A perceptual scheme for fully automatic video shot boundary detection

Murat Birinci*, Serkan Kiranyaz

Department of Signal Processing, Tampere University of Technology, Finland

ABSTRACT

In this paper, we propose a novel and robust modus operandi for fast and accurate shot boundary detection where the whole design philosophy is based on human perceptual rules and the well-known “Information Seeking Mantra”. By adopting a top–down approach, redundant video processing is avoided and furthermore elegant shot boundary detection accuracy is obtained under significantly low computational costs. Objects within shots are detected via local image features and used for revealing visual discontinuities among shots. The proposed method can be used for detecting all types of gradual transitions as well as abrupt changes. Another important feature is that the proposed method is fully generic, which can be applied to any video content without requiring any training or tuning in advance. Furthermore, it allows a user interaction to direct the SBD process to the user’s “Region of Interest” or to stop it once satisfactory results are obtained. Experimental results demonstrate that the proposed algorithm achieves superior computational times compared to the state-of-art methods without sacrificing performance.

1. Introduction

The amount of available video content is growing exponentially with the development in content creation technology. Moreover, content sharing has become immensely popular, enabling every individual to access a vast amount of video content. YouTube, globally the 3rd most popular website [1], announced that more than 72 h of video are uploaded to the website every minute, and more than 4 billion h of video are watched every month [2]. It is therefore inevitable that such amount of visual information and growth demands efficient content management tools.

In [3], Thompson et al. defined a video shot as the smallest unit of visual information captured at one time by a camera that shows a certain action or event. Therefore, in order to capture the entire visual content properly and attain a complete grasp of the video, shot detection is a fundamental step of content based video analysis. Whereas there is a wealth of research on shot boundary detection (SBD), the main bottleneck of the problem is the relative difficulty in detecting gradual transitions (GT) between shots compared to the detection of the abrupt changes, i.e. abrupt cuts (AC). Even though gradual transitions used to appear more frequently in professionally edited videos, nowadays even personal cameras and camera-equipped cell phones are capable of editing videos to comprise such transitions. Therefore, a proficient SBD algorithm should be able to handle gradual shot transitions regardless of their nature (dissolve, fade, wipe etc.), as well as abrupt changes. Whereas, any SBD algorithm stems from the same assumption that there is a visual discontinuity between consecutive shots, most of them
coefficients and motion vectors \[6\] further analyzed different compressed domain algorithms that utilize compressed domain features such as DCT coefficients and motion vectors \[6–12\]. However, they concluded that, despite being computationally efficient, their performance levels were even below histogram based approaches. In a more recent study, Teng \[13\] proposed a method based on texture features extracted from non-overlapping blocks, and classified video frames via Support Vector Machines (SVM) to detect shot boundaries based on cosine distances. Another classifier based method is presented in \[14\] utilizing the U component of the YUV histogram and classifying the difference curves using Particle Swarm Optimization (PSO). In \[15\], Hanjalic provided a thorough analysis of the previous methods and proposed a probabilistic method based on YUV color components from non-overlapping blocks. The method provides satisfactory results for AC and dissolves; however, it requires a specific implementation for each individual type of GTs. Another extensive evaluation came as a result of TRECVID, which had an activity track for SB from 2001 to 2007 joining 57 different research groups in order to determine the best approaches \[16\]. Whereas the idea of various algorithms working on a common dataset with common scoring metrics provides the means for objective evaluation, it also brings in a clear advantage for machine learning algorithms since the whole TRECVID dataset (development + testing) is composed of vastly similar content which in return inevitably bias the overall results. The fact that 9 out of top 10 performing groups utilize machine learning algorithms is a clear indication of such bias where the algorithms are specifically tuned for the development data which is highly similar to the test data. Moreover, the fact that 6 out of top 10 performing algorithms using flash detectors, which is a very specific case commonly appearing in news videos (which also constitutes most of the TRECVID dataset), is also a clear indication that the competing methods were tuned to perform only for the specific TRECVID dataset and seriously questions the applicability of such methods to generic video content.

In \[17\], Boccignone et al. proposed a perceptual stand point to the SB problem and suggested that “visual attention” is the key to detect scene changes. They extracted the focus of attention (FOA) points from each frame, where the variations in the consistency of FOA revealed shot boundaries. The motivation of the paper, as the authors stated, was not only to achieve high SB performance, but also to introduce a different angle for the problem. However, their results were still comparable to the state-of-art algorithms. Another high level analysis was proposed by Park et al. \[18\] where they made use of object recognition techniques, namely Scale Invariant Feature Transform (SIFT) \[19\]. They proposed that the objects or background do not differ significantly within the same shot, whereas a notable difference occurs across shot boundaries. In order to measure such dissimilarity, they extracted and matched interest points (SIFT) between consecutive frames and monitored the variation in the number of matches in order to detect ACs. However, their method failed to detect GTs since the visual similarity between two consecutive frames is significantly high during a GT, which in return yields a high number of matches. In order to cure this deficiency, the authors additionally compared every \(Nth\) frame in order to attain sufficiently high content change for GT detection. This method can tell that a shot transition occurred somewhere between those \(N\) frames, but it still fails to determine its exact location. However, even though the method suffered from the heavy computation of the SIFT that has to be computed for each frame, it can still be regarded as innovative due to its incorporation of objects and object recognition algorithms in order to bring in a higher level standpoint to SB problem. Moreover, similar to the work in \[16\] that used FOA in order to extract the essential information throughout the frame, utilization of local image features aims to achieve the same goal by detecting objects through such invariant (to the scale, rotation and translation) points and the features computed over the local regions around them.

There have been several methods concerning local image features prior to SIFT; however, it has been regarded as a milestone due to its remarkably high performance and stability under relatively reasonable computational costs. One of the oldest methods, yet still popular, is the Harris corner detector \[20\] that is based on the autocorrelation matrix. Whereas being translation and rotation invariant, Harris (corner) points are not scale invariant. The scale invariant version of the Harris detector was proposed by Lindeberg \[21\], which is also referred as Harris–Laplace detector. Mikolajczyk and Schmid further improved this method to provide an affine invariant detector called Harris–Affine \[22\]. Lowe in Ref. \[19\] proposed SIFT, which uses Difference of Gaussians (DoG) as an approximation to Laplacian of Gaussians (LoG) and their local maxima to detect scale and rotation invariant keypoints. Bay et al. \[19\] used integral images to detect keypoints in close to real time \[23\]. Integral images were already known to be used for fast computation of Haar wavelets. However, Bay et al. used those to approximate the Hessian matrix, which they claimed to be more stable and repeatable than Harris-based detectors. There are numerous adaptations and successors of the aforementioned detectors; however, while choosing the appropriate local feature, typically a trade-off has to be made between efficiency on one hand, and accuracy or repeatability on the other. Harding and Robertson \[24\] compared six keypoint detection methods (namely SIFT, Harris–Laplace, SURF, MSER \[25\], FAST \[26,27\] and Kadir–Brady Salienty \[28\]) with two visual saliency methods and concluded that SURF has the highest correspondence surpassing other methods by 15% higher overlap. For a more comprehensive study on keypoint detectors, the reader is referred to the survey by Tuyltaelaars et al. \[29\].
Despite the wealth of research in local image features and their relevance to SBD problem, utilization of these features in SBD has been surprisingly limited. Huang et al. [30] reported that the work proposed in [18] is the first method that employed keypoint-based analysis. They further proposed a parallel approach to [18] where they used a relatively light descriptor of their own design extracted around Harris keypoints, i.e. Contrast Context Histogram (CCH) [31], in order to avoid the computational load of SIFT feature extraction. Additionally, they performed a more in depth analysis of frame similarities in order to detect both ACs and GTs with high accuracy. They initially compare every adjacent frame and observe the variation in the number of matched keypoints, where they take every local minimum as a candidate shot boundary. They assume that the local maxima before and after the candidate transitions are the possible start and end frames of the GT. They further require those local maxima to be followed (and preceded) by a stable number of matches in order to claim it as a shot boundary. This also allows them to determine exact transition intervals. However, this method still considers the frame similarity only between adjacent frames. As we mentioned earlier adjacent frames have significantly high visual similarity, hence such an analysis can easily lead to false negatives or even false positives. In order to avoid such deficiency, they followed a similar strategy to [18] and compared frames that are certain distance apart, namely the frames at the beginning and end of the candidate transition interval. If those frames are found to be similar they regard it as a false alarm, otherwise the final decision is given as a shot boundary. Whereas the authors reported significantly higher accuracy compared to [18], computational cost is an obvious drawback of the algorithm – not because of the underlying feature detector and descriptor, but due to the manner the algorithms searches for the boundaries. The provided per-frame time analyses are encouraging, thanks to the low computational cost of CCH. However, the total processing time of the video will significantly be affected by the employed searching algorithm, where every single frame in the video is processed and matched to its neighboring frames, and on top of that, the interval around every single local minimum is inspected for a possible shot boundary. In other words, the reported per-frame execution times will undergo considerable amount of repetitions resulting in excessive computation times for videos. In order to exemplify this, consider Fig. 1, which shows the variation of the number of matched keypoints between adjacent frames in a video. It is obvious that an innumerable number of local minima exists due to the oscillations in the number of matches within a shot. Moreover, no significant changes occur for certain shot boundaries (particularly GT), verifying the fact that adjacent frames have significantly high visual similarity and seeking shot boundaries through such an inspection will inevitably lead to erroneous results.

Following the advancements in content based image and video analysis, state-of-the-art SBD algorithms are inclined to use high-level descriptors such as object/scene detection, visual attention analysis, rather than relying on low-level descriptors. However, whereas almost all previous attempts focused on the discriminative power of underlying features, few seem to have realized the importance of the employed search scheme. We believe in the search for shot boundaries, how you search is as important as what you search for. Therefore, it is of decisive importance that a proficient search scheme is followed in order to find the shot boundaries accurately and efficiently. Humans enjoy an extraordinary ability to recognize and interpret visual similarities, differences and alterations. Hence, understanding human visual perception and how humans perform visual search will lead to an effective and competent search scheme. There have been numerous studies on human visual perception; however, our understanding of visual perception comes substantially from the Gestalt Psychology [32]. By taking a holistic standpoint, Gestalism focuses on the emergent properties of visual stimuli rather than considering them individually. Following its well-known rallying cry, “The whole is greater than the sum of its parts,” Gestalism provides a set of perceptual rules (Prägnanz) in order to explain that perception cannot be reduced to parts or even to piece-wise relations among parts. Such a top-down manner has been neglected in SBD methods so far, since almost all of the previous approaches are designed in a bottom-up fashion to build on the information that is based on the relation between consecutive frames – which are basically the parts of the video – instead of considering the fact that transitions naturally emerge when the video is considered as a whole.

In addition to perceptual psychology, the field of Human-Computer Interaction is also particularly interested in the same question in order to understand “How humans perform visual search?” and reflect the answer to user interaction designs in order to provide effective means of search tools to users. The well-known “Information Seeking Mantra” was proposed accordingly by Ben Shneiderman in order to provide better means of information visualization [33]. In other words, it guides users to the data they are searching for in a fast and efficient way. In an abstract level, the Mantra abides by the following principle: Overview first, than zoom and filter. The overview phase lets the user gain an overall understanding of the data such as distribution, internal relations, etc. Then, the user zooms in to the particular item of interest and filters
out uninteresting items. Even though no perceptual roots were mentioned in the proposal, the Mantra also agrees with Gestaltism by its nature, where the whole visual perception is assumed to be a top-down process. Such a search scheme is particularly suitable for SBD since shot boundaries emerge as we take a broader view to the video instead of taking a close up view, i.e. consider only adjacent frames. In order to illustrate this further, consider that all frames of a video are arranged as a sequence of images in temporal order. When we try to find the shot boundaries (visually) we do not start from the first frame and proceed frame-by-frame until we reach visually different frames to judge as the boundary. Instead, we have a broader look at the images and recognize the difference between two shots and gradually narrow our focus down to the particular location where the transition occurs. In other words, we first overview the video, and then zoom in to the boundary filtering out the redundant frames. This is the complete opposite of the manner that [30] searches for the boundaries, where every single pair of adjacent frames is compared and then zoomed out at every suspicion of a boundary.

In [34], Feng et al. followed a similar “overview first, then zoom in” mind-set in utilizing the encoded bitstream in order to detect ACs. They compare the consecutive I-Frames and continue analyzing that particular GoP (Group of Pictures) if they notice a difference by sampling the GoP further via selecting P and B-Frames. However, despite getting the inkling of the idea, they failed to grasp the importance of it since their intention was purely to exploit the encoding algorithm. In [40] we have proposed a similar approach in order to obtain the overview of the video. But, instead of selecting the I-Frames from the encoded bitstream, we performed uniform temporal sampling since the number and frequency of I-Frames is an encoding decision that mainly depend on the application (streaming, storage, mobile etc.), not the content. If the quality is of biggest concern, I-Frames might be too close to each other, or if the video is intended for streaming they might be too infrequent to save bandwidth. However, such sampling is prone to errors in case of a GT since a sample can easily be selected among the frames of a GT. Authors realized this weakness in [34] and claim to detect ACs only, however we have addressed this issue in Section 2.

Multi-Resolution Analysis (MRA) has also been employed by several methods mostly in order to be able to detect GTs [35–39]. MRA analyzes the video under several “resolutions”, i.e. several levels of focus, such that high resolution provides high precision whereas low resolution enables to catch the GTs of various durations. In analogy with the aforementioned methods in [34,40], MRA samples the video with several frequencies varying from frame-by-frame to “overview”. However, that results in processing every frame in the video and analyzing every possible pair-comparison resulting in a lot of redundant computation. The method in [18] may also be considered as MRA where the authors examined only two resolutions, i.e. frame-by-frame and \( N \) frames apart.\footnote{Please cite this article as: M. Birinci, S. Kiranyaz, A perceptual scheme for fully automatic video shot boundary detection, Signal Processing-Image Communication (2014), \textit{http://dx.doi.org/10.1016/j.image.2013.12.003}}

As mentioned above, we have proposed an earlier version of this work in [40], where we have also employed the “Overview first, than zoom and filter” principle. However, the aforementioned overview phase, i.e. uniform sampling, in [40] suffers in GT detection performance since a sample can easily be taken within a GT which in return may result in missing that boundary. This deficiency is discussed in detail in Section 2 together with the provided solution. Moreover, we have proposed a better similarity judgment through a more robust “similarity rate” definition and significantly decreased the time spent on feature matching via “Fast Approximate Nearest Neighbors” proposed by Muja and Lowe in [41].

In order to address the aforementioned drawbacks of the state-of-the-art algorithms and provide an efficient and accurate solution to the SBD problem, in this paper we propose a method that is modeled based on Shneiderman’s Information Seeking Mantra that employs local image features in order to reveal inter frame dissimilarities. The proposed algorithm incorporates the proven potency of local image features, and the effectively utilized top-down search scheme provides a fast and systematic way to locate shot boundaries avoiding any unnecessary feature extraction and feature matching. We further analyzed spatial distribution of keypoints in order to increase the similarity judgment performance, which enables us to adapt to the content and content changes more accurately. The primary objective above all is to design a generic and robust SBD technique, which neither requires nor relies on any training or tuning while showing a superior performance on any video content in a computationally efficient manner.

The rest of the paper is organized as follows. In Section 2 the proposed method is explained in detail together with the underlying feature extraction, spatial analysis and the top-down search scheme. Section 3 provides the performance evaluation of the proposed method and Section 4 concludes the paper.

2. The proposed SBD algorithm

Under the light of earlier discussion, the proposed algorithm is designed to overcome the limitations and deficiencies of the preceding SBD algorithms. Such improvements are achieved by taking a perceptual point of view under the supervision of information visualization tools. The proposed algorithm starts with overviewing the video and gradually zooms in wherever a shot boundary exists as illustrated in Fig. 2. In order to judge frame (dis)similarities, local image features and their spatial distribution are analyzed. An earlier version of the proposed work was briefly described in [40]. The following subsections provide details of the proposed algorithm with justification. First, we explain the employed search scheme, i.e. how we perform the search, without going into details of the underlying feature, i.e. what we search for. The latter is

![Fig. 2. Outline of the proposed SBD method.](image-url)
clarified in the next subsection by giving details of how visual similarity judgment is performed.

2.1. The top-down search scheme

Stemming from perceptual rules of Gestalt psychology, we designed a top-down SBD scheme that follows the aforementioned “Information Seeking Mantra”. Recall that the Mantra suggests a perceptual path for efficiently accessing the desired information: Overview first, than zoom and filter. Accordingly, by completely rejecting a frame-by-frame processing manner, we implemented a method that provides the overview of a video, i.e. roughly giving the locations of shot boundaries. Hence, at the end of the overview phase the algorithm provides the information of how many shots exist in the video and imprecise locations of their boundaries. Then, in the next step, the algorithm gradually zooms in to those locations in order to localize the boundaries precisely. Since the algorithm only zooms in wherever there is a boundary, unnecessary processing of video frames within any shot can naturally be avoided. In other words such massive number of “uninteresting” frames are filtered out from the search.

In order to obtain the overview of the video the video is uniformly sampled in temporal domain and successively sampled frames are compared for visual similarity. The algorithm concludes that a shot transition has occurred between those frames whenever a visual discontinuity is detected. The judgment of such visual discontinuity will be detailed in Section 2.2. Let us denote the $n$th frame of the video as $F(n)$. Then, for every $n = N, 2N, 3N \ldots F(n)$ is compared to $F(n-N)$ where $N$ is the temporal sampling period. With the proper choice of $N$, such sampling permits sufficient content change to occur and hence enables the system to detect both AC and GT. Fig. 3 shows the overview of the same video that is used to generate Fig. 1 with $N = 30$ (1 sec.). Whereas sharper variations “near” shot boundaries are clearly seen, only $\sim 3\%$ of the total frames are processed in order to acquire that information.

In order to classify those variations as boundaries, we have detected the peaks and required the peaks to be “deep enough” to be regarded as boundaries. Let $s$ be the set of similarities obtained in Overview such that $s_i$ denotes the similarity between $(i \times N)$th and $((i-1) \times N)$th frames. Then the detection of the boundaries is achieved via the function $\text{FIND\_BOUNDARIES}(s)$ as follows:

\begin{verbatim}
FUNCTION FIND_BOUNDARIES(s)
  1 for every $s_i, i=1,2,\ldots$
  2   if $s_i < s_{i-1}$ and $s_i < s_{i+1}$ then $s_i$ is a peak
  3   then $L \leftarrow i-1, R \leftarrow i+1$
  4   while $s_L < s_{i-1}$ and $s_R > s_{i+1}$ do
  5     if $s_L < s_{i-1} \cdot (L-L-1)$ and $s_R > s_{i+1} \cdot (R-R+1)$
  6     then $s_{i-1}$ or $s_{i+1}$ is a peak
  7     if $s_i < s_{i} \cdot T_{ov}$ or $s_i > s_{i} \cdot T_{ov}$
  8     then zoom in to $F((i-1) \times N), F(i \times N)$
end
\end{verbatim}

where $T_{ov}$ is the threshold to judge how “deep” the peaks are.

Note that the choice of the sampling period $N$ is of decisive importance. Whereas a sparse under-sampling results in lower computational complexity, a reduced accuracy in return is inevitable especially for videos having many shots with short duration. This is due to the fact that if $N$ is too large, an entire shot may end up in between the sampled frames and, therefore, missed. Also high object and/or camera motion may easily yield false positives. On the other hand an oversampling with a very low $N$ value increases the computational cost and more importantly gets us closer to frame-by-frame analysis that we strive to avoid in the first place. Therefore, a reasonable and practical assumption for the minimum shot duration should be considered while deciding on $N$. Considering the definition of a shot (see Section 1) it should be long enough to comprise of a certain event or action. The selection of $N$ can also be left up to the encoding scheme as in [34] where the authors selected the I-Frames to sample the video and focus on that particular GoP if there is a noticeable change. However, as we have discussed in Section 1, the distance between two I-Frames is decided during encoding depending on the target application and does not reflect the content by any means.

It should be noted that this information, i.e. the number of shot boundaries and their approximate locations, might be sufficient for various applications; however, further analysis is needed for accurate localization of the shot boundaries. The next step of the proposed search scheme realizes that by zooming into the locations where significant discontinuities in visual similarity are observed during the overview phase. That is achieved by gradually decreasing the distance between the frames that are compared for similarity. As the distance decreases, the

![Fig. 3](image-url)  
Fig. 3. Overview of the video that is used to generate Fig. 1. Dashed lines denote shot boundaries.

![Fig. 4](image-url)  
Fig. 4. Overview and zoom in phases of the proposed search scheme.
change in frame similarity reveals not only the location of the shot boundary, but also the nature of it.

Consider the case in Fig. 4, where a shot boundary is detected between \( F(n) \) and \( F(n+N) \). Then, the algorithm gradually decreases the distance between frames and starts comparing \( F(n) \) to \( F(n+N-k) \) where \( k = 1, 2, 3... N-1 \). The variation in the similarity of frames as \( k \) approaches \( N-1 \) unveils the exact location and nature of the transition.

Fig. 5 illustrates how visual similarity between frames \( F(n) \) and \( F(n+N-k) \) changes as the algorithm zooms in (i.e. \( k: 1 \rightarrow N-1 \)). Two different cases are exemplified, namely for AC (Fig. 5a) and GT (Fig. 5b). In case of an AC, the visual similarity abruptly drops, also revealing the exact location of the transition. On the other hand, when there is a GT, frame similarity gradually diminishes. It is rather easy to detect and distinguish between AC and GT by simply monitoring the change in visual similarity (the number of matches in this case) at each distance. If the change is larger than a predefined threshold \( T_2 \), an AC is revealed with its exact location, otherwise it is a GT. Note that the algorithm zooms in to the interval between \( F(n) \) and \( F(n+N) \) if and only if a boundary is detected during the overview phase; otherwise, the frames in the interval are “filtered out” avoiding unnecessary feature extraction and matching.

2.1.1. Shifted tracing

Although the search scheme that is discussed so far is capable of achieving high accuracy with considerably low computational demand, it comprises an apparent imperfection. The fact that uniform temporal sampling is utilized “may” produce false negatives if \( F(n) \) is sampled among the frames within a GT. In that case, since the visual content changes gradually during that interval, the sampled frame will somewhat be similar to both the preceding and the succeeding shots; hence the aforementioned algorithm will fail to realize the shot boundary by filtering it out. In order to avoid such misjudgment, we propose an improved version of the overview phase. Fig. 6 depicts the proposed shifted tracing of the video where the same sampling and comparison procedure is applied with an \([N/2]\) frames shift. In other words, overview of the video is obtained twice with temporal shift of \([N/2]\) frames. This way, if a shot boundary is missed due to a GT during the first trace, it will be detected during the second one. Considering the insignificant computational weight of the overview phase, a significant improvement is achieved with such a minimal effort.

It should be noted that shifted tracing is not a blindfolded reiteration of the overview phase. As its purpose is to enhance the accuracy of single trace overview, it therefore, avoids any redundant recalculation that has already been carried out by the first trace. In other words unnecessary “zoom in” are avoided by simply ignoring any boundaries if they have already been detected by the other trace. Consequently, the shifted tracing allows a complete overview of the video, detecting every single shot boundary and eliminating the shortcomings arising from any GT and uniform sampling applied.

In addition to avoiding unnecessary processing of video frames, the proposed algorithm is also suitable for parallel processing by its nature which further enables significant performance improvement. Both the shifted traces in the overview phase and every single “zoom in” are independent processes, hence can be handled in parallel.

2.2. Frame similarity via local features

In order to judge whether two video frames belong to the same shot, the following test is performed: if the same objects are detected in two different frames, they are considered to belong to the same shot. Such a manner of similarity judgment also tackles the prominent problem of object and camera motions innately, since the objects will still be detected (either on the foreground or background) despite any object or camera motion assuming that the two frames are not excessively apart from each other in temporal domain. Still, the choice of image feature should be able to handle possible object deviations such as variations in scale, rotation, and translation.
Local image features that are invariant to those changes are thus utilized in order to match objects between frames. First, interest points are detected throughout the frames, and then descriptors around each point are extracted. The proposed SBD method is independent of the underlying point detector and descriptor, and in this work, SURF is chosen for both detection and description due to its high correspondence with human visual saliency, improved repeatability over other detectors and lower computational complexity (see Section 1). Finally, descriptors from two frames are matched against each other to find visual correspondence. However, in addition to feature extraction, another source of the computational load is feature matching particularly if a blunt linear search is utilized. In order to further reduce the overall computational cost, we employed “Fast Approximate Nearest Neighbors” [41] which has proven to speed up the matching process up to several orders of magnitude compared to linear search by using multiple randomized k-d trees. The algorithm was tested for SIFT descriptors and achieved significant performance improvement with minimal loss in accuracy. Similarly, during our experiments using SURF descriptor, no significant performance loss is observed despite the considerable decrease in computational cost.

As discussed in the previous section and depicted in Figs. 3 and 5, variations in the number of matches between frames reveal the location of the shot boundaries. However, it should be noted that the number of total interest points detected in a frame depends entirely on the content of that frame. The variations in substantial amount of matches are relatively informative; however, with the limited number of matches due to the limited number of keypoints, the reliability of such variations degrades significantly. Considering that the keypoints reflect the visual content of a video frame, the change in the content should be revealed regardless of the number of keypoints it is represented with. In order to achieve this, we normalize the number of matches with the total number of keypoints in both frames which gives us the degree of similarity between two frames. Consider the case where K and L are the number of keypoints extracted from $F(n)$ and $F(n+N)$, respectively, and $M$ is the number of matched keypoints between those frames. Then, the rate of similarity, $R$, between $F(n)$ and $F(n+N)$ can be formulated as:

$$R = \frac{2M}{K+L} \quad (1)$$

This phenomenon can easily be observed by comparing Figs. 3 and 8. Note that in Fig. 3 the variation around frame 500 can easily be mistaken as a boundary since it is comparable to real shot boundaries, whereas in Fig. 8 the variation in similarity rate is minor compared to the boundaries. Similarly the variation in the number of matches around frame 1650 may not be enough to detect it as a boundary, yet the change in similarity rate in Fig. 8 clearly signifies it as a shot boundary.

### 2.2.1. Spatial analysis of keypoints

Matching objects in order to reveal visual similarity is a well-reasoned perceptual approach; however, a comprehensive discussion has been made in [42] that merely matching individual local features is far from reflecting human perception. Again, following Gestalt’s rule of perception “the whole is greater than sum of its parts”, it is shown in [42] that matching complete objects is more (informative) than the sum of individually matched keypoints. Hence, following the aforementioned “Pragnanz”, certain perceptual constraints are imposed by considering keypoints’ spatial distribution. In other words their spatial proximity is taken into account and it is proposed that if two keypoints are spatially close to each other, it is highly unlikely that their corresponding matches are significantly isolated. This is due to the natural fact that the objects are solid and follow a slightly rigid motion within a shot. This is no longer valid for (accidental) matches between the frames from two distinct shots (e.g. see Fig. 7). Thus, whenever a match is found, their neighborhoods are matched against each other in order to validate the match and hence to avoid any potential false positives. Following this principle the number of both false positives and false negatives can be decreased considerably; and due to the nature of the imposed criteria, groups of matches emerge naturally instead of single individual matches as shown in Fig. 7b.

In addition to its undeniable improvement in matching performance, elimination of false negatives as in Fig. 7c particularly assists the detection of shot boundaries by providing sharper variations (i.e. deeper peaks) in the number of matches (thus, the rate of similarity) during shot transitions as shown in Fig. 8.

Huang et al. [30] also made use of the spatial information of keypoints in a similar manner. However, their analysis of spatial distribution is limited to matching adjacent frames only such that they simply limit the spatial displacement of possible matching keypoints to a certain number of pixels. In other words, a keypoint is not allowed to match another if their spatial locations are separated by more than a predefined distance threshold. However, such a limitation is reasonable only for neighboring frames due to the considerably limited content change among adjacent frames. For cases where frames at a certain temporal distance apart are compared for similarity (as in the overview phase of the proposed algorithm or the false alarm detection in [30]), such a restriction should definitely be avoided since any object or camera motion can easily violate this constraint.

Fig. 9 summarizes the whole algorithm visually. To sum up, the Overview phase uniformly samples the video by taking every Nth frame from the video and compares consecutive samples to obtain the similarities $s_i$. FIND_-BOUNDARIES function detects the intervals where a shot change has occurred by analyzing the change in $s$. Then the algorithm zooms in to each of these intervals and monitors how the similarity changes as the interval gradually narrows down. The nature of the change in similarity also reveals the nature of the transition, i.e. AC or GT.

### 3. Experimental results

In order to demonstrate the performance of the proposed algorithm and prove its improvements over the
state-of-the-art methods, we performed SBD experiments on two separate video databases: First set is the TRECVid 2005 SBD test set [16]. The dataset contains 12 videos (7 h, 744,604 frames) and has 4535 total transitions (60.8% AC, 39.2% GT). Even though we discussed in Section 1 that the strong similarity between the development and test sets of this dataset induces a strong bias to the results especially when machine learning algorithms are considered, we provide our results for the sake of completeness since it is still considered as one of the benchmark datasets for SBD. Moreover, there is still one fully automatic method that managed to make its way to the top 10 performing algorithms. The second dataset over which we performed our experiments is an extension of the dataset we have used in [40] and consists of five publicly available video sequences from Open Video Project [43]. The selected sequences were chosen to comprise various transition types such as wipe, dissolve, fade in/out etc., object/camera motions and to be in different video qualities. Additionally, we included some video sequences that are used in [30]. Strictly speaking, we believe that none of the videos used in [30] are suitable for testing the SBD performance due to their ambiguous content such as unclear and highly subjective shot boundaries and transitions, embedded subtitles, etc. For example, sequences shorter than 10 frames hardly qualify as shots since they are barely perceivable, yet they occur abundantly in the dataset. Transitions as long as 200 frames may even be considered as a separate shot (an overlaid shot for dissolve type transition for instance). An object passing in front of the camera is structurally identical to a wipe transition and it requires semantic comprehension to distinguish them. Embedded subtitles can be considered as a part of the visual information, but then the definition of a shot should also be well-defined in advance, i.e., what happens if subtitles stay intact but the background content changes – a new shot? Such occurrences and more arise abundantly in the dataset used in [30] which will inevitably bias the results both positively and/or negatively due to such ambiguous occurrences. Fig. 10 shows examples of such occurrences where on the top row the object in focus moves outside the camera scope within 10 frames and the blurred background gradually comes into focus (structurally this is not different than a dissolve transition). Similarly the second row shows a sequence where an object occludes the entire view as the camera moves to the right and the scene continues as the camera keeps moving and leaving the object outside the view (again, structurally the same as a wipe transition). A similar instance also occurs in the third example. Despite such deficiencies and inaptness, we decided to include three video sequences from the dataset used in [30] for the sake of completeness, namely News1, Documentary1 and TV Serial (Lost).

Table 1 summarizes the eight video sequences in the second set used in the experiments. Video#1 consists of 10 TV commercials each separated by ~50 blank frames. Video#2 and #3 are educational videos that contain various synthetic content (such as animations, frame borders, etc.). Video#4 and 5 are excerpts from industrial documentaries and together with Video #6, #7 and #8, they have relatively small frame size. Video#6 is a NASA documentary containing mostly dissolve type GT and also various shots with significantly short duration (only 25–30 frames). Video#1, #4, #5 and #6 are all from 1950s, thus have low video quality. Moreover, particularly Video#4 comprises challenging boundaries where shots with high motion are connected with slow wipe or dissolve type GT around 2 s. (~50–60
frames). Video #8 is particularly challenging due to style that the series is shot where very close facial shots dominate the video. Such a technique results in significant content change even under the slightest object movements. Moreover, significantly short shots (as short as 10 frames) and high motion content makes this video further challenging. Fig. 12 can be referred in order to grasp the gist of the contents of the videos.

As mentioned in Section 2.2, we used SURF for both feature detection and description due to its consistency.

Please cite this article as: M. Birinci, S. Kiranyaz, A perceptual scheme for fully automatic video shot boundary detection, Signal Processing-Image Communication (2014), http://dx.doi.org/10.1016/j.image.2013.12.003
Fig. 11. Performance vs. Computation Time analysis for the competing methods. The method proposed in this paper, in [30] and the only non-machine learning algorithm in TRECVID 2005 (CLIPS-IMAG) are labeled whereas the unlabeled data points are the remaining 9 of the top 10 performing algorithms in TRECVID 2005. The dashed horizontal line represents the speed equivalent to real-time operation.

Fig. 12. Excerpts from the SBD results of the proposed method from the second dataset. Video#1 (top row)-Video#8 (bottom row).

Please cite this article as: M. Birinci, S. Kiranyaz, A perceptual scheme for fully automatic video shot boundary detection, Signal Processing-Image Communication (2014), http://dx.doi.org/10.1016/j.image.2013.12.003
with human visual saliency and ease of computation [24]. Videos are sampled with 0.5 s. period (i.e. half of the frame rate of the video) in the overview phase, inferring from the definition of a video shot in Section 1 that any shorter duration will be impractical if not imperceptible. Moreover, as opposed to the initial conception, a more sparse sampling (larger N) does not yield smaller computation times since a larger N means more frames to process during zoom-in. Even though such a condition is heavily dependent on the video content (i.e. number of shots in the video), we observed insignificant variations in computation times for those N settings for 1sec. and 0.5 s. sampling. Also T_{ov}=0.5 and T_{2}=0.5 are used for all the experiments. The ground truths are extracted manually for all videos in the second dataset and precision-recall (P–R) values are calculated as performance measures. In order to provide a complete comparison in each dataset, performance measures used by competing methods have also been calculated: F1-score for the first dataset [16] and Q-value for the second dataset [45]. The experiments are carried out on a hardware with 4.00GB RAM and 2.20GHz Core2Duo CPU. The software relies on OpenCV libraries [44] for loading videos, querying frames and extracting/matching keypoints.

In addition to performance evaluation, computational time analysis is also provided in order to demonstrate that such performance is achieved with tremendous computational efficiency. In order to exhibit the improvement achieved by the employed search scheme over frame-by-frame methods in the second dataset, we followed the same search scheme utilized in [18,30] and compared every adjacent frame by extracting features from every frame in the video by the same descriptor, SURF. By doing so, we intend to demonstrate how much time it would take if a frame-by-frame search scheme is instead utilized as in [18,30]. In order to provide an accurate comparison against the competing methods, any approximate measure (such as [41]) is avoided. To our best knowledge, [30] achieved the best SBD performance using local image features. Despite the inappropriateness of the videos (Video #6, #7 and #8), the proposed approach achieved results on a par with [30]. Visual excerpts from the detection results are also provided in Fig. 12 where different types of GTs such as wipe, fade and dissolve are easily observed together with ACs. An immediate remark from Fig. 12 is that some of the ACs, particularly in Video#2 and #3, appear like dissolve type GTs. This is due to the fact that a single transition frame exists that is imperceptible by the human eye, yet detected by the proposed method.

Despite the clear advantage that the machine learning algorithms have which we discussed in Section 1, our method still managed to achieve a performance on par with all the top 10 performing algorithms on the first dataset. Excluding the machine learning approaches for the obvious reasons, our algorithm ranked the second among all algorithms involved in TRECVid 2005 based on F1-measure. The best performance came from the CLIPS-IMAG laboratory which does not use any machine learning algorithm, yet still uses specifically selected algorithms for TRECVid dataset. To be exact their system is composed of a cut detector, a flash detector (vastly present in the dataset) and a dissolve detector (78% of the GT’s in TRECVid dataset are of dissolve type). However, despite its significant performance, the algorithm runs considerably slow. Overall performance and execution time comparisons in the whole TRECVid 2005 dataset are given in Table 2 where time measures are given relative to real-time. The results in this dataset are a clear demonstration of the huge efficiency gain that the proposed SBD scheme provides without sacrificing high performance. In fact, the proposed method runs even faster than several machine learning algorithms (ranks 5th among all) despite the fact that the time for training the whole system is excluded from those algorithms’ execution time.

Table 3 summarizes the results of our experiments on SBD performance on the second dataset. The results indicate that on the average 96% of the shot boundaries can be detected by the proposed SBD technique in a generic way. This is most likely in the close vicinity of the upper recall limit that can be achieved without any training, learning or manual tuning involved. Considering that the experimental set contains a wide selection of GT types, video qualities, frame sizes and shot durations, it can easily be inferred from the results that the proposed algorithm is capable of detecting any type of GT and AC with such an elegant recall rate.

It should be noted that the main goal of the proposed algorithm is not only to achieve such an elegant performance, but also to achieve it under low computational costs. The computational times for the second dataset presented in Table 4 demonstrate that the proposed method is significantly superior in terms of computational efficiency. On the average around 87% improvement is achieved in terms of computation complexity compared to the methods [18,30] that employ frame-by-frame analysis. In other words, the proposed scheme enables around 7 times faster processing compared to any frame-by-frame processing scheme. Even though [30] has a comparable
detection performance, it is obvious that their frame-by-frame analysis leads to an impractical computational complexity for a real-time SBD operation. Moreover, note that the computation time for the proposed approach is obtained by fully employing the "Overview, zoom-in and filter" procedure (including the spatial analysis of the keypoints), whereas times for the frame-by-frame analysis approach includes only the feature detection, extraction and matching. It should be noted that, particularly [30] performs significant number of additional keypoint matchings and an intensive and computationally complex analysis on the number of matched keypoints which are excluded from the computational times reported in Table 4. It is possible to decrease the computation times given in Table 4 by using simpler and faster feature detectors/extractors (as in [30] via CCH), yet that possibility exists for any approach utilizing local image features bearing in mind that the proposed SBD scheme is independent of the utilized image feature. In other words, thanks to the efficiently utilized top-down search scheme, the computational supremacy of the proposed approach over any frame-by-frame processing algorithm will still prevail.

In short, the main advantage of the proposed algorithm is the ability to find out the exact locations of shot boundaries with a significantly low computational complexity (see Table 4). As discussed in Section 2.1 the outcome of the overview phase is the total number of shot boundaries and their imprecise locations (with a maximum deviation of \(N - 1\) frames). Note that this information alone can be useful and even sufficient for various applications, e.g., consider the case where a storyboard is to be extracted from a video where each shot is represented by a single video frame. Whereas the selection of representative frames (i.e. keyframes) among all shot frames is another research topic, the proposed overview scheme provides an immediate and fast solution to the problem without requiring any further implementation and computational cost. Another crucial advantage of the proposed method is that it allows the SBD results to be presented to the user in a progressive manner and furthermore allows user interactions with the ongoing process; i.e. the initial results (outcome of the overview phase) can immediately be presented to the user while the system can then continue to the zoom in phase if the need arises or alternatively, it can be stopped by the user if the results found so far are satisfactory. By doing so, not only excessive idle intervals are avoided, but also the possibility to interact with the system is granted to the user. Consider another use case where the user aims to extract only certain shots from the video. The overview phase initially provides representative frames from each shot in the video (those are the sampled frames mentioned in Section 2.1). This way, the user can directly access the shots of interest and the proposed method will then only zoom in to those shots' boundaries, thus avoiding redundant processing.

Performance vs. Computation Time analysis is also provided in Fig. 11 comparing the proposed approach with the top 10 performing algorithms in TRECVid 2005 dataset. For illustrative purposes, we have also added the algorithm in [30] despite the fact that there is no evaluation data on TRECVid dataset for that algorithm. Thus, we have
used the performance score they have reported in [30] and the computation time we have simulated and reported in Table 4. It is clear that despite the high performance score reported in [30], the computational efficiency is a huge handicap. The figure also demonstrates the on-par performance and computation time achieved by the proposed algorithm despite the aforementioned controversial objectivity of the machine learning algorithms used in TRECVID 2005.

Despite the fact that remarkable results are achieved in terms of accuracy, localization and computational complexity, there are rare cases where the proposed method failed to detect shot boundaries. One example of such occurrences is from Video#5 and shown in Fig. 13, where both shots are from the same scene and have the same camera angle. Moreover, considerably dark content of the shots weakens the discriminative power of the features, yielding a misjudgment that both frames belong to the same shot. Note that although these frames are from the same scene, a shot-cut occurred in between.

Another case, which is shown in Fig. 14, arises mainly from the uncertainty about the definition of a shot. The frames are from Video#2 and all from the same shot, where the color regions and text appears on top of the map gradually and leave the map vaguely visible. Such a change is regarded as a GT by the proposed algorithm. Yet, since those changes cannot be regarded as object or camera motion, it is hard to classify such artificial content changes as shot boundaries or not, even by a human observer.

4. Conclusions

A novel modus operandi for shot boundary detection is proposed where Gestalt laws of visual perception are taken as a model for both recognizing shot changes and seeking the location of the boundaries. In order to locate shot boundaries accurately and quickly, an efficient search scheme is proposed based on the “Information Seeking Mantra”. The proposed method provides an outstanding improvement in terms of computational complexity while achieving an elegant performance. Yet, the key contribution of the paper is in demonstrating how a proper understanding of human perception can lead a simple and effective solution for content analysis, and avoiding any over-engineering of the problem under the guidance of human psychology and human-computer interaction. Furthermore, the proposed method allows a user interaction to direct the SBD process to user’s “Region of Interest” or to stop it once satisfactory results are obtained. Considering that SBD is a prominent enabler in video content analysis, such interaction might be of valuable importance to certain applications minimizing user’s idle time and further lowering the computational cost significantly.

References


Please cite this article as: M. Birinci, S. Kiranyaz, A perceptual scheme for fully automatic video shot boundary detection, Signal Processing-Image Communication (2014), http://dx.doi.org/10.1016/j.image.2013.12.003