

Compressed Domain H.264/AVC Shot Detection

Hugo Santos Varandas

Instituto Superior Técnico, Technical University of Lisbon
Lisbon, Portugal

Abstract— Due to the advances in media coding and the increased availability on computer and network resources, the usage of digital video is widespread to the general public. This gives rise to new applications based on digital video, such as digital libraries and video-on-demand, which use large collections of video. Moreover, the increasing importance of user generated content made digital video more familiar to the general public leading to an exponential increase in video content creation. This originates the need for providing applications to efficiently browse and consume large amounts of video data, like content-based video retrieval and summarization applications. A fundamental step in these applications is to perform the temporal segmentation of the video into its elementary units. Since the unit most commonly used in this context is the shot, there is a growing need for shot transition detection algorithms. As digital video is usually compressed, shot detection algorithms benefit from operating directly in the compressed bitstream domain, without having to decompress the video. The video coding standard emerging in a large range of application domains is the H.264/AVC standard [1] which provides a major compression efficiency improvement at the cost of a significant increase in encoding and decoding complexity. The increased usage of compressed content further increases the need for efficient compressed domain shot transition detection solutions. Motivated by the aforementioned facts, the main objective of this Thesis is the design, implementation and evaluation of a shot transition detection algorithm operating in the H.264/AVC compressed domain both for hard and gradual transitions. In this report, the motivations, the state-of-the-art, the adopted architecture and the implemented algorithms will be presented; finally, a detailed performance analysis will be carried considering various alternative algorithms.

Keywords - Shot transition detection; H.264/AVC; Hard and Gradual Transitions; Hierarchical Detection; Suspect GOP; Prediction Modes.

I. INTRODUCTION

The extensive usage of digital video material gives rise to the need of improving the accessibility to video content by the users. A fundamental and initial step of the applications which provide this is, naturally, to structure the videos into shorter elementary units, i.e., to perform a temporal segmentation of the video. Among the possible types of elementary units, there is the shot which has been considered an appropriate elementary unit for this kind of applications and has been used by a great majority of them; a *shot* consists on a series of interrelated consecutive pictures taken contiguously by a single camera and representing a continuous action in time and space. Due to the importance of shot transition detection in this application context, shot transition detection tools have been an

extensively researched and reported in the relevant literature [2],[3],[4],[5].

However, digital video content is nowadays made available in a compressed format to reduce its storage and transmission requirements. The state-of-the-art on video compression is the H.264/Advanced Video Coding (AVC) standard [6] and, therefore, the state-of-the-art shot transition detection compressed domain systems are those which operate with H.264/AVC compressed videos.

There are many types of shot transitions in video content, notably depending on the content creator creativity. In this document, video shot transitions will be defined by the three parameters:

- *Pre-frame* – The last frame before the shot transition.
- *Post-frame* – The next frame after the shot transition.
- *Type* – The type of the shot transition.

Although there are several types of video shot transitions currently used in film editing to connect successive shots, they are usually grouped under two main classes: i) Abrupt transitions – when one frame belongs to the disappearing shot and the next to the appearing shot; and ii) Gradual transitions – where cinematic effects are added to combine the two shots gradually replacing one shot by another. This last kind of transitions is very customizable, according to spatial, temporal and chromatic characteristics, which makes them difficult to model and, therefore, to detect.

This paper is structured as follows: Section I – Introduction; Section II – Overview on the H.264/AVC; Section III – Short State of the art review; Section IV – Developed system architecture; Section V – Implemented Algorithms; Section VI – Performance Evaluation; and finally Section VII – Conclusions.

II. SHORT OVERVIEW ON THE H.264/AVC VIDEO CODING STANDARD

The video coding layer of the H.264/AVC standard splits the luminance and chrominance samples of each frame into blocks, the so-called macroblocks. To efficiently encode each macroblock, a prediction is made for its samples. To compute this prediction, the macroblock can be split into smaller blocks which are called prediction blocks. The encoder generates the bit stream containing the required information so that the decoder can compute the same prediction and the so-called prediction error, which is the difference between the actual original samples and the prediction. There are two major encoding prediction modes defined in the H.264/AVC standard:

- 1) *Intra Mode* – The prediction can be only based on samples from the current frame. Macroblocks can be encoded using different luminance prediction block sizes: *Intra4x4*;

Intra8x8 and *Intra16x16*. Each prediction block can be calculated using one prediction mode. For chrominance samples, the macroblock is not divided and a prediction is made for all the samples in the macroblock. This prediction is made in the same fashion as *Intra16x16*, since chrominance data is usually smooth over large areas.

2) *Inter Mode* – The prediction calculated for each prediction block is based on samples taken from, at most, two previously decoded frames which can, in visualization order, precede (forward prediction) or succeed (backward prediction) the current frame. For this purpose, two lists of reference frames are maintained: i) *list0* which is usually used for forward prediction, and ii) *list1* which is usually used for backward prediction; these lists define the frames that can be used for reference in the prediction. The prediction may be based on blocks in a different spatial position and, therefore, at least one motion vector is needed to indicate the displacement of the reference block; however, the number of motion vectors may significantly grow for more complex prediction modes. As for the intra case, inter macroblocks can be partitioned in several ways.

This new standard introduces a great flexibility in picture order and reference usage. This allows the creation of arbitrary coding structures and makes it possible to organize pictures in the bit stream in multiple ways. Usually, this is used for the creation of hierarchical coding structures which improve the coding efficiency and offer multi-layered temporal scalability in a straightforward way [6]. These structures consist on multiple layers of pictures which result in a coarse-to-fine structure. A particular example of such structures is shown in Fig. 1; in these structures, pictures can only use as reference for motion compensation pictures from the same or lower layers and pictures from the lower layer can only use as reference previous pictures in display order.

For each macroblock, the encoder decides which type of prediction should be used to maximize the coding efficiency. For this, it computes the prediction error, which is quantized and transformed; after, it entropy codes the prediction error along with other information so that the decoder can recompute that prediction; the outcome of the entropy coder is the H.264/AVC coded bit stream.

III. SHOT TRANSITION DETECTION IN H.264/AVC BIT STREAMS

In the context of compressed domain shot transition detection, the processing relies on the encoding tools used by the encoder to compress the video content. To efficiently compress the video content, H.264/AVC defines many encoding tools which can be used for shot transitions detection, such as intra and inter prediction.

Although, as previously referred, shot transition detection has been extensively reported in the literature, few publications report on shot transition detection algorithms working on H.264/AVC compressed domain since this is a recent coding standard. In [5], from 2004, Lui et al. propose to detect both abrupt and gradual transitions using a two-step detection; first, the GOPs are classified into suspect/non-suspect of having a transition based on the comparison of inter prediction modes used in intra frames and, afterwards, the suspect GOPs are

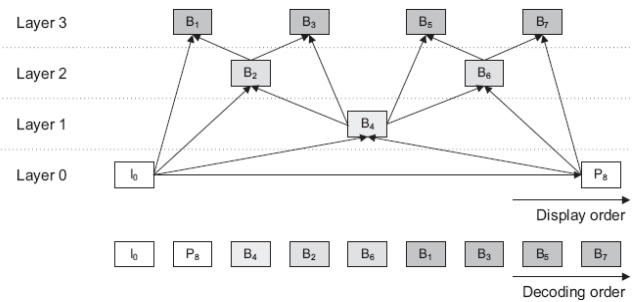


Fig. 1 - Hierarchical coding pattern with four temporal layers.

analyzed more thoroughly by inspecting the inter prediction modes and reference directions used in the inter frames of the GOP. In [7], from 2007, and [8], from 2008, Schöffmann et al. propose inspecting the partition sizes and the ratio of intra macroblocks to detect both abrupt and gradual transitions. In [4], from 2008, De Bruyne et al. describe a shot transition detection algorithm which effectively explores hierarchical bit streams in order to reduce the computational complexity by only analyzing a subset of the frames. This algorithm detects both abrupt and gradual transitions using spatial dissimilarities of intra prediction modes and temporal dependencies, more precisely, motion vectors and reference indexes.

IV. SYSTEM ARCHITECTURE

A. General Framework for Shot Transition Detection

Inspired by the formal study presented in [2], the shot transition detection can be composed of three steps:

- 1) *Feature Extraction or Frame Descriptions Generation*: The first step regards the extraction of features from each frame to obtain a compact content representation using an appropriate feature extraction method to map the image into a feature space. This yields a frame description for each analyzed frame.
- 2) *Similarity/Difference Scores Calculation*: The second step regards the determination of a continuity (similarity) or discontinuity (difference) signal between descriptions for different frames;
- 3) *Decision*: Given the continuity signal representing content variations, the final step regards the detection and classification of the transitions as cuts or as various types of gradual transitions.

B. Developed System Architecture

In the developed system), the two-phase hierarchical procedure used in [5] was adopted, see Fig. 2. As referred earlier, it is composed of the following two phases:

- 1) *Suspect GOP detection*: This is the part of the processing chain which is first executed; it aims at classifying each GOP in the video sequence as a suspect or a non-suspect GOP, depending on whether a transition is likely to occur in the GOP under analysis or not. This is performed by solely analyzing those frames which are the first from the corresponding GOP.
- 2) *Transition Detection*: In the second phase, the GOPs which were considered suspect of having transitions are analyzed more thoroughly, by considering all of its composing frames. In most of the shot detection systems, this second

phase is the only performed which is the equivalent, in this system, as considering all GOPs as suspect.

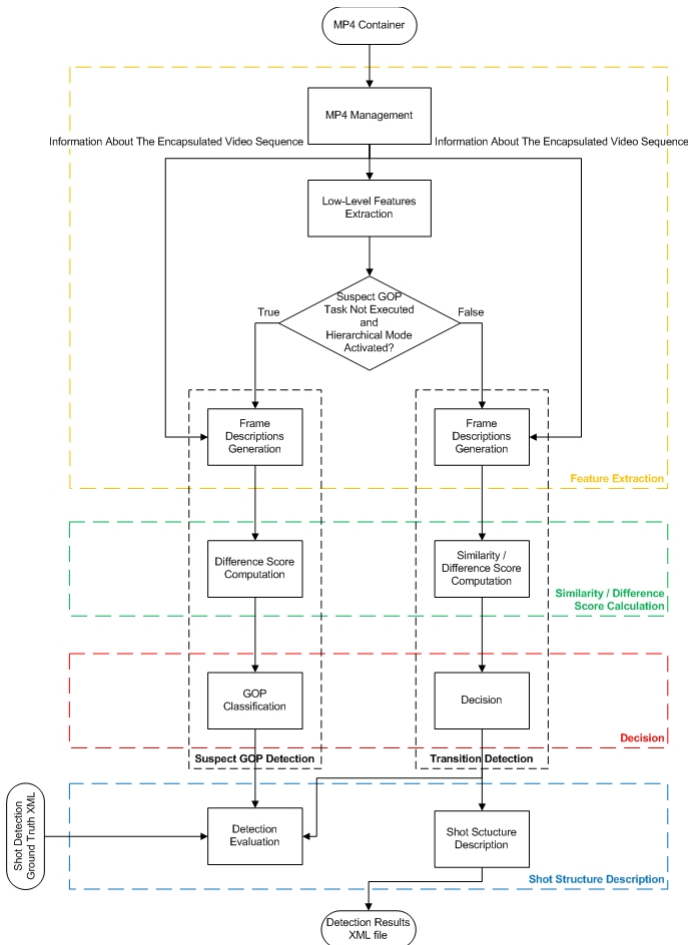


Fig. 2 – Overall system architecture.

Each of these phases is, basically, a shot transition detector consisting of the three modules referred to in Section IV.A. Besides these six modules (three for each detection phase), there are three more which provide the required elements to these detectors and, at the end, process the results. These are:

1) *Mp4 Management*: This is the first module of the system. It targets at providing the H.264/AVC bit stream which is encapsulated in a MP4 file. This is required since the H.264/AVC encoded videos are usually multiplexed along with other multimedia or metadata streams in a container file, like the MP4 file. Therefore an inspection of the Mp4 file is performed by this module to recognize and extract the H.264/AVC bit stream. This module was developed by the author of this work based on the GPAC – Project on Advanced Content.

2) *Low-level Features Extraction*: This is the second module, appearing immediately before the shot detection modules. It targets at providing the information about the encoding tools used to compress each frame. This module was developed by the author based on the H.264/AVC Reference Software distributed by the Joint Video Team (JVT) which designed the standard.

3) *Shot Structure Description*: This is the last module in the architecture, right after the detection modules. This module gathers the detection results in a XML file to support interoperability with other applications that may need the shot information from a video.

V. IMPLEMENTED ALGORITHMS FOR SHOT TRANSITION DETECTION

The developed shot transition detection is organized into a two-phase hierarchical procedure; so, first, the implemented algorithms for the suspect GOP detection will be described and, afterwards, a description of those implemented for the transition detection phase will be provided.

A. Suspect GOP detection

As referred earlier, this phase was originally proposed by Liu et al. in [5]. In this work, the authors propose using a 1-D histogram of 13 bins to describe each I frame at the beginning of the GOP, according to the used intra prediction modes. Each bin corresponds to the ratio between the number of blocks encoded using the corresponding intra prediction mode and the total number of blocks in a frame. Afterwards, a sum of absolute differences between the histograms of consecutive I frames is calculated and the obtained score is compared to a fixed threshold. If the score is above this threshold, the GOP is considered suspect and has to be analyzed more thoroughly; otherwise, a further analysis on the GOP is skipped. Inspired by this solution, the author of this work implemented several different algorithms in order to compare different approaches. Each algorithm is described by four characteristics:

1) *Features used*: The descriptions can be histograms representing the relative usage of the following features:

- *Luminance Prediction Modes (LUM)* – Introduced by the authors in [5].
- *Luminance and Chrominance Prediction Modes (LUMCOL)* – Based on the assumption that the luminance prediction modes reflect the encoded visual content, the author of this work proposes to add the 4 chrominance prediction modes to the histogram.
- *Luminance Partition Types (LUMPART)* – This is presented in [4] for a slightly different purpose. This yielded poorer performances when compared to the other two and, therefore, was not extensively tested.

2) *Feature Granularity*: The frame descriptions created for the features in each frame can be created using two granularity levels:

- *Frame (FRM)* – The descriptions generated in this mode do not include any type of spatial information. Using this granularity, the various types of occurrences are counted at frame level. This is the type of descriptions used in [5].
- *Window (WIN)* – The frame description is composed of block descriptions for each window of $N \times M$ macroblocks around each macroblock (N and M being odd numbers). This approach is presented in [4] and has the advantage of providing the generated descriptions with some spatial information.
- *Non-overlapping blocks (BLK)* – In this case, the frame is partitioned into non-overlapping blocks of size $N \times M$ macroblocks; if the height or width of the frame cannot be divided by the block dimension, the remaining

macroblocks at the edges are discarded. Comparing to the window approach, this is faster since the blocks are not overlapping.

3) *Difference Score*: To generate the difference score between two frames for the features defined in the previous section, a metric has to be chosen to compare the block descriptions from the corresponding blocks in those frames (frame descriptions taken at frame granularity are considered as block descriptions with only one block); afterwards, the scores obtained for the blocks are summed to generate the frame score difference. A difference score is generated for each GOP which is computed by comparing the first frame of the current GOP with the first frame of the next one. This score can be calculated using one of two approaches:

- *Sum of absolute differences (SAD)* – The sum of absolute differences was the metric originally proposed in [6]; in the current implementation, the only modification was the normalization of the metric leading to (1)..

$$D(h_1, h_2) = \frac{1}{N} \sum_{i=1}^{nb=number\ of\ bins} |h_1^i - h_2^i|, \quad N = \sum_{i=1}^{nb=number\ of\ bins} h_1^i + h_2^i \quad (1)$$

- *Variation of Pearson's homogeneity test (VPT)* – A variant of the Pearson's homogeneity test was implemented (2);

$$D(h_1, h_2) = \frac{1}{N} \sum_{i=1}^{nb=number\ of\ bins} \frac{|h_1^i - h_2^i|^2}{\max(h_1^i, h_2^i)}, \quad N = \sum_{i=1}^{nb=number\ of\ bins} h_1^i + h_2^i \quad (2)$$

4) *Threshold*: The difference scores created need to be evaluated in order to classify GOPs as suspect or non-suspect. Three types of threshold were implemented:

- *Fixed Threshold* - Each score is compared to a fixed threshold which is heuristically set before the analysis; this is the procedure used in [5].
- *Adaptive Threshold based on the Median* – An adaptive threshold (3) is computed for each frame taking into consideration the difference scores from surrounding GOPs which form a window of difference scores. There are some alternatives regarding the difference scores to consider in this window: it has N samples which may be centered on the current GOP or contain only values obtained from previous GOPs and the value of the current GOP may or not be discarded (depending on the chosen option).

$$T_a = a + b \times Median \quad (3)$$

- *Adaptive Threshold based on the Average* - This threshold is computed using (4), where a and b are heuristically set coefficients and μ (average) and σ (standard deviation) are calculated using the window of difference scores.

$$T_a = a + b \times \mu + c \times \sigma \quad (4)$$

These characteristics define the implementation to be used in each module related to the suspect GOP detection in the system. Feature type and granularity define the algorithm used in the generation of the frame descriptions; the difference score used defines the difference score calculation module and the threshold defines the classification module.

B. Transition Detection

This phase targets the detection of the frames in which a transition occurs; it analyses GOPs which have previously been considered as suspect. For this phase, four algorithms were implemented:

- *Algorithm 1*: This is the algorithm proposed by Schöffmann et al. in [7] and [8].
- *Algorithm 2*: A shot detection algorithm inspired by Algorithm 1 but with some modifications proposed by the author of this Thesis to improve its performance.
- *Algorithm 3*: A shot detection algorithm based on the system proposed in [4] with some modifications made by the author of this Thesis.
- *Algorithm 4*: A shot detection algorithm using hierarchical detection based on the system proposed in [4] with some modifications made by the author of this Thesis.

These four algorithms will be described in detail in the next sections. This description assumes the algorithm is using fixed GOP structures with $N = 15$ and $M = 3$, which will be used for the evaluation. Despite that fact, the algorithms can be easily extended to support other GOP structures.

1) Algorithm 1

This algorithm targets at detecting both abrupt and gradual transitions analyzing the partitions sizes and types used and the ratio of intra coded macroblocks.

a) Frame Description Generation

In this algorithm, only B and P frames are evaluated; each of these frames is characterized by two different descriptors:

- *Partition histogram (PH)* – This descriptor accounts for the inter partition sizes and types used in each frame. For the generation of this type of description, each frame is split into each 4x4 blocks and each block is grouped according to its prediction type (P, if forward prediction, B, if backwards, interpolated or direct prediction, or skipped prediction) and size of the corresponding prediction partition into 15 bins: P16x16, P16x8, P8x16, P8x8, P8x4, P4x8, P4x4, B16x16, B16x8, B8x16, B8x8, B8x4, B4x8, B4x4 and S16x16. Intra prediction partitions are not considered because the authors argue it would produce too many false positives since these prediction modes may be used also due to fast motion; instead, the usage of such prediction type is indirectly considered due to the effect its rises and falls produce in the usage of the considered partition types.
- *Intra block ratio (IBR)* – This descriptor contains the ratio of intra coded macroblocks in the current frame. As it is done for generating the PH descriptor, in this case the frame is also split into 4x4 blocks; afterwards, the ratio of those blocks belonging to intra prediction partitions is computed. As the intra blocks are used for new content, high usages of intra prediction modes may appear when a shot transition is taking place; however, this may also happen when encoding frames with fast motion.

b) Difference Score Computation

In this transition detection phase, this score accounts for the discontinuity in the visual content at the frame being analyzed; higher values mean a higher probability of a shot change taking

place and vice-versa. With this purpose, two scores are implemented for this algorithm, for each frame, notably:

- *Partition histogram difference (PHD)* – This metric evaluates the differences between frames by comparing the corresponding PH descriptions; the histograms of the current and previous frames are compared using a sum of absolute differences or a sum of non-absolute differences. According to the experiments realized by the authors who proposed this algorithm, the later performs better yielding less false positives when compared to the first [7], since there are some cases where partitioning changes, not due to real content change, but due to compression efficiency decisions of the encoder, e.g., if an encoder starts to use Skipped macroblocks instead of predicted macroblocks. However, this seems contradictory with the partition change detection since the changes using non-absolute differences will be only due to intra ratio change (rises and falls) and not due to partition changes.
- *Intra block ratio (IBR)* – This metric regards the direct usage of the IBR description for the current frame; for each frame, this is equal to the ratio of intra coded macroblocks in that frame.

c) Decision

In this last sub-module of the second phase, the similarity scores previously obtained are analyzed. As it was previously referred, high difference scores stand for a high degree of dissimilarity in the frames analyzed; therefore, by detecting those frames which correspond to high difference scores transitions may be detected. In the original algorithm [7], [8], Schöffmann et al. state that a frame should be considered as a candidate for an abrupt transition if its PHD is equal or above a predefined fixed threshold (T_{PHD}) or if its IBR is equal or above another fixed predefined threshold (T_{IBR}). These candidates are added to the respective PDH or IBR candidate set which will be provided to a post-processing procedure to transform this candidate set into a definitive transition set. This post-processing is a three step procedure including:

- *Gradual transition detection* – This step is meant to group frame candidates that seem to belong to gradual transitions. In this step, frames in the candidate set which are less than Δ frames apart from each other, as in (5), they are grouped; this is to tolerate “detection holes” which span over a maximum of Δ frames. If this group obeys to the size constraints as in (6), then it is considered a valid group, added to a gradual transition candidate set and the corresponding original abrupt candidates are removed from the set; otherwise, the group is discarded and the original abrupt candidates remain in the abrupt candidate set. There are two sets of these three parameters: one for the PHD and the other for the IBR.

$$G = \{c_1 \dots c_j\} \quad f_h(c_{i+1}) - f_h(c_i) - 1 \leq \Delta \quad (5)$$

$$\min GTsize \leq |G| - 1 \leq \max GTsize \quad (6)$$

- *Consecutive cut removal* – This rule (7) excludes from the candidate set abrupt candidates which are too close from each other assuming that shots have to be more than μ frames length. This comparison is checked starting in the

last cut candidate, which is compared to the previous cut candidate and excluded if it is too close, and performed until the first candidate is reached.

$$f_h(c_{i+1}) - f_h(c_i) \leq \mu \quad (7)$$

c_{i+1} excluded from candidate set

- *IBR/PHD combination* – This last step aims at combining the IBR and PHD approaches in order to create the detection set. In their experiments, Schöffmann et al. found that PHD alone works fine for cut detection; however, it lacks in gradual detection. On the other hand, IBR works better for gradual detection since it yields many false positives in cut detection. Therefore, after the previous post-processing steps, the PHD candidate cuts are added to the detected transition set and only gradual transitions are added to that set among the IBR candidates.

2) Algorithm 2

Algorithm 1 was only tested by its authors using videos encoded with the Baseline Profile; in fact, the description of the algorithm’s operation when using sequences encoded with other profiles provided in both [7] and [8] seems to lack functionality. Therefore, a second algorithm – Algorithm 2 - was designed by the author of this Thesis, still inspired by the ideas underpinning Algorithm 1 with the main purpose of improving its performance.

a) Frame Description Generation

Algorithm 2 uses the same type of descriptors as proposed for Algorithm 1. Comparing the descriptors in the two algorithms, the significant differences are in the PH descriptor; these modifications aim at enhancing the previous algorithm for B frames. With this purpose in mind, two major modifications in the definition of the descriptors are proposed. The major one regards partition classification where it is proposed to classify the partitions based on their size and prediction direction; since the objective of this algorithm is to use the partition approach adopted by Algorithm 1, the size still plays a major role in these descriptors. Therefore, the B prediction type is split into interpolated (I) and backward (B) prediction types with the skipped partitions being considered as forward partitions (either $P^{16 \times 16}$ or $P^{8 \times 8}$ depending in the partition size); this extends the histogram to 21 bins ($P^{16 \times 16}$, $P^{16 \times 8}$, $P^{8 \times 16}$, $P^{8 \times 8}$, $P^{8 \times 4}$, $P^{4 \times 8}$, $P^{4 \times 4}$, $B^{16 \times 16}$, $B^{16 \times 8}$, $B^{8 \times 16}$, $B^{8 \times 8}$, $B^{8 \times 4}$, $B^{4 \times 8}$, $B^{4 \times 4}$, $I^{16 \times 16}$, $I^{16 \times 8}$, $I^{8 \times 16}$, $I^{8 \times 8}$, $I^{8 \times 4}$, $I^{4 \times 8}$ and $I^{4 \times 4}$). In this way, the prediction direction is meant to be provided with more importance than it had in the original algorithm which is only based on the prediction types.

b) Similarity Score Calculation

The algorithms used in this module were also changed regarding the solutions from Algorithm 1; two difference scores are proposed:

- *IBR* – The same as in the original algorithm with no modifications;
- *PHD* – In this score, some modifications are proposed to enhance its operation. They target the better functioning of the original algorithm when B frames are involved since,

as previously outlined, the original algorithm does not cope well with B frames. Besides the slight change which the new extended descriptions would obviously impose, the assumption that frames should be compared equally disregarding their type or relative position does not seem accurate. Instead, before computing the PHD scores outlined for the previous algorithm, the frame types and relative positions are considered as follows in order to make those frames comparable:

- *B Frame vs. P Frame* – When the previous frame is a B frame and the current is a P frame, the B frame descriptions are modified for the purpose of this comparison by considering all interpolated and backwards predicted partitions as forward prediction. This is done so there are less false positives; in fact, if a B frame followed by a P frame uses mainly interpolated or backwards prediction, a shot transition should not be detected due to the decrease in the usage of those prediction directions.
- *P Frame vs. B Frame* – When the previous frame is a P frame and the current is a B frame, the B frame descriptions are changed by summing the values which correspond to the interpolated predicted bins with the corresponding bin in the forward prediction bin, for the same reason as in the previous case, and by considering backwards predicted blocks as intra blocks, since this is a what was expected to happen if there was a P frame in that place.
- *I Frame vs. B Frame* – Contrary to what happens in Baseline profile, using the Main profile a shot may be detected only considering the prediction direction of the B frames that follows an I frame. For that matter, a score in this comparison will be calculated considering the macroblocks in the I frame as P macroblocks and considering the B frame as in the last comparison (P frame vs B frame).
- *Same Type (P or B)* – When comparing frames of the same type, no change in the descriptions is needed.

In these scores, the regular intra frame processing is also performed in a similar fashion as in the original algorithm (Algorithm 1).

c) *Decision*

The algorithm used to identify the transitions based on the difference scores is similar to that in Algorithm 1. As in the previous algorithm, two candidate sets are created using the same thresholding procedure (based on T_{PHD} and T_{IBR}).

Afterwards, a similar post-processing is employed to transform the candidate sets into a transition set:

- *Gradual transition detection* – This step is meant to group IBR frame candidates that seem to belong to gradual transitions and is equal to that presented for the original algorithm (5) and (6).
- *Consecutive cut removal* – This excludes from the candidate set abrupt candidates which are too close from each other and is equal to that presented in (7).
- *Consecutive gradual transition joining* – This aims at joining gradual transitions which are overlapped or too close from each other, in which case would yield a very short shot between the two.

- *Cut/Gradual transition set combination* – Cuts from PHD candidate set and gradual transitions from the IBR candidate set are added to the transition set.

3) *Algorithm 3*

This third algorithm was defined mainly to compare the partition approach, outlined in the previous algorithms, with a gap-in-prediction chain approach which was partially adopted from [4]. This algorithm compares successive frames to detect both gradual and abrupt transitions. This algorithm can be divided in three steps:

- *Abrupt transition detection relying on temporal dependences (Inter step)* – This uses information from macroblocks belonging to inter frames and can be compared to the previous abrupt detection approaches.
- *Abrupt transition detection relying on spatial information (Intra step)* – This uses information from both inter and intra coded frames and is meant to complement to the Inter procedure.
- *Gradual transition detection (Grad step)* – This is meant to detect gradual transitions.

a) *Frame Description Generation*

Two frame descriptors were adopted and implemented from [4] for this Algorithm 3, notably:

- *Prediction direction* – This is used to describe the temporal dependencies of the frame under analysis; with this purpose, each frame is partitioned into 8×8 blocks and each is classified according to the prediction direction used: intra, forward, backwards or interpolated. This gives rise to a 4 bin histogram which is normalized by diving each bin by the number of 8×8 blocks which form the frame. This is related with the Inter and Grad steps above.
- *Intra prediction map* – This is used to describe the spatial characteristics of a certain frame. It is constructed for two frames in a GOP: the first and the last, and contains the intra prediction encoding information (as it is done in the suspect GOP detection phase); each prediction map starts being constructed at the beginning of the GOP (I frame) and advances through its P frames until the frame for which the map is being constructed is reached; meanwhile, every time an intra coded macroblock is found in those frames, the corresponding macroblock prediction information in the prediction map is updated. After updating this prediction map with the current frame, intra frame descriptors are generated for that prediction map using the same algorithms presented for suspect GOP detection. This is related with the Intra step above.

In the original algorithm in [4], another descriptor is proposed: the motion intensity for the foreground and background areas of the picture which is used in the gradual transition detection. However, motion extraction from the H.264/AVC bit stream is not a straightforward procedure since the motion vectors are not directly available from the bit stream; instead, only the differential motion vectors are available and can be parsed from the bit stream. To compute the motion vectors, a motion vector prediction has to be inferred from neighbor partitions, which is only done in late stages of the decoding process.

b) Similarity Scores Computation

In this algorithm, four scores are calculated with the purpose to express the continuity and discontinuity between frames:

- *Sum of intra and forward predicted block ratios for previous frame (s_1)* - This expresses continuity in the previous frame related to the video content before it and it is calculated for every inter frame. This is related with the Inter step.
- *Sum of intra and backward predicted block ratios for current frame (s_2)* - This expresses discontinuity in the video content between the previous and current frame and it is calculated for every inter frame. This is related with the Inter step.
- *Intra block ratio (IBR)* - This is the IBR for the current frame; this is only calculated in P frames and is related with the Grad step.
- *Intra frame difference (D_{intra})* - Unlike the previous scores, this is not computed for all frames; instead, it is used to calculate differences between an intra prediction map belonging to a P frame, that at the end of each GOP, and an intra frame, that at the beginning of the succeeding GOP; to calculate such score, the algorithms described for suspect GOP detection are used. This is related with the Intra step.

c) Decision

The decision process for transition detection in this algorithm is based on the similarity scores defined earlier as in [4]:

- *Abrupt Transitions* - If both s_1 and s_2 are above a predefined fixed threshold (T_{Inter}), then a gap in the prediction chain is detected. The outcome of this comparison may be:
 - If the current frame is neither an I nor an IDR frame, a transition is detected.
 - If the current frame is an IDR or an I frame, the D_{intra} score must be considered; this score is computed between the current frame and the intra prediction map of the previous frame. An adaptive threshold (T_{intra}) is also computed, similar to the average-based threshold for suspect GOP detection; a window of N previous intra frame difference values is considered to calculate the terms μ and σ in (4); the rest of the terms are defined heuristically. If the obtained score is above the computed threshold, an abrupt transition is detected.
- *Gradual Transitions* - This is focused on the analysis of the IBR scores of P frames. In this case, another adaptive threshold is computed based on the expression (4), by analyzing a window of N previous IBR scores in P frames. If the current IBR score is above the threshold computed for the corresponding frame, it is considered as a candidate for a gradual transition; afterwards, a post-processing stage as for Algorithm 2 is executed.

At the end, the detected transitions, both the gradual and abrupt are added to the transition set.

4) Algorithm 4

This fourth algorithm was inspired by the hierarchical approach in [4]. It is meant to improve Algorithm 3 in two ways:

- A different method for detecting abrupt transitions comparing P frames;
- The introduction of hierarchy in the detection to avoid false positives. By observation of the results of the previous algorithms in the Main profile, it can be noted that B frames sometimes trigger abrupt transitions which do not occur and could be avoided by analyzing the P/I reference frames that surround the B frames. Therefore, a two-layers algorithm is suggested where one layer is composed by the non-reference B frames.

This algorithm is designed to detect both gradual and abrupt transitions.

a) Frame Description Generation

The same frame descriptors used in Algorithm 3 are used without any kind of modification.

b) Similarity Scores Computation

In this algorithm, the same four scores, as in Algorithm 3, are used to access continuity / discontinuity.

c) Decision

This is the module where the modifications introduced above take effect. As in the previous algorithm, the decision process for transition detection in this algorithm is based on the similarity scores defined earlier:

- *Abrupt Transitions* - To detect abrupt transitions, this algorithm starts at comparing base layer frames (I and P reference frames). For this purpose, scores s_2 are evaluated against one of two thresholds; two possibilities will be tested:
 - T_{inter} - A heuristically set threshold as used in Algorithm 3 for the Inter step;
 - T_{inter2} - An adaptive threshold, proposed by the author of this Thesis, which aims at detecting peaks of s_2 . This is composed by a fixed component (T_{interp}) and an adaptive one and is calculated in the following way: a) If the previous and next P frames are within 3 frames, i.e., $P_i - P_{i-1} \leq 3$ or $P_{i+1} - P_i \leq 3$, the threshold is equal to $T_{interp} + \text{average}(s_2(P_{i-1}), s_2(P_{i+1}))$; b) Else, if there is only one of such frames then $T_{interp} + s_2(P_i \text{ or } P_{i+1})$.
- If a positive is found while comparing s_2 in the base layer with the chosen threshold, the process analyses the s_1 and s_2 scores from the frames between the previous base layer frame and the current base layer frame, including this last to avoid some false positives for low T_{interp} thresholds, against the T_{inter} to detect transitions, as done in algorithm 3. This can detect the exact placement of the transition or exclude the possibility of a transition.
- Afterwards, whenever a positive is found:
 - If the current frame is neither an I nor an IDR frame, a transition is detected.
 - If the current frame is an IDR or an I frame, the D_{intra} score must be considered; this score is computed between the current frame and the intra prediction map of the previous frame. An adaptive threshold (T_{intra}) is also computed, similar the average threshold

in suspect GOP detection; a window of N previous intra frame difference values is considered to calculate the terms μ and σ in (4); the rest of the terms are defined heuristically. If the obtained score is above the computed threshold, an abrupt transition is detected.

- **Gradual Transitions** – To detect gradual transitions, the same process as in Algorithm 3 is used.

In the end, transitions detected, both gradual and abrupt, are added to the transition set.

VI. PERFORMANCE EVALUATION

A. Dataset

In this performance evaluation, the video collection from the TRECVID 2007 was adopted [9]. This collection consists of 17 MPEG-1 encoded videos, yielding a cumulative length of 6 hours. The videos have a luminance resolution of 288x352 pixels, a frame rate of 25 fps and are encoded at 1157 kbps. This video set consists of 2,463 transitions; 2,236 cuts (90.8%); 134 dissolves (5.4%); 2 fade-out/-in (<0.1%); 91 other special effects (3.7%). Manually-annotated ground truth provided for TRECVID was also used without any modification.

For the purpose of the work presented in this Thesis (see the graphical user interface in Fig. 3), the test videos had to be recompressed using the H.264/AVC standard. Two datasets were created:

- **Baseline 512kbs**
 - GOP Size = 15 frames;
 - Average bit rate = 512 kbps;
- **Main 512kbs**
 - GOP Size = 15 frames;
 - Average bit rate = 512 kbps;
 - Number of B frames (between I/P frames) = 2;
 - B Frame Mode = Auto (can use both temporal and spatial direct)

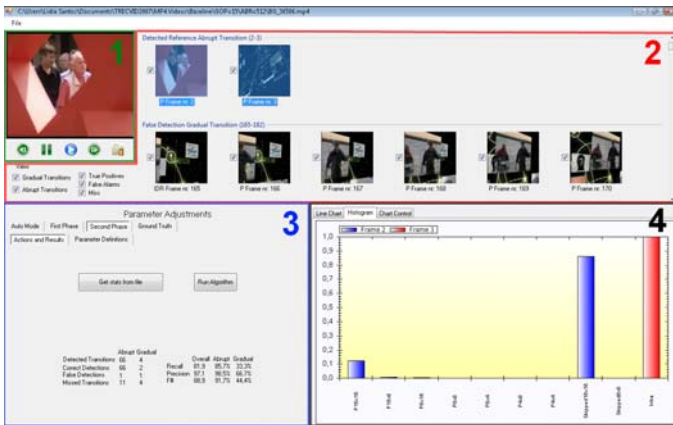


Fig. 3 – Graphical User Interface overview.

B. Performance Evaluation Procedures

Most of the shot detection algorithms in the literature evaluate their performance based on two main metrics: Precision and Recall; Recall denotes the proportion of the number of detected shots to the number of existing shots and Precision quantifies how many of the detected shots are correct shots. They are usually separately computed separately for abrupt and gradual transitions, since the detection difficulty

and, usually, the algorithm used for the detection of each kind of transition are rather different.

The performance evaluation procedure used was adopted from TRECVID [9]. The only difference between the evaluation procedure which was used and the original procedure from TRECVID is that the last expands the ground truth for each abrupt transition five frames in each direction whereas the used does not. A 1-1 matching procedure, which requires one frame overlap between the ground truth and the detected transitions, is carried out to classify each detected transition as a true or a false positive.

To evaluate the performance of the shot detection algorithm first phase, a novel procedure was designed based on the procedure presented in the previous section for the second phase. This novel procedure is proposed since no adequate performance evaluation procedure could be found in the relevant literature.

C. Performance Results and Analysis

1) First Phase: Suspect GOP Detection Performance

To evaluate the suspect GOP detection phase, several parameters were varied to cover the most relevant solutions. The dataset used for these tests was encoded using the Baseline 512kbs profile (in the tests using the Main profile, the algorithm seemed to achieve similar performances).

Each combination of feature type and granularity, difference score and threshold type, the evaluation results were obtained by performing the detection varying the threshold parameters (in the average and median threshold the parameter used for this was the b , while the others were set to 0). This yielded several recall/precision points which were used to construct precision/recall charts, where the results obtained with the several approaches can be easily compared. In Fig. 4, the best combinations of parameters for each threshold type are compared. From the conducted tests, it is possible to conclude that the best overall detection performance is achieved using luminance and chrominance prediction modes, sum of absolute differences and the median or average thresholds, which yield similar results.

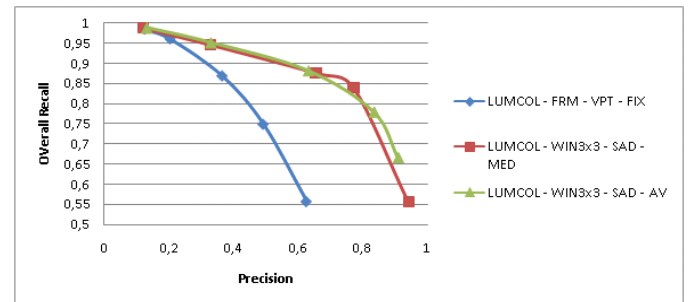


Fig. 4 – Recall/Precision for suspect GOP detection using the best combinations of parameters for each threshold type.

2) Second Phase: Transition Detection Performance

In this section, the results of the tests performed on the second phase algorithms will be presented. These tests were carried out by ignoring the first phase, i.e., by considering all GOPs in the videos as suspects. The tests will be presented organized first by the dataset profile used and then by transition type.

a) Baseline Profile

The performance results achieved by all implemented algorithms will be presented next; first, for the abrupt transition detection and, afterwards, for the gradual transition detection.

ABRUPT TRANSITION DETECTION

In the context of abrupt transition detection, the procedures for the algorithms implemented can be grouped between those which process only P frames (PHD in Algorithms 1 and 2 and Inter in Algorithms 3 and 4) and those which also use IDR frames (Intra in Algorithm 3 and 4). Fig. 5 shows the results obtained for abrupt transition detection by the algorithms which only use P frames.

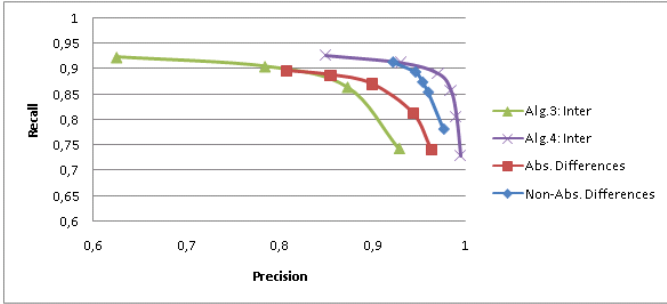


Fig. 5 - Recall/Precision for abrupt transition detection for the algorithms relying on temporal dependencies in Baseline profile.

From the analysis of Fig. 5, it is possible to conclude that:

- The Recall has a maximum at about 92%. This is due to the IDR frames, as transitions between P and IDR cannot be detected by any of these algorithms.
- PHD absolute differences perform better than non-absolute differences, as it was mentioned by the authors.
- Inter seems to perform better than PHD, yielding better precision for similar recall. In fact, the partition approach does not yield any improvement over a simple evaluation of prediction directions.
- The inter procedure from Algorithm 4 (using T_{inter2}) performs much better than that from Algorithm 3. This happens since algorithm 4 (T_{inter2}) detects peaks of intra usage while Algorithm 3 only detects high intra usages, which triggers many false alarms.

To detect those transitions between P and IDR frames, the Intra procedure is used in both Algorithms 3 and 4. To test this intra procedure, three feature types were used: LUM, LUMCOL and LUMPART. As for the threshold, a fixed one was used by limiting the T_{min} and T_{max} in the adaptive threshold. This was done because the threshold proposed by the original authors, despite being more complex, did not perform significantly better.

From the analysis of Fig. 6, one may conclude that the LUMCOL based features achieve a better performance than the others, notably than the LUMPART feature proposed by the authors. Note that the recall is very low since all missed transitions are considered, not just those between P and I frames, which this algorithm aims to detect.

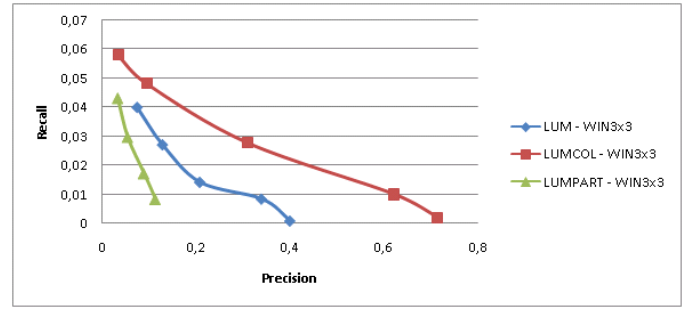


Fig. 6 - Recall/Precision for abrupt transition detection for the spatial differences (Intra procedure) using a fixed threshold in Baseline profile.

GRADUAL TRANSITION DETECTION

In the tests carried out, all algorithms used IBR to detect gradual transitions; the only difference is the addition of a post-processing step to join gradual transition detections which are close to each other, in Algorithms 2, 3 and 4 in comparison to Algorithm 1. Despite a different threshold was proposed for Algorithms 3 and 4, it was not used; this adaptive threshold did not improve the performance over a fixed threshold. Therefore, for all algorithms the same procedure as used.

The IBR detector has four parameters that need to be set. For Algorithm 1, only $maxGTsize$ was made constant; this was set to a very high value, since over the dataset the lengths of the gradual transitions vary considerably and the false positives rejected by setting this parameter were not significant. The T_{IBR} , $minGTsize$ and Δ were the test variables used to generate the results. As for Algorithm 2, $minGTsize$ and Δ were also made constant ($minGTsize=5$ and $\Delta=2$). Fig. 7 shows the performance of IBR at detecting gradual transitions for different parameters.

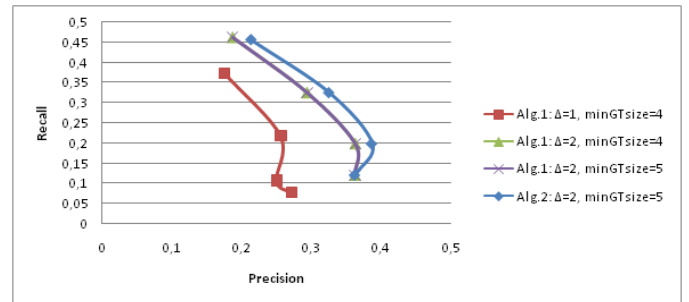


Fig. 7 - Precision/Recall for the gradual transition in Baseline profile.

From the analysis of Figure 7, it is possible to note that:

- In Algorithm 1, the first parameter configuration, which is that proposed by the authors perform worse than the others.
- Between the two algorithms, one may conclude that the concatenation of gradual transitions proposed improves the precision for gradual transition detection.

b) Main Profile

In this section, the performance results achieved by Algorithms 2, 3 and 4, while processing bit streams encoded in Main profile will be presented. The results for these algorithms will be presented next; first for the abrupt transition detection and, afterwards, for the gradual transition detection

| | First Phase | | | Overall System | | | | | |
|---------------|-------------|-----------|------------------|----------------|-----------|--------|-----------|---------|-----------|
| | | | | Overall | | Abrupt | | Gradual | |
| | Recall | Precision | Suspect GOPs (%) | Recall | Precision | Recall | Precision | Recall | Precision |
| b=0 | 100% | 100% | 100% | 85% | 84,7% | 90,5% | 91,1% | 25,2% | 22,6% |
| b=0,95 | 99,5% | 9,3% | 62,3% | 84,6% | 85,3% | 90,1% | 91,6% | 25,2% | 23,1% |
| b=1,1 | 95,2% | 32,6% | 17% | 81,9% | 89,3% | 87,4% | 93,3% | 21,4% | 30,6% |

Table 1 - Some performance results for the developed system.

ABRUPT TRANSITION DETECTION

In order to detect abrupt transitions in videos encoded in the Main profile, there were presented three algorithms relying of inter coded frames and one, which as in the baseline profile, tackles the problem of IDR frames.

In Fig. 8, the results for the various abrupt detection procedures relying on temporal dependencies are shown.

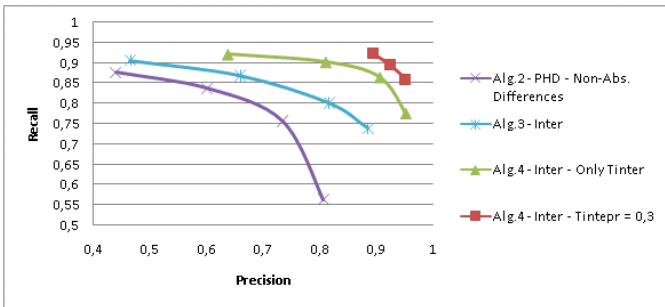


Fig. 8 - Precision/Recall for the abrupt transition detection relying on temporal dependencies in Main profile.

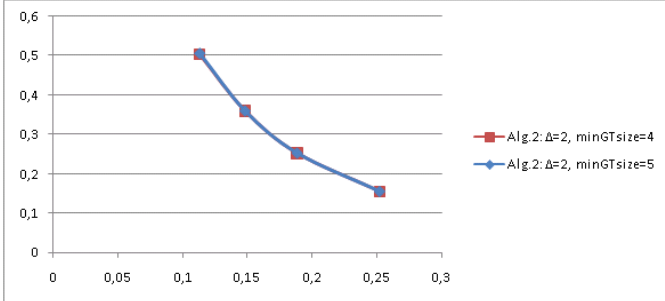


Fig. 9 - Precision/Recall for the gradual transition detection in Main profile.

From the analysis of Figure 8, one may conclude that:

- The PHD is that which performs the worse. In fact, it yields to much false positives even when compared to the Inter procedure in Algorithm 3.
- The hierarchical approach introduced in Algorithm 4 improved the detection results significantly when compared to Algorithm 3.
- The introduction of the peak detector threshold (T_{inter2}), to compare P frames from the base layer, improved the precision over usage of the T_{inter} threshold for that purpose, which only detects high usage of intra coded macroblocks in these frames.

As for the performance of the intra procedure, in the preliminary testing LUMCOL still yielded better results than the other features. Besides, this procedure seems to work better in this kind of sequences than it did in the Baseline profile.

GRADUAL TRANSITION DETECTION

For the detection of gradual transitions, the same procedure was used for all algorithms, for the same reasons as in the Baseline profile. Fig. 9 shows the performance of IBR at detecting gradual transitions for different parameters.

3) Overall System

After the analysis of the performance of first and second phases, in the following some results are presented about the two-phase overall system in Table 1. To obtain these results, the following parameters were set:

- *First Phase* – LUMCOL features; WIN3×3 granularity; SAD score and average-based score.
- *Second Phase* – Algorithm 4;
 - Inter procedure : $T_{inter} = 0,7$; $T_{interp} = 0,3$.
 - Intra procedure: WIN3×3; LUMCOL; $T_{intra} = 0,55$.
 - Grad procedure: $T_{grad} = 0,6$; $minGTsize = 5$; $\Delta = 2$.

VII. CONCLUSIONS

For the suspect GOP detection phase, the obtained results were below those expected and reported in the original algorithm [5]. Despite that fact, the introduction of this phase allowed several GOPs to be skipped from a detailed analysis in the second phase. Many modifications were proposed to the original algorithm which yielded improvements in the algorithm's performance.

For the transition detection phase, four algorithms were implemented. Many conclusions can be drawn from the tests carried out and the presented results: notably, inspecting inter partitions sizes does not yield better performance detection when compared to the simpler analysis of inter prediction direction. Also, the usage of hierarchical detection inside the GOP improves performance over analyzing successive frames. Finally, there are two main aspects which may need a more proper solution; first, gradual transitions are very difficult to detect based only on the ratio of intra prediction usage; second, the usage of IDR frames limit the prediction direction to be used and, although a solution is proposed in [4], it is still a problem needing a better solution, since it is the main problem limiting the performance of abrupt transition detection.

REFERENCES

- [1] "ISO/IEC 14496-10: Advanced Video Coding."
- [2] J. Yuan et al., "A formal study of shot boundary detection", IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, n° 2, pp. 168-186, Feb. 2007.
- [3] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?", IEEE Transactions on Circuits and Systems for Video Technology, , vol. 12, n° 2, pp. 90-105, Feb. 2002.
- [4] S. De Bruyne et al., "A compressed-domain approach for shot boundary detection on H.264/AVC bit streams", *Signal Processing: Image Communication*, vol. 23, n° 7, pp. 473-489, Aug. 2008.
- [5] Y. Liu et al., "A novel compressed domain shot segmentation algorithm on H.264/AVC", International Conference on Image Processing 2004, Singapore, 2004.
- [6] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF", *IEEE International Conference on Multimedia and Expo*, Toronto, Ontario, Canada: 2006
- [7] K. Schöffmann and L. Böszörményi, "Early Stage Shot Detection for H.264/AVC Bitstreams", *Technical Report*, Jul. 2007; [http:// www.itec.uni-klu.ac.at/~klschoef/papers/shotdetection.pdf](http://www.itec.uni-klu.ac.at/~klschoef/papers/shotdetection.pdf).
- [8] K. Schöffmann and L. Böszörményi, "Fast segmentation of H.264/AVC bitstreams for on-demand video summarization", *14th International Multimedia Modeling Conference*, Kyoto, Japan: 2008.
- [9] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID", *8th ACM International Workshop on Multimedia Information Retrieval*, CA, USA, 2006.