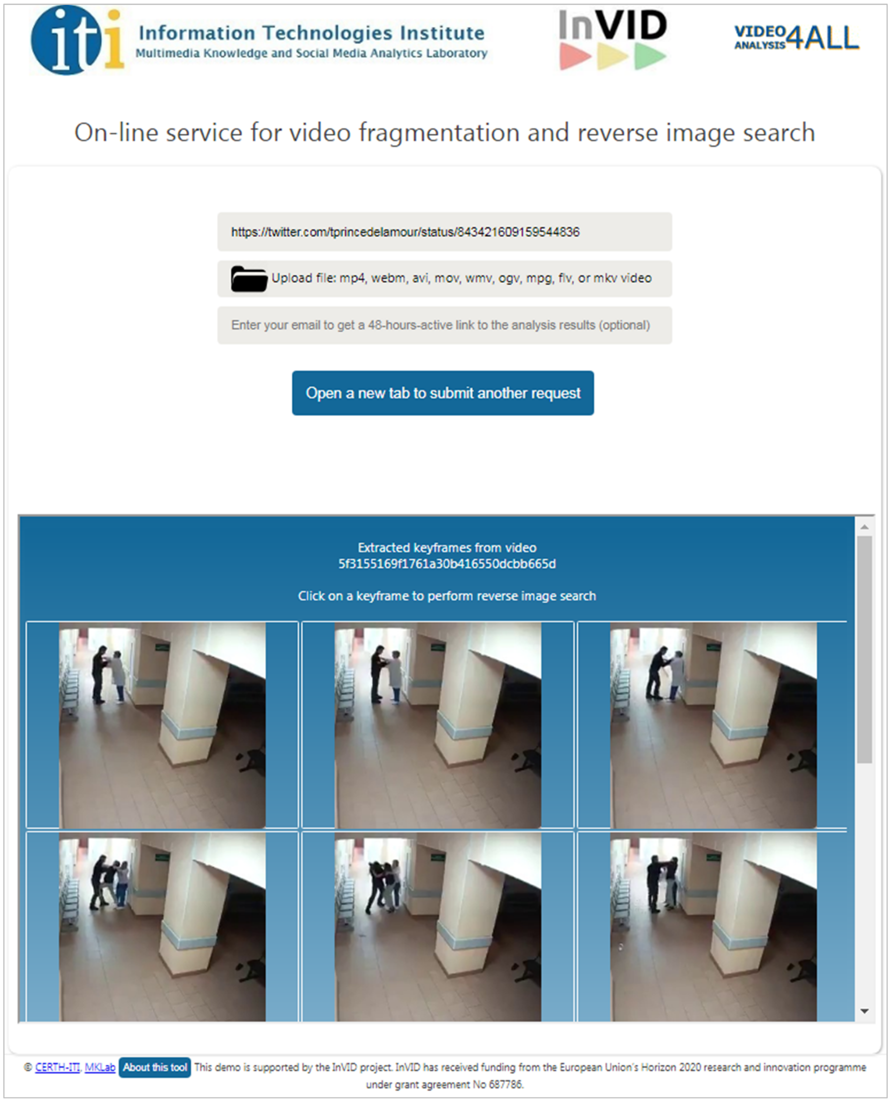
[15] Available at: http://www.invid-project.eu/verify/

in several video formats, including “mp4”, “webm”, “avi”, “mov”, “wmv”, “ogv”, “mpg”, “flv”, and “mkv”.

After submitting a video for analysis the user is able to monitor the progress of the analysis and, after its completion, to get on the screen the collection of extracted keyframes (see Fig. 1.8). Alternatively, if the user provided an e-mail account (which is optional as described before) s/he may close the browser and be notified by e-mail when the analysis results are ready. The provided collection of keyframes allows the user to explore the video structure (in the sub-shot-level) and perform reverse keyframe search, simply by (left) clicking on any desired keyframe. This action initiates a Google image search and the results of this search are served to the user in a new tab of his/her browser (see Fig. 1.9). In case more keyframes are needed for performing a more extended search for the video, the user can click on the “Show more keyframes” button that appears right after the initial collection of extracted keyframes (right part of Fig. 1.10); these keyframes correspond to the same video fragments with the initially provided ones, so they could contain duplicates (left part of Fig. 1.10). The generated results from the analysis (i.e. the collection of keyframes) are available only for 48 hours, and are automatically deleted from the server after this time period. All video rights remain with the uploader, who is assumed to have the right to submit the video to this service for analysis.

The feedback concerning the performance of this tool, received mainly from the users of the corresponding component of the InVID Verification Plugin, is very positive and encouraging. According to the analytics about the use of this web application since the public release of the plugin, more than $4,000$ users have submitted (in total) over $13,500$ requests, for analysing more than 650 hours of video content. Moreover, a group of approximately 150 “power-users” - coming mostly from news agencies, human rights organizations and media verification networks - have used the tool more than 20 times each, while the top-10 of them have used the tool more than 100 times each. The collected traffic data indicate that there is significant (and constantly raising) community of regular users that exploit the verification functionality of the tool on a frequent basis. The functionality of this component enabled the users to debunk a number of fake news that are based on the reuse of a previously published video. Indicative examples of such fakes and the corresponding original video sources that were identified with the help of the web application, can be found in Table 1.1.

Last but not least, this web application has a complementary role with the near duplicate detection utility of the InVID Verification Application, which is presented in the next Chapter 4. The former allows the fragment-level reverse search of videos on the Web using the extracted keyframes, while the latter enables the video-level reverse search of videos within a constantly extendible collection of selected newsworthy video material.



Information Technologies Institute
Multimedia Knowledge and Social Media Analytics Laboratory
InVID
VIDEO ANALYSIS 4ALL
Open a new tab to submit another request
About this tool

On-line service for video fragmentation and reverse image search

https://twitter.com/tprincedelamour/status/843421609159544836

Upload file: mp4, webm, avi, mov, wmv, ogv, mpg, flv, or mkv video

Enter your email to get a 48-hours-active link to the analysis results (optional)

Extracted keyframes from video
5f3155169f1761a30b416550dcbb665d

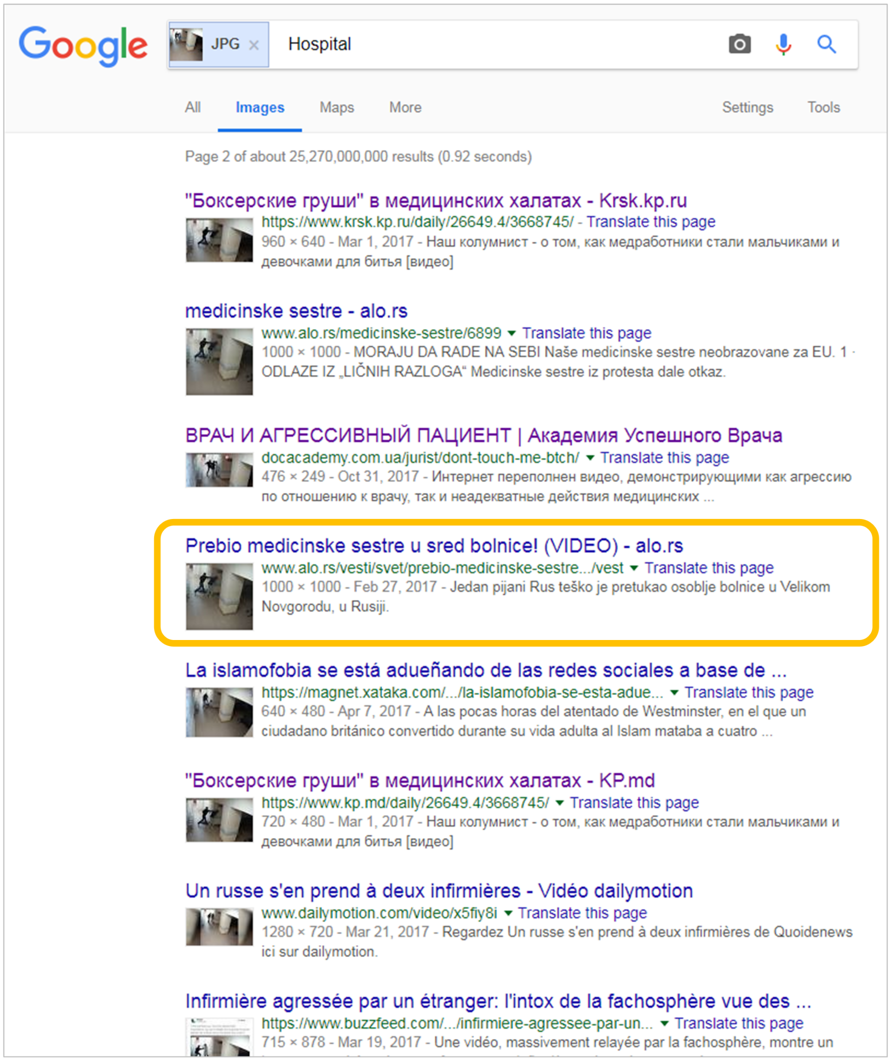
Click on a keyframe to perform reverse image search

© CERTH-ITI, MKLab This demo is supported by the InVID project. InVID has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687786.

Fig. 1.8: Provision of extracted keyframes after the completion of the analysis.

1.4 Performance Evaluation and Benchmarking

This part reports on the conducted experiments for evaluating the performance of the developed algorithms for video sub-shot segmentation, and for assessing the usefulness and effectiveness of the InVID web application for reverse video search on the Web.



Hospital

Page 2 of about 25,270,000,000 results (0.92 seconds)

"Боксерские груши" в медицинских халатах - Krsk.kp.ru
https://www.krsk.kp.ru/daily/26649.4/3668745/ - Translate this page
960 × 640 - Mar 1, 2017 - Наш колумнист - о том, как медработники стали мальчиками и девочками для битья [видео]

medicinske sestre - alo.rs
www.alo.rs/medicinske-sestre/6899 ▾ Translate this page
1000 × 1000 - MORAJU DA RADE NA SEBI Naše medicinske sestre neobrazovane za EU. 1 · ODLAZE IZ „LIČNIH RAZLOGA" Medicinske sestre iz protesta dale otkaz.

ВРАЧ И АГРЕССИВНЫЙ ПАЦИЕНТ | Академия Успешного Врача
docacademy.com.ua/jurist/dont-touch-me-btch/ ▾ Translate this page
476 × 249 - Oct 31, 2017 - Интернет переполнен видео, демонстрирующими как агрессию по отношению к врачу, так и неадекватные действия медицинских ...

Prebio medicinske sestre u sred bolnice! (VIDEO) - alo.rs
www.alo.rs/vesti/svet/prebio-medicinske-sestre.../vest ▾ Translate this page
1000 × 1000 - Feb 27, 2017 - Jedan pijani Rus teško je pretukao osoblje bolnice u Velikom Novgorodu, u Rusiji.

La islamofobia se está adueñando de las redes sociales a base de ...
https://magnet.xataka.com/.../la-islamofobia-se-esta-adue... ▾ Translate this page
640 × 480 - Apr 7, 2017 - A las pocas horas del atentado de Westminster, en el que un ciudadano británico convertido durante su vida adulta al Islam mataba a cuatro ...

"Боксерские груши" в медицинских халатах - KP.md
https://www.kp.md/daily/26649.4/3668745/ ▾ Translate this page
720 × 480 - Mar 1, 2017 - Наш колумнист - о том, как медработники стали мальчиками и девочками для битья [видео]

Un russe s'en prend à deux infirmières - Vidéo dailymotion
www.dailymotion.com/video/x5fiy8i ▾ Translate this page
1280 × 720 - Mar 21, 2017 - Regardez Un russe s'en prend à deux infirmières de Quoidenews ici sur dailymotion.

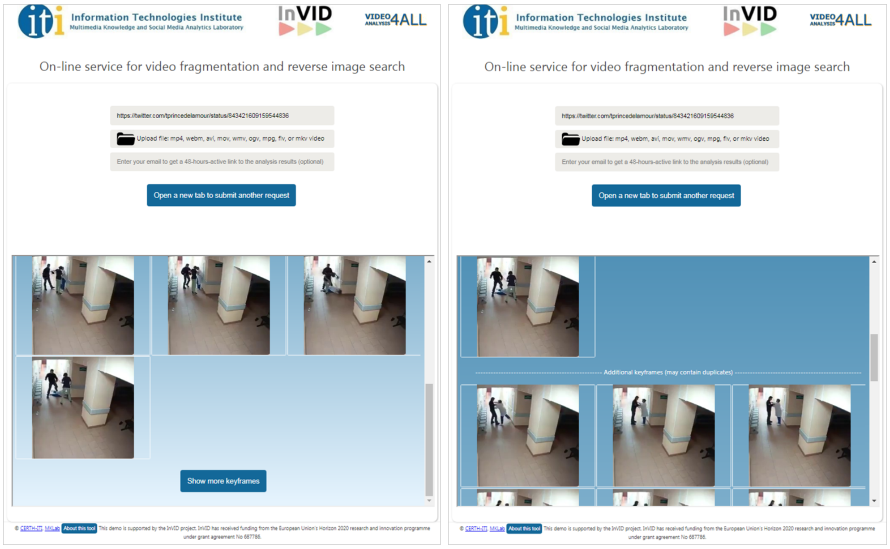
Infirmière agressée par un étranger: l'intox de la fachosphère vue des ...
https://www.buzzfeed.com/.../infirmiere-agressee-par-un... ▾ Translate this page
715 × 878 - Mar 19, 2017 - Une vidéo, massivement relayée par la fachosphère, montre un

Fig. 1.9: The results after applying reverse image search on one of the extracted keyframes. Within the yellow bounding box is the result that leads to a near-duplicate of the video, that corresponds to the originally published one.

1.4.1 Video Fragmentation

Driven by the lack of publicly available datasets for evaluating the performance of video sub-shot fragmentation algorithms[16], we built our own ground-truth dataset. This dataset is publicly available[17] and consists of:

[16] Some works reported in Section 1.2.1 use certain datasets (TRECVid 2007 rushes summarization, UT Ego, ADL and GTEA Gaze) which were designed for assessing the efficiency of methods



On-line service for video fragmentation and reverse image search

https://twitter.com/tprincedelamour/status/843421609159544836

Upload file: mp4, webm, avi, mov, wmv, ogv, mpg, flv, or mkv video

Enter your email to get a 48-hours-active link to the analysis results (optional)

© CERTH-ITI, MKLab About this tool This demo is supported by the InVID project. InVID has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687786.

Information Technologies Institute
Multimedia Knowledge and Social Media Analytics Laboratory
InVID
VIDEO ANALYSIS 4ALL
Open a new tab to submit another request
Show more keyframes
Additional keyframes (may contain duplicates)

Fig. 1.10: Additional keyframes can be optionally provided to the user for more extended search.

- 15 single-shot videos of total duration 6 minutes, recorded in our facilities; these videos, denoted as “own videos” in the sequel, contain clearly defined fragments that correspond to several video recording activities.
- 5 single-shot amateur videos of total duration 17 minutes, found on YouTube; these videos are denoted as “amateur videos” in the sequel.
- 13 single-shot parts of known movies of total duration 46 minutes; these videos, denoted as “movie excerpts”, represent professional video content.

Ground-truth fragmentation of the employed dataset was created by human annotation of the sub-shot boundaries for each video. Adopting the most commonly used approach from the relevant literature for segmenting a single-shot video into sub-shots, each determined sub-shot boundary indicates the end of a visually and temporally contiguous activity of the video recording device and the start of the next one (e.g. the end of a left camera panning, which is followed by a camera zooming). This approach might not be strictly aligned to the fragmentation criterion of methods relying on visual resemblance among frames, but we can claim that all sub-shots defined by the aforementioned strategy exhibit high levels of visual similarity, and thus could be identified by similarity-based fragmentation algorithms as well. Overall, our dataset contains 674 sub-shot transitions.

targeting specific types of analysis, such as video rushes fragmentation [4] and the identification of everyday activities [41] and thus, ground-truth sub-shot fragmentation is not available for them.

[17] http://mklab.iti.gr/project/annotated-dataset-sub-shot-segmentation-evaluation

Table 1.1: Indicative list of fakes debunked using the web application for video fragmentation and reverse keyframe search.

Fake news	Claim	Date	Original source	Fact	Date
https://www.facebook.com/ Pakkorner/videos/ 365709494264601/	Pakistani soldiers making a floating bridge over a river	Apr. 2019	https://www.youtube.com/ watch?v=mju6XUIlm6I	Troops build pontoon bridge during NATO drills in Lithuania	Jun. 2017
https://www.facebook.com/ Army.Of.Pakistan/videos/ 417288652413285/	Firing by Pakistan's military at the border with India	Mar. 2019	https://www.youtube.com/ watch?v=WHlMoz2E-tw	Pakistan Army Random Infantry Fire Power Show	Jul. 2017
https://www.facebook.com/ LogKyaKahengyy/videos/ 404467560327831/	Effigy of Pakistani Prime Minister Imran Khan being burnt	Feb. 2019	https://www.youtube.com/ watch?v=DruBl3Py3zY	Congress workers injured while burning Modi effigy in Shimla	Dec. 2015
https://twitter.com/i/status/ 1096811492098289664	Pulwama terror attack footage	Feb. 2019	https://www.youtube.com/ watch?v=8l-IUqsHR9Q	Truck bomb in Iraq	Apr. 2008
https://www.facebook.com/ halimhusin.my/videos /2119944594692914/	China opened 880 km highway linking their country to Pakistan	Feb. 2019	https://www.youtube.com/ watch?v=YbzT8ycTjQc	Yaxi Highway, a 240 km-long highway in China's Sichuan province	Jan. 2019
https://www.facebook.com/ TimeNewsInternational/videos /2187809244837800/	Plane caught in a typhoon in China	Sep. 2018	https://www.youtube.com/ watch?v=AgvzhJpyn10	Video of a company specialised in digital special effects	Jun. 2017
https://www.facebook.com/ 100009631064968/videos/ 730611413936554/	Muslims attack cars in Birmingham UK	May 2018	https://www.youtube.com/ watch?v=rAoQTQE_YTY	Hooligans from Zurich faced off with hooligans from Basel	May 2018
https://www.youtube.com/ watch?v=C4BjUoQAw5Y	Migrants attacking cars in Metz, France	May 2018			
https://twitter.com/kwilli1046 /status/872106123570163712 https://www.youtube.com/ watch?v=OVAxQA3gMEo	Attack in Notre Dame, Paris	Jun. 2017	https://www.youtube.com/ watch?v=W2IA9UwmHCA	World War Z making off	Sep. 2012
https://twitter.com/mikethecraigy /status/877248566384873472	Attack in Brussels Central Station	Jun. 2017			
https://www.youtube.com/ watch?v=HDZWj6MjYbk	Casino robbery in Manila Philippines	Jun. 2017	https://www.youtube.com/ watch?v=MX3YCSpl2tM	Robbery in a hotel in Suriname	Jan. 2012
https://twitter.com/ tprincedelamour/status/ 843421609159544836	Immigrant attacks nurse in public hospital in France	Mar. 2017	https://www.youtube.com/ watch?v=CuyfdZKc3TQ	Drunk patient beats up doctors in Novgorod hospital in Russia	Feb. 2017
https://twitter.com/FuegoNugz /status/905246797123203072	Hurricane Irma in Barbados, US	Sep. 2017	https://www.youtube.com/ watch? v=0lHDVel-NPw	Hurricane Dolores in Uruguay	May 2016
https://www.youtube.com/ watch?v=fmUEI0L2alY	Hurricane Otto in Panama	Nov. 2016			
https://www.facebook.com/ anisrahimrahim/videos/ 1113757488646580/	Inside the EgyptAir airplane before the crash	May 2016	https://www.stuff.co.nz/travel/ travel-troubles/79637635/severe -turbulence-injures-32-on- etihad-flight-to-indonesia	Severe turbulence injures 32 on Etihad flight to Indonesia	May 2016
https://www.youtube.com/ watch?v=mZes8-tzZ0w	Explosion in Brussels airport	Mar. 2016	https://www.youtube.com/ watch?v=yhO7gZobaqY	Attack in Domodedovo airport in Russia	Jan. 2011
https://www.youtube.com/ watch?v=nkQ-ij3LTTM	Video showing a Hezbollah sniper	Feb. 2016	https://www.youtube.com/ watch?v=Xjq5VlPdAe4	"Let's play" video from "Medal of Honor"	Aug. 2012
https://www.youtube.com/ watch?v=Q-yWYQLwm5M	Brave revel against tank in Syria	Jul. 2015	http://www.military.com/ video/operations-and- strategy/antitank-weapons/ rocket-hits-syrian-tank-at- close-range/1826311054001	Rocket hits Syrian tank	Sep. 2012

The performance of the algorithms presented in Section 1.3.1.1 (denoted as S_DCT) and Section 1.3.1.2 (denoted as SP_OF) was compared against other methods from the relevant literature, that include:

- A straightforward approach (denoted S_HSV in the sequel) which assesses the similarity between subsequent video frames with the help of HSV histograms and x^2 distance.
- A method (denoted B_HSV) similar to [36], that selects the first frame of the video F_a as the base frame and compares it sequentially with the following

ones using HSV histograms and x^2 distance until some frame F_b is different enough, then frames between F_a and F_b form a sub-shot, and F_b is used as the next base frame in a process that is repeated until all frames of the video have been processed; a variation of this approach (denoted B_DCT) that represents the visual content of the video frames using DCT features and estimates their visual resemblance based on the cosine similarity was also implemented.
- The algorithm of [9] (denoted A_SIFT), which estimates the dominant motion between a pair of frames based on the computed parameters of a 3×3 affine model through the extraction and matching of SIFT descriptors; furthermore, variations of this approach that rely on the use of SURF (denoted A_SURF) and ORB [37] (denoted A_ORB) descriptors were also implemented for assessing the efficiency of faster alternatives to SIFT.
- An implementation of the best performing technique of [10] (denoted A_OF), which computes the optical flow using the PLK algorithm and identifies camera movement by fitting it to a 2×2 affine model containing parameters that represent the camera pan, tilt, zoom and rotation actions.
- Variations of the local-feature-based approaches documented in [10], that rely on the extraction and matching of SIFT, SURF and ORB descriptors (denoted H_SIFT, H_SURF and H_ORB, respectively) or the computation of the optical flow using PLK (denoted H_OF), for estimating the dominant motion based on specific parameters of the homography matrix computed by the RANSAC method [14]; an example of SURF-based homography estimation between a pair of frames is depicted in Fig. 1.11.

For each one of the tested approaches the number of correct detections (where the detected boundary can lie within a temporal window around the respective ground-truth boundary, equal to twice the video frame-rate), misdetections and false alarms were counted and the algorithms' performance was expressed in terms of Precision (P), Recall (R) and F-Score (F), similarly to [1, 2]. Time efficiency was evaluated by computing the ratio of processing time over the video's duration (a value below 1 indicates faster-than-real-time processing). All experiments were conducted on a PC with an i7-4770K CPU and 16GB of RAM.

Table 1.2 reports the evaluation results of each compared approach, both separately on each of the three parts of the dataset, as described above, and on the overall dataset. General observations regarding the different implemented methodologies are the following.

- Approaches that estimate the dominant motion based on a homography matrix seem to be more effective compared to the methods that rely on affine models or the assessment of visual similarity, with the latter ones being slightly better compared to the affine-based methods.
- Among the examined similarity-based techniques, the use of HSV histograms results in better performance in terms of precision; however, the utilization of DCT features leads to remarkably higher recall scores, and thus a better overall performance (F-score).

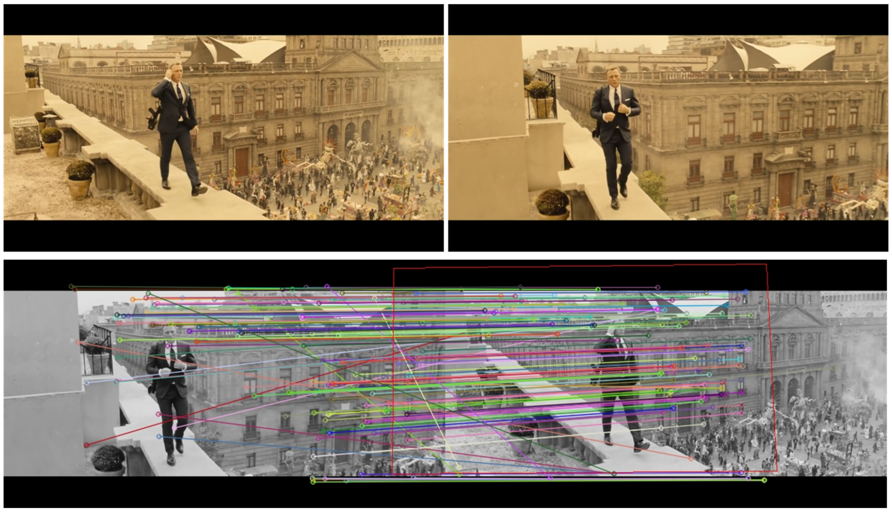


Fig. 1.11: Local feature extraction and matching for computing the homography between a pair of frames (i.e. how the image on the left should be translated to match the one on the right), that subsequently allows motion detection and estimation. Note: frames were extracted from the opening scene of the “Spectre” movie.

Table 1.2: Evaluation results for different sub-shot fragmentation approaches (P: Precision, R: Recall, F: F-score).

	“Own videos”			“Amateur videos”			“Movie excerpts”			Overall dataset		
Method	P	R	F	P	R	F	P	R	F	P	R	F
S_HSV	0.31	0.28	0.30	0.23	0.09	0.13	0.28	0.44	0.34	0.28	0.36	0.32
S_DCT	0.54	0.88	0.67	0.14	**0.86**	0.25	0.25	**0.84**	0.38	0.22	**0.84**	0.36
B_HSV	0.30	0.09	0.14	**0.55**	0.09	0.16	0.43	0.12	0.18	0.44	0.11	0.18
B_DCT	0.50	0.23	0.32	0.36	0.40	0.38	0.43	0.24	0.31	0.41	0.27	0.32
A_OF	0.41	0.68	0.50	0.20	0.82	0.31	0.30	0.78	0.43	0.27	0.78	0.40
A_SIFT	0.55	0.62	0.59	0.20	0.09	0.12	0.30	0.14	0.19	0.33	0.17	0.23
A_SURF	0.54	0.64	0.58	0.29	0.30	0.29	0.36	0.25	0.30	0.36	0.29	0.33
A_ORB	0.40	0.25	0.30	0.09	0.02	0.03	0.46	0.02	0.05	0.38	0.05	0.08
H_OF	**0.98**	0.62	0.76	0.26	0.67	0.38	0.41	0.58	0.47	0.37	0.60	0.45
H_SIFT	0.90	0.74	0.82	0.27	0.78	0.39	0.35	0.63	0.45	0.34	0.66	0.45
H_SURF	0.88	0.73	0.80	0.26	0.70	0.38	0.36	0.64	0.47	0.36	0.66	0.46
H_ORB	0.85	0.67	0.75	0.18	0.76	0.30	0.30	0.73	0.43	0.28	0.72	0.40
SP_OF	0.96	**0.90**	**0.93**	0.42	0.71	**0.53**	**0.48**	0.64	**0.55**	**0.52**	0.70	**0.59**

- Concerning the implemented affine-based techniques, the most efficient is the one that relies on the optical flow, showing the highest recall scores in all different video categories and comparable precision scores with the other related methods.
- Regarding the suitability of local descriptors for computing an affine model that helps with the identification of the performed movement, SURF are the

most effective ones, SIFT perform slightly worse, and ORB exhibit the weakest performance.

- With respect to the evaluated homography-based approaches, the use of different local descriptors or optical flow resulted in similar efficiency, with ORB being the least competitive descriptor due to lower precision.
- In terms of precision, the achieved scores are rather low in general, a fact that indicates the limited robustness of all tested approaches against the challenging task of segmenting an uninterruptedly captured video in sub-shots; methods that evaluate the similarity between distant frames (B_HSV and B_DCT) achieve precision slightly over 0.4, and are being more efficient than the sequentially operating ones (S_HSV and S_DCT); motion-based methods that rely on affine transformations or homography matrices exhibit slightly worse performance, reaching a precision around 0.35; finally, the highest precision (slightly over 0.5) is scored when the video recording activities are modelled by computing and evaluating the optical flow in both the spatial and temporal dimension of the video.

Regarding the best performing approaches, the last row of Table 1.2 indicates that the most effective method is the spatio-temporal motion-based algorithm of Section 1.3.1.2. This algorithm achieves the highest F-score both on the overall dataset, as well as on each different part of it. On the first collection of videos it exhibits the highest recall score, with the method of Section 1.3.1.1 being the second best, while its precision is slightly lower than the one achieved by the H_OF method. On "Amateur videos" the SP_OF technique is again the best performing one, while the B_HSV method and the technique of Section 1.3.1.1, that presented competitive precision and recall respectively, achieved significantly lower overall performance. Similar efficiency is observed when analysing single-shot parts of professional movies; the SP_OF approach is the best in terms of F-score and precision. The above are reflected in the last three columns of Table 1.2 which show the superiority of this algorithm over the other evaluated techniques in the overall dataset. Two indicative examples of how this algorithm fragments (a part of) two UGVs are presented in Fig. 1.12. Another finding that can be easily extracted from the results reported in Table 1.2 relates to the ability of the DCT-based algorithm of Section 1.3.1.1 to achieve high recall scores. This method proved to be the most effective one in terms of recall when analysing "Amateur videos" and "Movie excerpts", while it was the second best performing approach on "Own videos" with a score slightly lower than the one achieved by the motion-based technique of Section 1.3.1.2. The competency of the DCT-based technique to achieve high recall scores is recorded also for the entire dataset, as shown in the penultimate column of Table 1.2.

With respect to the time-efficiency, as shown in Table 1.3, the more straightforward approaches that fragment a video based on the visual resemblance of video frames are faster than methods computing the parameters of affine models or homography matrices, as expected. Moreover, the use of DCT features, especially in the way that the method of Section 1.3.1.1 utilises them, outperforms the HSV histograms, while the extraction and matching of complex local descriptors (SIFT and SURF) is more computationally expensive compared to the matching of binary de-

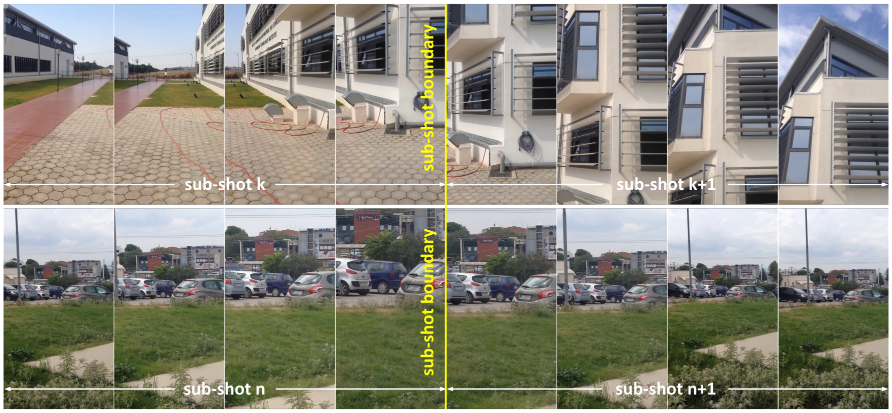


Fig. 1.12: Top row: a sequence of video frames (sampled for space and presentation efficiency) fragmented by the proposed algorithm into two sub-shots; one related to a horizontal and one related to an upward camera movement. Bottom row: a sequence of video frames (sampled for space and presentation efficiency) fragmented by the proposed algorithm into two sub-shots; one related to a camera zooming in and one related to camera zooming out.

scriptors (ORB) or the extraction of optical flow for computing the affine or homography matrices. The SP_OF approach of Section 1.3.1.2 exhibits competitive time performance, being a bit slower than the straightforward similarity-based methods and faster than almost the entire set of the evaluated affine- and homography-based techniques. Its time efficiency permits sub-shot fragmentation to be performed 9 times faster than real-time analysis, while this performance can be further improved by introducing simple parallelization in the algorithm's execution. In fact, a multi-threaded software implementation of this technique splits the group of analysed frames into 4 different and non-overlapping parts which are being processed (i.e. for extracting the optical flow among each pair of frames) in parallel on the CPU. The lightweight post processing of the computed displacement vectors for motion detection and recognition is still carried out using a single thread. Experiments on the same dataset showed a 267% speed-up compared to the single-thread version, which means that the analysis of a single-shot video with the multi-thread implementation of the algorithm takes only 4.1% of the video's duration. This performance is comparable with the time-efficiency of the fastest approach, namely the DCT-based algorithm of Section 1.3.1.1 which completes the analysis in less than 3% of the video length and thus, enables video processing more than 30 times faster compared to real-time processing.

The findings concerning the detection accuracy and the time-efficiency of the comparatively evaluated approaches document that:

- the motion-based algorithm of Section 1.3.1.2 combines the time-efficiency of similarity-based approaches that rely on the extraction of lightweight visual de-

Table 1.3: Time-efficiency of the evaluated sub-shot fragmentation approaches.

Method	S_HSV	S_DCT	B_HSV	B_DCT	A_OF	A_SIFT	A_SURF	A_ORB	H_OF	H_SIFT	H_SURF	H_ORB	Proposed
Proc. time % of video length	7.1	**2.9**	3.8	6.7	7.8	127.2	56.3	12.7	14.5	132.6	70.2	16.1	11.1

scriptors with the detection effectiveness of more complex state-of-the-art techniques that estimate the dominant motion with the help of affine transformations and image homography;

- the similarity-based method of Section 1.3.1.1 can be a reasonable choice when high recall is needed, i.e. when the over-fragmentation of the video and the creation of an over-wealthy set of representative keyframes favors the next steps of the analysis, as in the case of fragment-level reverse video search;
- there is room for further improvement (mainly in terms of Precision) of the current methods for video sub-shot fragmentation, and for this purpose the performance of modern deep network architectures that capture the visual and temporal dependency among video frames (such as Recursive Neural Networks with Long Short-Term Memory (LSTM) units) could be exploited.

1.4.2 Reverse Video Search on the Web

The InVID approach for indexVideo sub-shot fragmentation video sub-shot fragmentation and keyframe extraction (presented in Section 1.3.1.1) was comparatively evaluated against two alternative baseline approaches for keyframe extraction; one extracting 1 keyframe per second, and another one that extracts the reference frames (a.k.a. I-frames) of the mp4 video stream[18]. This benchmarking was conducted with the help of two journalists - one coming from Agence France-Presse (AFP) and one coming from Deutsche Welle (DW) - with media verification background, and its focus was bilateral. In particular, it aimed to assess:

- the efficiency of each tested approach in defining a set of keyframes that represents the visual content of the video without missing any important pieces of information, with the least amount of frames;
- the usefulness / appropriateness of each generated keyframe collection for supporting the task of finding near duplicates of the analysed video on the Web.

Given that the evaluated InVID method is integrated into the web application for reverse video search, this testing allowed to assess how concise and complete the produced collection of keyframes is, and to which extend the generated collection

[18] Both of these approaches were implemented using the FFmpeg framework that is available at: https://www.ffmpeg.org/

(and thus this web application) facilitates the quick identification of prior occurrences of a given video on the Web.

According to the evaluation protocol each tester was asked to select 10 user-generated videos; these videos could be either online available videos from the Web or local videos from the testers' machines. Experimentation with non-user-generated videos (i.e. edited professional videos) was also permitted. Subsequently, each selected video should be submitted for analysis to:

- the InVID web application for reverse video search that uses the InVID approach for video fragmentation and keyframe selection;
- a variation of this tool that creates a keyframe collection by applying the first alternative and extracts one keyframe per second;
- another variation of this tool that defines a keyframe collection by applying the second alternative and extracts the reference frames (a.k.a. I-frames) of the mp4 video stream.

After analysing each selected video with the above listed technologies, the testers had to answer the following questions.

- Q1: How many keyframes were extracted by each tested approach?
- Which collection is the most concise and complete one (i.e. represents the visual content of the video without missing any important pieces of information, with the least amount of frames)?
- Q3: If you try reverse image search: which collection helps you the most to quickly identify near duplicates of the video on the Web?
- Q4: if the used videos are publicly accessible, please copy-paste the links at the end of this document.

Their feedback was provided by filling-in the following tables Tables 1.4 and 1.5, while the submitted videos by each tester are listed in Table 1.6. In the utilised ranking system for answering questions Q2 and Q3, "1" stands for the worse performance and "5" stands for the best performance.

Table 1.4 contains the evaluation results of the AFP journalist. The collected feedback showed that the InVID approach exhibits competitive performance compared to the other tested approaches. Concerning the generation of a concise and complete keyframe-based summary of the video content, the InVID algorithm was the highest-voted one in 7 cases, and the second best performing one in the remaining 3 cases. The representation efficiency of the first alternative, which extracts 1 keyframe per second, was positively appreciated by the AFP journalist in 4 cases where the algorithm was voted as the best (or among the best) performing one(s). The second alternative that selects the I-frames of the video was indicated as the least efficient one and marked as the second best in 4 cases only.

The good ranking of the first alternative approach reveals the AFP journalist's preference in having keyframe collections that sufficiently cover all the details of the video, even if this entails a compromise regarding the comprehensiveness of the created collection and the existence of information redundancy. As further detailed in his evaluation report, the explanation behind this choice is governed by his

Table 1.4: The votes of the AFP journalist regarding the tested approaches for video keyframe extraction and keyframe-based reverse video search.

		Q1: extracted keyframes	Q2: concise and complete					Q3: helps the most in reverse search				
	Method		1	2	3	4	5	1	2	3	4	5
Video #1	InVID	17				X					X	
	Alt. #1	43		X						X		
	Alt. #2	12			X				X			
Video #2	InVID	6		X					X			
	Alt. #1	17				X				X		
	Alt. #2	4	X					X				
Video #3	InVID	101				X					X	
	Alt. #1	371		X					X			
	Alt. #2	127			X					X		
Video #4	InVID	4				X					X	
	Alt. #1	19					X			X		
	Alt. #2	5				X					X	
Video #5	InVID	9				X				X		
	Alt. #1	29				X				X		
	Alt. #2	46			X				X			
Video #6	InVID	10				X				X		
	Alt. #1	43			X					X		
	Alt. #2	43			X					X		
Video #7	InVID	65				X				X		
	Alt. #1	210			X					X		
	Alt. #2	92			X					X		
Video #8	InVID	13				X				X		
	Alt. #1	46			X					X		
	Alt. #2	45			X					X		
Video #9	InVID	85		X						X		
	Alt. #1	303				X					X	
	Alt. #2	72		X					X			
Video #10	InVID	31			X						X	
	Alt. #1	74			X					X		
	Alt. #2	32			X					X		

news verification background and relies in the fact that some video frames might contain an element that helps to confirm the location, identify a person, a scene, an event or something useful for the verification or debunking of a news video. As a consequence, the appearance of these frames in the keyframe collection, even if near-duplicates of them - that are less informative though - are already included in this collection, is positively assessed. Finally, the keyframe collections generated by the second alternative, even being comparatively-sized with the ones created by the InVID algorithm (see Table 1.4), proved to be less useful that the other evaluated techniques due to more missing frames that are needed for effectively conveying the reported story in the video.

An example that illustrates the findings reported above is depicted in Fig. 1.13 which contains the generated keyframe collections by the three evaluated algorithms for the analysed video #4. The top left corresponds to the InVID method, the bottom left corresponds to the second alternative and the right-sided one corresponds to the first alternative. The video reports a story about the first woman in Saudi Arabia that receives her driving license, and it is recorded within an office by a (mainly) standing cameraman. The InVID-extracted keyframe collection contains 3 keyframes that show the provision of the license by the officer to the woman. The keyframe collection created by the second alternative conveys (visually) the same information but exhibits more redundancy, as keyframes #3 and #4 are near-duplicates of keyframes #2 and #5 respectively. Finally, the collection generated by the first alternative covers the story in much more details, but the cost of with much higher duplication of the visual information. Nevertheless, the last keyframe of this collection shows a photographer that is also in the room and takes a photo of this event. His appearance in the video does not change or affect the main subject of the video, but it can provide a hint that could help a journalist to verify or debunk this video (e.g. by observing a badge on his uniform that relates to a specific country or army). Hence, the journalist's voting (as shown in Table 1.4) rewards the existence of this keyframe in the collection, considering it as more important than the information redundancy that this collection presents.

Concluding, the keyframe selection strategy of the first alternative combined with the competitive performance of the InVID approach in most examined cases, indicates the InVID method as the most efficient one in generating keyframe-based video summaries that are well-balanced according to the determined criteria for the descriptiveness (completion) and representativeness (conciseness) of the keyframe collection.

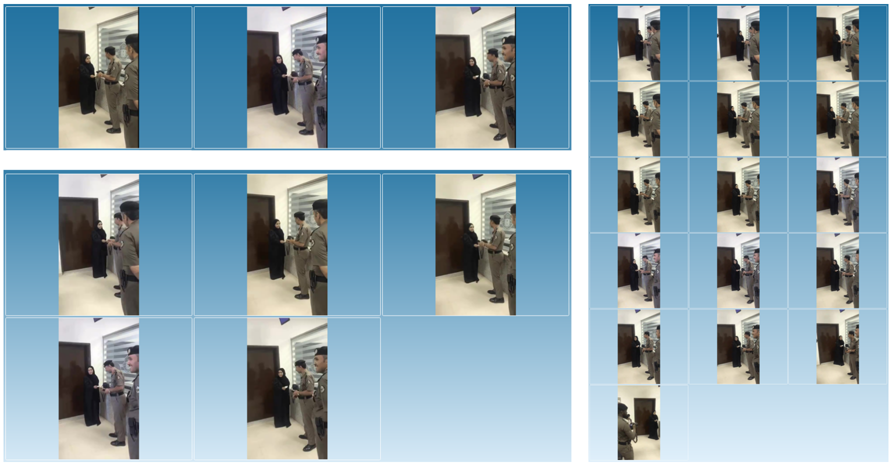


Fig. 1.13: The keyframe collections generated for an AFP-selected video by the three tested approaches. The top left corresponds to the InVID method, the bottom left corresponds to the second alternative and the right-sided corresponds to the first alternative.

Concerning the use of the generated keyframe collections by the 3 evaluated methods to facilitate the identification of near-duplicates of the processed videos on the Web, the InVID method was generally determined as the most useful one. The keyframe collections extracted by this method, were considered as helping the most in reverse video search in 3 cases, and as equally effective one with the collections produced by other approaches in 5 cases. The first alternative proved to be the second most appreciated method, and this finding is aligned with the journalist's interest, explained previously, to get and use any visual detail of the video that could assist its verification. Finally, the second alternative was ranked as the less efficient one since the extracted keyframe collections were denoted as less useful for video reverse search in several of the tested scenarios.

These outcomes are consistent to the findings regarding the comprehensiveness and fullness of the generated keyframe collections, and show that the InVID developed algorithm and the first alternative can (almost) equally support the users' needs when performing a fragment-level reverse search of a video on the Web.

Table 1.5 includes the evaluation results of the DW journalist. The received feedback clearly indicates the InVID approach as the best performing one in producing a concise and complete keyframe-based summary of the video. In most cases (specifically in 9 out of 10) the InVID method got the highest score compared to the other tested approaches. The keyframe collections generated by this algorithm were voted as best (4 times) or well (4 times) performing ones. A similar, but in some cases less efficient, performance was shown by the second alternative which extracts the I-frames of the video. This technique was evaluated as approximately equally performing one with the InVID approach in 6 cases, while in one case it was voted as the most effective technique. This finding is reasonable if we take under consideration that this method: a) selects the frames of the video that are the most complete and descriptive ones in terms of visual information (in order to be used as the reference basis for the compression of the subsequent p- and b-frames of the video) and b) usually results in a small set of keyframes that is comparable in size with the collection of keyframes extracted by the InVID method, as reported in Table 1.5 and shown in the example of Fig. 1.14 below (left column). The least competitive one was the first alternative that extracts 1 keyframe per second. This method results in a keyframe-based representation of the video with high amount of redundant information (due to the occurrence of near-duplicate frames) and limited usefulness when the need is to quickly discover the video content.

The above described findings are illustrated in the example Fig. 1.14 which shows the extracted keyframe collections by the three tested approaches for the submitted video #7. Once again, the top left corresponds to the InVID method, the bottom left corresponds to the second alternative and the right-sided one corresponds to the first alternative. As can be seen, the latest one offers a very detailed and complete representation of the video content; however, several keyframes exhibit high visual resemblance, thus resulting in significant information redundancy which, in case of long videos, makes the discovery of the video content a time-consuming process. On the contrary, the left-sided keyframe collections provide a concise but also complete summary of the video content, as they contain all the key parts of the

Table 1.5: The votes of the DW journalist regarding the tested approaches for video keyframe extraction and keyframe-based reverse video search.

		Q1: extracted keyframes	Q2: concise and complete					Q3: helps the most in reverse search				
	Method		1	2	3	4	5	1	2	3	4	5
Video #1	InVID	41				X					X	
	Alt. #1	150		X					X			
	Alt. #2	31				X						X
Video #2	InVID	20					X				X	
	Alt. #1	81		X					X			
	Alt. #2	18					X				X	
Video #3	InVID	6					X				X	
	Alt. #1	20		X					X			
	Alt. #2	7					X				X	
Video #4	InVID	6				X					X	
	Alt. #1	25			X					X		
	Alt. #2	9				X					X	
Video #5	InVID	42				X					X	
	Alt. #1	153	X					X				
	Alt. #2	68		X				X				
Video #6	InVID	14			X					X		
	Alt. #1	52		X				X				
	Alt. #2	21			X					X		
Video #7	InVID	6					X				X	
	Alt. #1	26		X					X			
	Alt. #2	8				X					X	
Video #8	InVID	36					X					X
	Alt. #1	139	X						X			
	Alt. #2	53				X						X
Video #9	InVID	10		X				X				
	Alt. #1	54			X						X	
	Alt. #2	15				X					X	
Video #10	InVID	20				X		X				
	Alt. #1	64		X				X				
	Alt. #2	17				X		X				

presented story. The collection generated by the second alternative (at the bottom left of Fig. 1.14) includes a couple of near-duplicate frames, and thus was voted as slightly worse that then collection produced by the InVID approach.

As an overall comment, the keyframe selection strategy of the second alternative, in combination with the competitive performance that the InVID method exhibits in most cases, indicates that the developed algorithm for video sub-shot fragmentation and keyframe selection is highly efficient in extracting a set of keyframes that represent the visual content of the video without missing any important pieces of information, with the least amount of frames.

In terms of keyframe-based reverse search for quickly finding near-duplicates of the submitted videos on the Web, the InVID approach and the second alternative were voted as equally performing ones in 7 cases. Moreover, the InVID method was

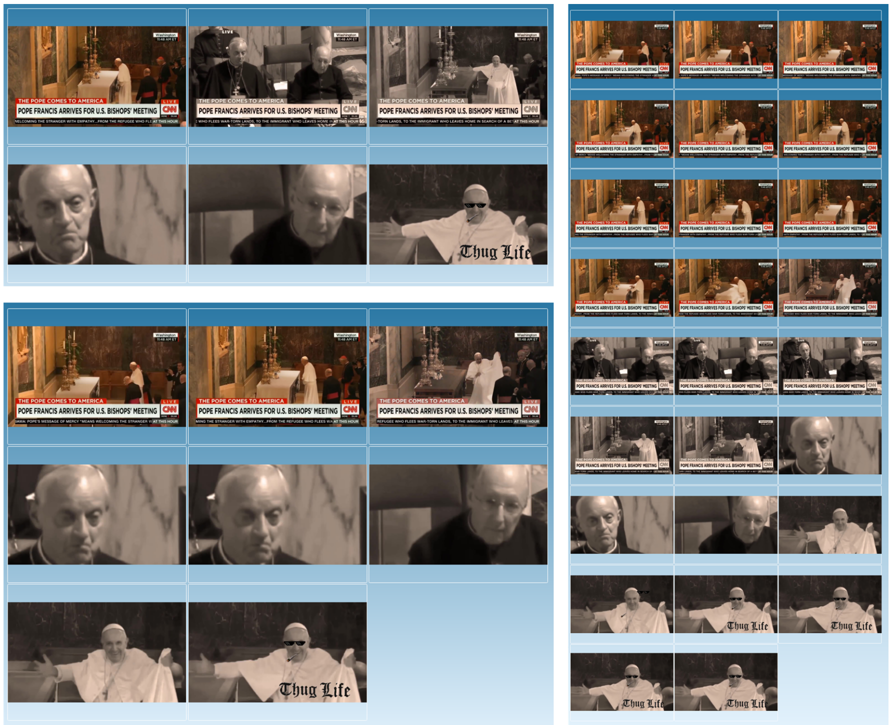

THE POPE COMES TO AMERICA
POPE FRANCIS ARRIVES FOR U.S. BISHOPS' MEETING
CNN
Thug Life

Fig. 1.14: The keyframe collections generated for a DW-selected video by the three tested approaches. The top left corresponds to the InVID method, the bottom left corresponds to the second alternative and the right-sided corresponds to the first alternative.

the best performing one in 1 case and the second best performing one in 2 cases. The second alternative was voted as the best one in 2 cases, while the first alternative was marked as the less efficient one in all tested cases. The later can be explained by the fact that, even providing a very fine-grained representation of the video content, this collection increases the amount of the time and effort needed to discover the keyframe collection and select the most appropriate keyframes for performing the keyframe-based reverse video search on the Web.

These findings are aligned to the ones extracted regarding the conciseness and completeness of the generated keyframe collections, and indicate the InVID method and the second alternative as the best choices for performing a fragment-level reverse search of a video on the Web.

Summing up the collected feedback regarding the competence of the developed video fragmentation approach for creating a concise and complete summary of the video content, and the appropriateness of this visual summary for supporting the task of video verification, we reach the conclusion that this technology is the best trade-off between two desirable but, to some extent, incompatible features. It results

Table 1.6: The submitted videos by the AFP and DW journalists for evaluating the InVID and the two alternative methods for video keyframe extraction and keyframe-based reverse video search.

#	**AFP journalist**	**DW journalist**
1	https://www.youtube.com/watch?v=GhxqIITtTtU	https://www.youtube.com/watch?v=okvoLbHlaVA
2	https://www.youtube.com/watch?v=oKQiTUjHlQ4	https://www.youtube.com/watch?v=ziOvZSUwU_c
3	https://www.facebook.com/Oker.Turgut/videos/1708996762482817/	https://twitter.com/AZeckenbiss/status/1033790392037199873
4	https://twitter.com/kengarex/status/1003749477583413249	https://twitter.com/JorgeaHurtado/status/1018125444158279682
5	https://www.youtube.com/watch?v=sza-j0nubNw	https://www.facebook.com/nafisa.alharazi/videos/10156699747657790/
6	https://twitter.com/tprincedelamour/status/843421609159544836	https://www.facebook.com/goodshitgoOdsHitthatssomegoodshitrightthere/videos/347521802658077/
7	https://www.youtube.com/watch?v=r5aBqCniQyw	https://www.youtube.com/watch?v=szKPipLRFsM
8	https://video.twimg.com/ext_tw_video/876820481919397889/pu/vid/360x640/VWTPEvrV8vVJFf4d.mp4	https://www.youtube.com/watch?v=BU9YAHigNx8
9	Local copy of the Thailand cave rescue video	https://www.youtube.com/watch?v=DeUVsmWji8g
10	https://www.youtube.com/watch?v=UTeqpMQKZaY	https://www.youtube.com/watch?v=-sWZuykJy9Q

in keyframe collections that adequately maintain the visual details of the video content and can be highly-valued for evidence-based video authentication or debunking through the visual inspection of such details (e.g. signs, labels, business marks, car plates, etc.), thus being aligned to the AFP journalist's focus of interest. Moreover, it secures a concise representation of the presented story that allows quick discovery of the video content and its verification through a sufficiently fine-grained, fragment-level search for finding near-duplicates of the video on the Web, thus meeting the DW journalist's demand.

1.5 Conclusions and Future Work

Video content captured by amateurs and shared via social media platforms constitutes a valuable source of information, especially in the case where these amateurs are eyewitnesses of a breaking or evolving story. Driven by this reality, journalists and investigators alike are constantly searching these platforms to find media recordings of breaking events. Nevertheless this rise of information diffusion via social networks came along with a rise in fake news, i.e. the intentional misinformation or disinformation to mislead people about a person, event or situation. Hence, the publicly shared user-generated video comes into question and needs to be verified before being used by a journalist for reporting the story.

One of the easiest ways to produce fake news (such fakes are known as "easy fakes" in the media verification community) is based on the reuse of a video from an earlier circumstance with the assertion that it presents a current event, with the aim to deliberately misguide the viewers about the event. To detect and demystify such a fake the investigator needs to identify the original video by looking for prior instances of it on the Web. To support the reverse video search process several tools have been developed over the last years; however they introduce some limitations that relate to a) the use of a (usually) limited group of video thumbnails provided by the platforms that host the video, b) the time-demanding extraction of video frames for performing reverse image search, c) the searching for near-duplicates within closed and restricted collections of videos, and d) the inability to handle local copies of a video from the user's machine.

Driven by the current state of the art on tools and methods for reverse video search on the Web, in InVID we designed and developed a method that decomposes a single-shot video (such as the majority of UGVs) into visually and temporally coherent parts called sub-shots, and we integrated this method into a web-based interactive tool that allows the fine-grained reverse search of a given video (either found online or locally stored on the user's machine) on the Web. This search is based on the extraction of a set of keyframes that adequately represent and summarize the video content, and the use of these keyframes for performing a fragment-level web-based search for near-duplicates of the video.

To give an overall view of this type of fake news and of the existing solutions for addressing it, this chapter: discussed the main characteristics of this fake (Section 1.1); provided an overview of current methods for video fragmentation and tools for reverse video search on the Web (Section 1.2); presented on two state-of-the-art approaches for the temporal decomposition of videos into sub-shots (the most suitable granularity when dealing with UGVs) and a web application that facilitates the quick identification of near-duplicates of a given video on the Web (Section 1.3); and described the conducted experimental evaluations concerning the performance of the aforementioned technologies (Section 1.4). The reported findings indicate the competitive performance of the developed algorithms for video sub-shot fragmentation compared to other state-of-the-art approaches, highlight the capability of the technique that relies on visual coherence to produce a concise and complete keyframe-based summary of the video content, and point out the competence of the InVID tool for video reverse search to facilitate the quick and effective discovery of near-duplicates of a video on the Web.

Regarding the future outlook of the presented technologies, motivated by the adoption and use of the developed web application for reverse video search by hundreds of users on a daily basis (through its integration into the InVID Verification Plugin[19]), our work will focus on: a) the user-based evaluation of the efficiency of

[19] Available at: http://www.invid-project.eu/verify/

the motion-based method of Section 1.3.1.2 to produce a comprehensive and thorough keyframe-based summary of the video content; b) the possibility to combine the algorithms of Sections 1.3.1.1 and 1.3.1.2 in order to exploit the fragmentation accuracy of the latter one and the visual discrimination efficiency of the former one (especially on the keyframe selection part of the process); c) the exploitation of the performance of modern deep-network architectures (such as DCNNs and LSTMs) for advancing the accuracy of the video fragmentation process; and d) the further improvement of the keyframe selection process to minimize the possibility of extracting black on blurred video frames of limited usability for the user, thus aiming to an overall amelioration of the tool's effectiveness.

1.6 Acknowledgments

The work reported in this chapter was supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-687786 InVID and H2020-732665 EMMA.

References

1. Abdollahian, G., Taskiran, C.M., Pizlo, Z., Delp, E.J.: Camera motion-based analysis of user generated video. IEEE Transactions on Multimedia **12**(1), 28–41 (2010). DOI 10.1109/TMM.2009.2036286
2. Apostolidis, E., Mezaris, V.: Fast shot segmentation combining global and local visual descriptors. In: Proc. of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6583–6587 (2014)
3. Apostolidis, K., Apostolidis, E., Mezaris, V.: A motion-driven approach for fine-grained temporal segmentation of user-generated videos. In: K. Schoeffmann, T.H. Chalidabhongse, C.W. Ngo, S. Aramvith, N.E. O'Connor, Y.S. Ho, M. Gabbouj, A. Elgammal (eds.) MultiMedia Modeling, pp. 29–41. Springer International Publishing, Cham (2018)
4. Bai, L., Hu, Y., Lao, S., Smeaton, A.F., O'Connor, N.E.: Automatic summarization of rushes video using bipartite graphs. Multimedia Tools Appl. **49**(1), 63–80 (2010). DOI 10.1007/s11042-009-0398-1. URL `http://dx.doi.org/10.1007/s11042-009-0398-1`
5. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). Computer Vision and Image Understanding **110**(3), 346–359 (2008). DOI 10.1016/j.cviu.2007.09.014. URL `http://dx.doi.org/10.1016/j.cviu.2007.09.014`
6. Benois-Pineau, J., Lovell, B.C., Andrews, R.J.: Motion Estimation in Colour Image Sequences, pp. 377–395. Springer New York, New York, NY (2013). DOI 10.1007/978-1-4419-6190-7_11. URL `http://dx.doi.org/10.1007/978-1-4419-6190-7_11`
7. Bouguet, J.Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation **5**(1-10), 4 (2001)
8. Chu, W.T., Chuang, P.C., Yu, J.Y.: Video copy detection based on bag of trajectory and two-level approximate sequence. In: Matching, Proc. of IPPR Conference on Computer Vision, Graphics, and Image Processing Conference (2010)
9. Cooray, S.H., Bredin, H., Xu, L.Q., O'Connor, N.E.: An interactive and multi-level framework for summarising user generated videos. In: Proc. of the 17th ACM International Conference on Multimedia, MM '09, pp. 685–688. ACM, New York, NY, USA (2009). DOI 10.1145/1631272.1631388. URL `http://doi.acm.org/10.1145/1631272.1631388`
10. Cooray, S.H., O'Connor, N.E.: Identifying an efficient and robust sub-shot segmentation method for home movie summarisation. In: 2010 10th International Conference on Intelligent Systems Design and Applications, pp. 1287–1292 (2010). DOI 10.1109/ISDA.2010.5687086
11. Cricri, F., Dabov, K., Curcio, I.D.D., Mate, S., Gabbouj, M.: Multimodal event detection in user generated videos. In: 2011 IEEE International Symposium on Multimedia, pp. 263–270 (2011). DOI 10.1109/ISM.2011.49
12. Dumont, E., Merialdo, B., Essid, S., Bailer, W., et al.: Rushes video summarization using a collaborative approach. In: TRECVID 2008, ACM International Conference on Multimedia Information Retrieval 2008, October 27-November 01, 2008, Vancouver, BC, Canada. Vancouver, CANADA (2008). DOI http://doi.acm.org/10.1145/1463563.1463579. URL `http://www.eurecom.fr/publication/2576`
13. Durik, M., Benois-Pineau, J.: Robust motion characterisation for video indexing based on MPEG2 optical flow. In: International Workshop on Content-Based Multimedia Indexing, CBMI01, pp. 57–64 (2001)
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. ACM Communications **24**(6), 381–395 (1981). DOI 10.1145/358669.358692. URL `http://doi.acm.org/10.1145/358669.358692`
15. González-Díaz, I., Martínez-Cortés, T., Gallardo-Antolín, A., Díaz-de María, F.: Temporal segmentation and keyframe selection methods for user-generated video search-based annotation. Expert Syst. Appl. **42**(1), 488–502 (2015). DOI 10.1016/j.eswa.2014.08.001. URL `http://dx.doi.org/10.1016/j.eswa.2014.08.001`
16. Grana, C., Cucchiara, R.: Sub-shot summarization for MPEG-7 based fast browsing. In: Post-Proc. of the Second Italian Research Conference on Digital Library Management Systems (IRCDL 2006), Padova, 27th January 2006 [16], pp. 80–84

17. Guo, Y., Xu, Q., Sun, S., Luo, X., Sbert, M.: Selecting video key frames based on relative entropy and the extreme studentized deviate test. Entropy **18**(3), 73 (2016). URL `http://dblp.uni-trier.de/db/journals/entropy/entropy18.html#GuoXSLS16a`
18. Haller, M., et al.: A generic approach for motion-based video parsing. In: 15th European Signal Processing Conference, pp. 713–717 (2007)
19. Karaman, S., Benois-Pineau, J., Dovgalecs, V., Mégret, R., Pinquier, J., André-Obrecht, R., Gaëstel, Y., Dartigues, J.F.: Hierarchical hidden markov model in detecting activities of daily living in wearable videos for studies of dementia. Multimedia Tools and Applications **69**(3), 743–771 (2014). DOI 10.1007/s11042-012-1117-x. URL `http://dx.doi.org/10.1007/s11042-012-1117-x`
20. Kasutani, E., Yamada, A.: The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: Proc. 2001 International Conference on Image Processing (Cat. No.01CH37205), vol. 1, pp. 674–677 vol.1 (2001). DOI 10.1109/ICIP.2001.959135
21. Kelm, P., Schmiedeke, S., Sikora, T.: Feature-based video key frame extraction for low quality video sequences. In: 2009 10th Workshop on Image Analysis for Multimedia Interactive Services, pp. 25–28 (2009). DOI 10.1109/WIAMIS.2009.5031423
22. Kim, J.G., Chang, H.S., Kim, J., Kim, H.M.: Efficient camera motion characterization for mpeg video indexing. In: 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proc.. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532), vol. 2, pp. 1171–1174 vol.2 (2000). DOI 10.1109/ICME.2000.871569
23. Koprinska, I., Carrato, S.: Video segmentation of mpeg compressed data. In: 1998 IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No.98EX196), vol. 2, pp. 243–246 vol.2 (1998). DOI 10.1109/ICECS.1998.814872
24. Lan, D.J., Ma, Y.F., Zhang, H.J.: A novel motion-based representation for video mining. In: Proc. of the 2003 International Conference on Multimedia and Expo (ICME '03), vol. 3, pp. III–469–72 vol.3 (2003). DOI 10.1109/ICME.2003.1221350
25. Liu, Y., Liu, Y., Ren, T., Chan, K.: Rushes video summarization using audio-visual information and sequence alignment. In: Proc. of the 2Nd ACM TRECVid Video Summarization Workshop, TVS '08, pp. 114–118. ACM, New York, NY, USA (2008). DOI 10.1145/1463563.1463584. URL `http://doi.acm.org/10.1145/1463563.1463584`
26. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the 7th IEEE International Conference on Comp. Vis., vol. 2, pp. 1150–1157 (1999)
27. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, pp. 2714–2721. IEEE Computer Society, Washington, DC, USA (2013). DOI 10.1109/CVPR.2013.350. URL `http://dx.doi.org/10.1109/CVPR.2013.350`
28. Luo, J., Papin, C., Costello, K.: Towards extracting semantically meaningful key frames from personal video clips: From humans to computers. IEEE Transactions Cir. and Sys. for Video Technol. **19**(2), 289–301 (2009). DOI 10.1109/TCSVT.2008.2009241. URL `http://dx.doi.org/10.1109/TCSVT.2008.2009241`
29. Mei, T., Tang, L.X., Tang, J., Hua, X.S.: Near-lossless semantic video summarization and its applications to video analysis. ACM Transactions Multimedia Comput. Commun. Appl. **9**(3), 16:1–16:23 (2013). DOI 10.1145/2487268.2487269. URL `http://doi.acm.org/10.1145/2487268.2487269`
30. Mohanta, P.P., Saha, S.K., Chanda, B.: Detection of representative frames of a shot using multivariate wald-wolfowitz test. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4 (2008). DOI 10.1109/ICPR.2008.4761403
31. Ngo, C.W., Ma, Y.F., Zhang, H.J.: Video summarization and scene detection by graph modeling. IEEE Transactions on Circuits and Systems for Video Technology **15**(2), 296–305 (2005). DOI 10.1109/TCSVT.2004.841694
32. Ngo, C.W., Pong, T.C., Zhang, H.J.: Motion analysis and segmentation through spatio-temporal slices processing. IEEE Transactions on Image Processing **12**(3), 341–355 (2003). DOI 10.1109/TIP.2003.809020

33. Nitta, N., Babaguchi, N.: Content analysis for home videos. ITE Transactions on Media Technology and Applications **1**(2), 91–100 (2013). DOI 10.3169/mta.1.91
34. Ojutkangas, O., Peltola, J., Järvinen, S.: Location Based Abstraction of User Generated Mobile Videos, pp. 295–306. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-30419-4_25. URL `http://dx.doi.org/10.1007/978-3-642-30419-4_25`
35. Omidyeganeh, M., Ghaemmaghami, S., Shirmohammadi, S.: Video keyframe analysis using a segment-based statistical metric in a visually sensitive parametric space. IEEE Transactions on Image Processing **20**(10), 2730–2737 (2011). DOI 10.1109/TIP.2011.2143421
36. Pan, C.M., Chuang, Y.Y., Hsu, W.H.: NTU TRECVID-2007 Fast Rushes Summarization System. In: Proc. of the International Workshop on TRECVID Video Summarization, TVS '07, pp. 74–78. ACM, New York, NY, USA (2007). DOI 10.1145/1290031.1290045. URL `http://doi.acm.org/10.1145/1290031.1290045`
37. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proc. of the IEEE International Conference on Computer Vision (ICCV 2011), pp. 2564–2571 (2011)
38. Shi, J., et al.: Good features to track. In: Proc. of the IEEE Conference on Comp. Vis. and Patt. Recogn., pp. 593–600 (1994)
39. Teyssou, D., Leung, J.M., Apostolidis, E., Apostolidis, K., Papadopoulos, S., Zampoglou, M., Papadopoulou, O., Mezaris, V.: The invid plug-in: Web video verification on the browser. In: Proc. of the First International Workshop on Multimedia Verification, MuVer '17, pp. 23–30. ACM, New York, NY, USA (2017). DOI 10.1145/3132384.3132387. URL `http://doi.acm.org/10.1145/3132384.3132387`
40. Wang, G., Seo, B., Zimmermann, R.: Motch: An automatic motion type characterization system for sensor-rich videos. In: Proc. of the 20th ACM International Conference on Multimedia, MM '12, pp. 1319–1320. ACM, New York, NY, USA (2012). DOI 10.1145/2393347.2396462. URL `http://doi.acm.org/10.1145/2393347.2396462`
41. Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J.M., Singh, V.: Gaze-enabled egocentric video summarization via constrained submodular maximization. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [41], pp. 2235–2244. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#XuMLWRS15`