Classification of Indecent Videos by Low Complexity Repetitive Motion Detection

Tadilo Endeshaw
Karlstad University
Universitetsgatan 2
65188 Karlstad, Sweden
tadilo291@yahoo.com

Johan Garcia
Karlstad University
Universitetsgatan 2
65188 Karlstad, Sweden
johan.garcia@kau.se

Andreas Jakobsson *
Lund University
Sölvegatan 18
221 00 Lund, Sweden
aj@maths.lth.se

Abstract—This paper proposes a fast method for detection of indecent video content using repetitive motion analysis. Unlike skin detection, motion will provide invariant features irrespective of race and color. The video material to be evaluated is divided into short fixed-length sections. By filtering different combinations of B-frame motion vectors using adjacency in time and space, one dominant motion vector is constructed for each frame. The power spectral density estimate of this dominant motion vector is then computed using a periodogram with a Hamming window. The resulting power spectrum is then subjected to a Slepian selection window to restrict the spectrum to a limited frequency range typical of indecent movement, as empirically derived by us. A threshold detector is then applied to detect repetitive motion in video sections. However, there are instances where repetitive motion occurs in these shorter sections without the video as a whole being indecent. As a second step, an additional detector can be employed to determine if the sections over a longer period of time can be classified as containing indecent material. The proposed method is resource efficient and do not require the typical IDCT step of video decoding. Further, the computationally expensive spectral estimation calculations are done using only one value per frame. Evaluations performed using a restricted set of videos show promising results with high true positive probability (>85%) for a low false positive probability (<10%) for the repetitive motion detection.

I. INTRODUCTION

In many applications there is a need to be able to classify video material as containing indecent contents or not. One approach to detect indecent content in videos is to treat each frame as an image and to apply image based approaches such as [1], [3], [9] to the frames. This, however, requires rather large amounts of computations and does not utilize the temporal aspect of video. Many researchers use skin detection as a clue for filtering indecent content. Motion, however, will provide invariant features irrespective of race and color. An alternate approach using the temporal aspect to detect indecent content is based on periodicity analysis of audio as proposed in [5], which also briefly consider the use of motion information. In this work, we focus on the use of motion information to infer whether or not a video sequence contains repetitive motion, and use this as an indicator for indecent material. The use of repetitive or periodic motion for video analysis has been used previously [2], [4], [8]. However, our focus is on indecent material where there is explicit sexual activity of a repetitive nature, and our solution will thus not necessarily cover all facets of indecent videos1. Further, we focus our effort on a solution that is fast, with the aid of being able to process video material considerably faster than real time. We have designed a solution that works in the compressed domain and which is based on spectral evaluation of the dominant motion in a video sequence. Our initial experimental evaluation shows promising results. The contents of this paper is structured as follows. In Section II the extraction and processing of motion vectors is discussed. Section III then covers the signal processing done to extract the spectral information. Then Section IV presents the initial evaluation and Section V provides some conclusions.

II. MOTION EXTRACTION

This work focuses on video material that is being distributed over a network or by some storage medium. To allow efficient distribution, such video is typically encoded with some efficient video encoder to reduce the number of bytes required for storage and transfer. A typical characteristic of efficient video coders is that they use motion estimation to increase the attainable compression ratios. This fact is used in the presented approach, to in effect recycle the motion estimation work performed by the video encoder for the purpose of spectral analysis of dominant motion. Consequently, the proposed techniques are less applicable for computer vision and other applications where the raw uncompressed video stream is processed.

A. Frame selection

In motion compensating video coding schemes such as MPEG-2 there are generally several frame types, that have varying amount of motion information in the form of motion vectors (MVs). For MPEG-2 there are three frame types: I-frames, that do not contain any motion vectors vectors; P-frames, which contain only forward predicted motion vectors; and B-frames, comprised of both forward and backward

1What is considered “indecent” will vary greatly between different cultural environments as well as individuals and it is thus hard to make an indisputable definition.
predicted vectors. A typical frame sequence in an MPEG-2 video is organized as IBPBBPBBPBBIBPBBPBBPBB. The sequence IBPBBPBBPBB is defined as group of pictures (GOP). In our continued discussion, we use the IBPBBPBBPBB frame sequence, but the methods are adaptable to any block-based motion compensating video compression scheme. As I-frames do not hold motion information, obviously they can not be used as motion sampling frames. The problem of using P-frames is that they are not uniformly spaced. Although the motion information in the P blocks could have been used, the discontinuity produced by the periodic presence of an I frame instead of a P frame would have caused additional complexity for the spectral estimation. Remaining are the B-frames that can contain both forward and backward predicted motion vectors. Consider a truncated GOP of IB1B2P frames. B1 is temporally more correlated with the I-frame than with the P-frame and B2 is more correlated with the P-frame than with the I-frame. It is generally believed that vectors estimated from a more correlated reference frame have more information than less correlated reference frames. After an examination of the motion vector characteristics in a number of video sequences, this belief was verified. Consequently, the focus was on the B-frames, and alternating between forward and backward predicted motion vectors as discussed later.

B. Motion vector filtering

When examining the motion vectors of video sequences, it is common to find motion vectors that do not represent true motion, but instead are a result of the encoder trying to minimize the amount of information needed to create the difference image for the macro-block. This is especially true for areas which have a low amount of texture, are poorly lit or are out of focus. This effect is clearly visible in Figure 1, which shows one frame in a video sequence where a black dot is being moved repeatedly across a uniform background. This video sequence was created with typical home-video equipment. Although the dominant motion is discernible from the motion vectors in the image there is a huge amount of non-motion related, noisy, motion vectors. When instead looking at a frame from a video sequence in a commercially produced movie without low texture, as shown in Figure 2, which depicts a frame where a dinosaur is moving, it is evident that there are much fewer noisy motion vectors in this case. Since we would like our method to work also for indecent movies produced by home-type equipment under non-ideal lighting conditions, it becomes apparent that filtering is necessary to cancel the effects of the noisy motion vectors. Motion vectors relating to actual motion are typically surrounded by motion vectors pointing in the same direction [6], which is also visible in Figures 1 and 2. This fact was used to implement a simple filtering scheme to discard the noisy motion vectors that do not signify actual movement. This filtering is done as follows:

1) The relative change between a vector belonging to one block and the vectors in at least one of the eight directly neighboring blocks should be zero.

2) The normalized angular difference between the current macro block’s motion vector and the motion vector of at least one of the nine neighboring blocks vector in the adjacent B-frame should be zero.

Two general criteria are thus used to filter outliers, which involve either the eight blocks in the immediate vicinity of the current frame (definiteness property) or nine blocks in the immediate vicinity of the adjacent frame (continuousness property). If these two conditions do not hold true the motion vector is treated as noisy, and set to zero. Experiments were performed to also allow some relaxation of the requirements of zero angular difference, but this did not have any significant impact on the end result. Mathematically, these two criteria can be formulated as follows:

First condition (definiteness)

\[ \prod_{i=1}^{8} (V_i(x, y) - V_i(x + \Delta x, y + \Delta y)) = 0 \]  (1)
Second condition (continuousness)

\[
\prod_{i=0}^{8} (\arctan V_t(x,y) - \arctan V_{(t+k)}(x+\Delta x, y+\Delta y)) = 0
\]

where

\[
\Delta x = \text{sign}(i\sin(\pi/4(i-1)))
\]

and

\[
\Delta y = \text{sign}(i\cos(\pi/4(i-1)))
\]

and where \(V_{(t+k)}(x,y) = V_{(t+k)}(x,y) - V_{(t+k)}(x,y)\) with \(V_{F}(x,y)\) denoting the Forward predicted motion vector of the current frame at macro block location \((x,y)\), and \(V_{B}(x,y)\) denoting Backward predicted motion vector of the current frame at macro block location \((x,y)\), respectively.

\[
sign(x) = \begin{cases} 
-1 & \text{if } x < 0 \\
0 & \text{if } x = 0 \\
1 & \text{if } x > 0 
\end{cases}
\]

and

\[
k = \begin{cases} 
1 & \text{For the next frame} \\
-1 & \text{For the previous frame} 
\end{cases}
\]

When either of these two preconditions fail, the vector field will not be used for the upcoming process.

C. Establishing a representative motion vector

To extract the motion from two consecutive B frames two approaches to collect data were used as a consequence of the temporal correlations. For the first of the two B-frames, the forward predicted motion vectors were collected, i.e., First B-frame, Forward predicted (FBF). For the second of the two B-frames, the backward predicted motion vectors were used, i.e., Second B-frame, Backward predicted (SBB). This created two sets of filtered motion vectors, one for each frame. However, in order to do the later spectral estimation, it is not really necessary to represent the motion in a frame at the detail level represented a large number of motion vectors. By instead creating a single representative motion vector for each frame, the amount of processing needed in the spectral estimation step is drastically reduced, which fits nicely with our aim of high-speed classification. Using the filtered motion vectors from the previous step, two simple approaches to creating a single representative motion vector for the entire frame were used:

1) **Weighted mean** : Sum up all filtered motion vectors and assign the representative motion vector as the weighted average of filtered motion vectors by the following equation:

\[
V_R(x,y) = \frac{1}{L_{non}} \sum_{x=1}^{N} \sum_{y=1}^{M} V(x,y)
\]

where \(L_{non}\) is the number of non-zero vectors in the filtered frame, \(N\) is the number of rows of the filtered frame, and \(M\) is the number of columns.

2) **Median** : Assigning the dominant movement by calculating the median of the filtered motion vectors.

The x and y direction representative motion vectors are calculated independently. While this simple way of calculating the representative motion vectors is susceptible to not always detecting the motion of interest and also being sensitive to camera panning, the later spectral estimation and filtering steps will to some extent suppress these effects. Whether a different method for calculating representative motion vectors can provide an improvement for the processing cost versus performance trade-off is dependent on the usage requirements and is a subject for future research.

III. SPECTRAL ESTIMATION AND FILTERING

The representative motion vectors that have been obtained can now work as a basis for a spectral estimation in order to examine the frequency distribution of the dominant motion in a video sequence.

A. Spectral estimation

To get a spectral estimation of the frequency content, the video material was divided into 16 second subsequences (SS)\(^2\). To perform the spectral estimation a periodogram is used. For the frequencies considered here, the effect of spectral leaking due to the use of a finite series is considered unimportant. By using various amounts of data it is possible to perform a trade-off between accuracy and computational requirements. We have used the following five methods to estimate the frequency content:

1) **FBF**. In this case the spectrum is computed based only on the FBF representative vectors. The frequency content is calculated by the equation:

\[
\phi_{FB}(\omega) = \frac{1}{\sqrt{N}} \left| \sum_{t=1}^{N} y_F(t) e^{-i\omega t} \right|^2
\]

where \(y_F(t)\) is the representative motion vector of the first forward predicted B frame.

2) **SBB**. Here, the SBB data is used instead of the FBF data. The spectrum is then estimated similarly to (8).

3) **ASFBFSBB**. Averaging the spectrum of FBF and SBB. Here, the spectra for the FBF and SBB are both calculated, and then averaged together. This can be calculated by:

\[
\phi_{ASFBFSBB}(\omega) = \frac{1}{2} \left\{ \phi_{FB}(\omega) + \phi_{SBB}(\omega) \right\}
\]

4) **ADFBFSBB**. Averaging the data values from FBF and SBB. In this case, the FBF and SBB values are averaged

\(^2\)Initial testing was performed for a number of interval lengths. 16 seconds was found to be a suitable length.
before the spectrum is computed. It is this averaged value that is then used in the spectral estimation. The equation below is used to calculate the frequency content:

\[
\phi_{pDA}(\omega) = \frac{1}{2\sqrt{N}} \left| \sum_{t=1}^{N} (y_F(t) + y_B(t)) e^{-i\omega t} \right|^2
\]  

(10)

5) IDFBFSBB: Interlacing the data values from FBF and SBB. From Nyquist sampling criteria, we know that the more sampled data that is available, the more accurate the estimate will be. So if the FBF and SBB data values are interlaced, taking care of the sign, a more comprehensive spectral estimation can be created. To calculate the frequency content the following equation is used:

\[
\phi_{pI}(\omega) = \frac{1}{\sqrt{2N}} \left| \sum_{t=1}^{2N} g(t) e^{-i\omega t} \right|^2
\]  

(11)

For a 16 second SS, the input data processed is 128 representative motion vectors for FBF, SBB, ADFBSBB and IDFBFSBB, and 256 vectors for IDFBFSBB. The x and y parts of the motion vectors were then processed separately to get the frequency content for movement in the horizontal and vertical directions. From the above five different algorithms, we get different results for the same video sequences. The choice of which method is best will depend on the trade-off between computational complexity versus accuracy. An example of the IDFBFSBB spectral estimate is shown in Figure 3, which gives the frequency content of one SS from the video shown in Figure 1. The x-axis shows frequency going from 0 to 0.5 times the base sampling frequency, so the figure shows the mirrored frequency content between 0 and 4 Hz.

B. Spectral filtering

It should be stressed that when designing the window, it is necessary to ensure that it is positive definite. To guarantee this characteristic we can form our window sequence, i.e., \( W(\omega) \) as the square of the Fourier transform of another window sequence \( v(k) \), let \( v = [v(0), v(1), v(2), ..., v(M-1)]^T \), implying

\[
V(\omega) = \sum_{k=0}^{M-1} v(k) e^{-i\omega k}
\]  

(12)

and set

\[
W(\omega) = |V(\omega)|^2
\]  

(13)

thus yielding

\[
w(k) = \sum_{n=0}^{M-1} v(n) v^*(n - k)
\]  

(14)

This derivation is valid under the assumption that \( v(k) = 0 \) for \( k < 0 \) and \( k \geq M \). Here, the symbol \((\cdot)^*\) will be used to denote the complex conjugate of a scalar variable or conjugate transpose of a vector or matrix, and \((a+b)\) is used to represent the convolution of \( a \) and \( b \). The objective here is to reduce the leakage acquired by \( w(k) \) as much as possible; this can be formulated as the problem of minimizing the relative energy in the side lobes of \( W(\omega) \) or as the problem of maximizing the relative energy in the main lobe of \( W(\omega) \), i.e.,

\[
\max_v \left\{ \frac{\int_{-\beta \pi}^{\beta \pi} |W(\omega)|^2 d\omega}{\int_{-\beta \pi}^{\beta \pi} |V(\omega)|^2 d\omega} \right\}
\]  

(15)

The above optimization problem can be written as

\[
\max_v \left\{ \frac{v^* \Gamma u}{v^* v} \right\}
\]  

(16)

where

\[
\Gamma = \frac{1}{2\pi} \int_{-\beta \pi}^{\beta \pi} a(\omega) a^*(\omega) d\omega
\]  

(17)

\[
a(\omega) = [1, e^{-i\omega}, e^{-i2\omega}, ..., e^{-i(M-1)\omega}]^T
\]  

(18)

The baseband equivalent solution of the optimization problem in (16) is given by the dominant eigenvector, i.e., the eigenvector corresponding to the maximum eigenvalue, of the matrix \( \Gamma \) defined in (17) [7]. For a given \( M \) and \( \beta \) values, the eigenvectors and eigenvalues of \( \Gamma \) can be computed as follows: Let

\[
\beta \geq \frac{1}{M}
\]  

(19)

and define

\[
K = M \beta \geq 1
\]  

(20)

For reasonably large value of \( M \), i.e., the data length, \( \Gamma \) can be approximated by

\[
\Gamma \simeq \frac{1}{2\pi} \sum_{i=\beta}^{K-1} a(\frac{2\pi}{M} i) a^*(\frac{2\pi}{M} i) \frac{2\pi}{M} = \Gamma_o
\]  

(21)

The matrix \( \Gamma_o \) has \( M \) eigenvectors, known as the Slepian sequences, \( S_k \), and has \( K \) eigenvalues equal to unity and \( M - K \) eigenvalues equal to zero. Thus, there will be \( K \) dominant eigenvectors, each of which has a bandwidth of \( \frac{1}{K} \).
with center frequency $\frac{2\pi}{M}$. The set of these K eigenvectors cover the baseband $(-\beta\pi, \beta\pi]$ [7]. From this analysis, it is possible to conclude that the eigenvalues and the eigenvectors of $\Gamma_o$ and $\Gamma$ are closely similar. Therefore, for large values of $M$, the matrix $\Gamma$ has K eigenvalues close to unity and $M - K$ eigenvalues close to zero.

The superposed power spectrum of the first K Slepian sequences is calculated by

$$\phi_{un}(\omega) = \sum_{p=1}^{K} \left\{ \frac{1}{M} \left| \sum_{t=1}^{M} v_p(t)e^{-i\omega t} \right|^2 \right\}$$  \hspace{1cm} (22)

The above equation employs unwindored periodogram, i.e., implicitly rectangular window, for the data sequences of $v_p(t)$. However, when estimating the spectrum it is advisable to choose a window for the whole data set in order to reduce the sidelobes. For this reason, the Hamming window, $h(t)$, is introduced as shown below:

$$h(t) = \begin{cases} 
0.54 - 0.46\cos\left(\frac{2\pi t}{M-1}\right) & \text{for } 0 \leq t < M \\
0 & \text{Otherwise}
\end{cases}$$  \hspace{1cm} (23)

Hence, the spectrum (22) using the Hamming window is given by

$$\phi_{UnmodulatedWindow}(\omega) = \sum_{p=1}^{K} \left\{ \frac{1}{M} \left| \sum_{t=1}^{M} v_p(t)h(t)e^{-i\omega t} \right|^2 \right\}$$  \hspace{1cm} (24)

where $v_p(t)$ is the $p_{th}$ Slepian sequence.

C. Window parameters and modulation

To parametrize the window appropriately it is necessary to know the frequency range that should be filtered out. To get a better estimation of the frequency range of indecent repetitive motion, an example dataset of decent and indecent videos were used to derive spectral content. This dataset was used mainly to derive values for the parameterization during testing, and to design a more general detector a larger dataset should be used.

Figures 4 and 5 display the histograms resulting from the examined decent and indecent videos. For both decent and indecent video there is considerable repetitive motion for frequencies below 0.5 Hz while there is insignificant motion above 1.8 Hz. However, in the case of the indecent videos in the example dataset there are significant repetitive motion around the frequency 1.4 Hz in both X and Y directions. Consequently, spectral energy in the region around 1.4 Hz can be a fairly reliable discriminator between decent and indecent video material for the example data set. Windowing the spectrum around 1.4Hz thus seems to be an appropriate solution in order to extract the spectral information relevant for classification of the video. It is desirable to design a filter which has a peak location around 1.4 Hz and which passes the signal 1.25 Hz to 1.54 Hz as undistorted as possible while attenuating the frequencies outside this interval as much as possible. This requirements needs to be translated into suitable values for the user parameters $M$ and $K$. The bandwidth ($BW$) of the required baseband equivalent window is given as

$$BW \approx \frac{1}{2} \frac{(1.54Hz - 1.25Hz) \pi}{8Hz} = 0.0091\pi$$  \hspace{1cm} (25)

In order to "cover" the frequency range of interest with the spectral components of the first K Slepian sequences, the minimum required $\beta$ is given as

$$\beta = \frac{K}{M} = 0.0091$$  \hspace{1cm} (26)

Setting the minimum value to $K = 1$ results in $M \approx 110$; however, choosing $K$ to some value larger than unity will reduce the variance of the spectral window; then, $K = 2$ and $M \approx 220$ can be another possible parameters. For comparison, the superposed power spectrum of the first K Slepian sequences for some values of $M$ and $K$ are shown in Figure 6. As can be seen in Figure 6, when $K$ is increased the spectral magnitude at the center (zero) will decrease, which is contrary to what is desired. For this reason, $K = 1$ is selected.
However, when $K = 1$, setting $M = 110$ results in a spectral magnitude at $f = 0.0091\pi$ compared with the spectrum at the center (0 Hz) that is less than -3dB. The falloff is too steep, and to remedy this the $M$ value can be adjusted to until the power spectrum of the first Slepian sequence at $f = 0.0091\pi$ is around -3dB. Setting the parameters to $K = 1$ and $M = 64$ were deemed sufficient for the bandwidth requirement.

After the baseband equivalent window function is derived, superimposing (shifting) it on to the carrier frequency ($f_c = 1.39\ Hz = 0.087\pi$) would place the window appropriately. Let us assume $\cos(2\pi f_c t)$ is the carrier signal then its spectrum becomes

$$\phi_{\text{carrier}}(\omega) = \frac{1}{M} \sum_{t=1}^{M} \cos(2\pi f_c t) h(t) e^{-i\omega t}^2$$

(27)

The time and frequency convolution property states that if

$$g_1(t) \iff G_1(\omega) \text{ and } g_2(t) \iff G_2(\omega)$$

(28)

then

$$\begin{align*}
\text{time domain} \\
g_1(t) * g_2(t) &\iff G_1(\omega) G_2(\omega)
\end{align*}$$

(29)

and

$$G_1(\omega) * G_2(\omega) \iff g_1(t) g_2(t)$$

(30)

where $M$ is the data length, and $G_1(\omega)$ and $G_2(\omega)$ are the Fourier transforms of $g_1(t)$ and $g_2(t)$, respectively.

From the above time and frequency convolution property the spectrum of the modulated window is calculated as

$$\phi_{\text{ModulatedWindow}}(\omega) = \phi_{\text{carrier}}(\omega) \phi_{\text{UnmodulatedWindow}}(\omega)$$

(30)

Figure 7 shows the detailed spectrum of the modulated window.

To classify a SS as decent or indecent, it’s spectrum is estimated according to one of the five approaches discussed in Section III-A. The spectrum is passed through the modulated window (30). If we define the spectrum of a SS as

$$\phi_{SS}(\omega) = \frac{1}{M} \left| \sum_{t=1}^{M} (y(t)h(t)e^{-i\omega t}) \right|^2$$

(31)

then the windowed spectrum of the SS is given by

$$\phi_{\text{WindowedSS}}(\omega) = \phi_{\text{ModulatedWindow}}(\omega) \phi_{SS}(\omega)$$

(32)

Therefore, there is an indication of offensive content in a SS when the peak amplitude of $\phi_{\text{WindowedSS}}(\omega)$ exceeds some threshold value. The setting of the threshold value allows a trade-off between true positive and false positive to be performed, as illustrated in Section IV-A.

IV. INITIAL EVALUATIONS

To examine the performance of the proposed approach some initials measurements were made. All results were obtained based on a video collection of more than 400 indecent video files and around 350 decent video files of varying lengths. Evaluations were made both to examine the frequency content for establishing a frequency windowing region, and to evaluate the performance of the repetitive windowing region as well as indecent video detection.

A. Repetitive motion detection performance

Once the appropriate frequency range had been established, an appropriate windowing function could be devised as discussed in Section III-C. By applying the windowing function to the spectral contents of a 16-second SS, the absolute value of the peak amplitude after windowing can be established. Since the windowing function suppresses all repetitive motion outside of the frequency region that is significantly different between decent and indecent movies, the absolute value of the
peak amplitude after windowing can be used as a marker to discriminate between likely indecent and likely decent video sequences. The effectiveness of the different spectral estimation approaches can thus be examined. Figure 8 shows the receiver operating curve (ROC) for different spectral estimation methods. It is clearly seen that the IDFBFSBB approach has superior performance, but it also has a higher computational cost. The results for the FBF and SBB approaches were similar to the results for ASFBFSBB. It should be noted that the presented results should be considered indicative since they are based only on limited video material. Additional experiments using additional video material are needed to get a wider picture of the performance the proposed approach.

![ROC curve for repetitive movement detection](image)

Fig. 8. ROC curve for repetitive movement detection

B. Computational complexity

It is noteworthy that the majority of the required processing will come from the Huffman decoding necessary to extract the motion vectors, the processing needed for filtering and calculating the representative motion vectors, and only a minority from the spectral estimation. An overview calculation of the computational requirements for the different methods is provided in Table I. The complexity is expressed as the number of operations (NOS) per 16-second subsequence. The formulation of the complexity uses N, the number of motion vectors horizontally in a frame, and M, the number of motion vectors vertically in a frame. For instance, FBF requires 17 NM to filter vectors (since each motion field requires 8 spatial and 9 temporal comparisons), 1 NM to calculate the gradients, 1 NM to calculate the compensated motion vectors, and 1 NM to compute translational representative motion vector (sub-total = 20 NM in each direction). Thus, for the 16 second frame sequence, there will be in total = 2X16X8X20 = 5120 NM additions and 128log(128)=896 multiplications (to estimate the spectrum) required. The computational cost thus includes the approximate number of operations necessary to filter the frame sequences, calculate the representative motion vector and to calculate the spectrum. It does not include the operations to decode the video to access the motion vectors. Note that no inverse DCT needs to be done since only the motion vectors are used. Considering the performance results shown earlier, the IDFBFSBB method seems to provide the best balance between detection performance and computational cost.

V. Conclusions

We have in this paper described an effective and robust motion vector based method for classifying video contents as decent or indecent. The method is based on filtering out the representative motion and then using spectral estimation by means of a periodogram to detect repetitive motion in a specific frequency band during 16 second intervals. Several spectral estimation approaches were devised and examined. An initial experimental evaluation indicate that spectral estimation based on the interlaced first B-frame forward predicted and second B-frame backward predicted motion vectors (IDFBFSBB) provides the best trade-off between detection probability, false alarm rate, and computational requirements. Compared to other alternative approaches, the initial results suggest that the proposed method can quite reliably classify video content as decent or indecent at a low computational cost, allowing classification to be performed considerably faster than real time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Additions</th>
<th>Multiplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBF</td>
<td>5120</td>
<td>896</td>
</tr>
<tr>
<td>SBB</td>
<td>5120</td>
<td>896</td>
</tr>
<tr>
<td>ASFBFSBB</td>
<td>10240</td>
<td>1792</td>
</tr>
<tr>
<td>ADFBFSBB</td>
<td>10368</td>
<td>896</td>
</tr>
<tr>
<td>IDFBFSBB</td>
<td>10240</td>
<td>2048</td>
</tr>
</tbody>
</table>

Table I Computational requirements

REFERENCES