Int. j. inf. tecnol. (March 2018) 10(1):21–34
https://doi.org/10.1007/s41870-017-0054-3

ORIGINAL RESEARCH

# An improved video key-frame extraction algorithm leads to video watermarking

Soumik Das[1] · Monalisa Banerjee[2] · Atal Chaudhuri[1]

Received: 5 July 2017 / Accepted: 26 October 2017 / Published online: 7 November 2017

**Abstract** The epidemic growth of internet services has created a broad attention on digital content authentication and copyright protection. Heading the issue we are going to propose a DCT based novel approach on invisible video watermarking along with an improved video key-frame extraction algorithm. In this article first the key-frames (video bookmarks) of the video sequence are being extracted. In order to identify the video bookmarks, a novel method is proposed based on boundary luminosity analysis that is proven efficient regarding fast camera movement and moving objects in the video frames. A pair of secret keys will be applied during the watermark embedding phase. Our framework embeds watermarks in only those video frames which have been identified as key-frames of that video stream because, the key-frames are mostly kept as it is in different compression algorithms. The low frequency DCT coefficients of key-frames are chosen for watermarking, that again offers robustness to the compression algorithms. Instead of embedding watermark bits directly, a new method is proposed- scaled average. The watermark is perceptually invisible in the video key-frames according to human visual system. At last uncompressed video is re-reconstructed with all frames of the video in appropriate order of appearance. The watermark extraction is blind i.e., except the keys nothing is needed for watermark extraction.

**Keywords** Video key frame · Video watermarking · Frequency domain · DCT · Blind extraction

## 1 Introduction

In our proposed framework a video stream is watermarked with a DCT based watermarking technique. Two algorithms are executed in the proposed framework. One will extract the key-frames of the video in which the other will embed watermark invisibly. The available key-frame extraction algorithms are generally used to summarizing video for online streaming. In the current communication we are going to propose an efficient framework for key-frame extraction—Boundary Luminosity Analysis.

The key aesthetics of our key-frame extraction (video bookmark) algorithm are:

- Scene segmentation needs not to be performed.
- Very efficient for the scene having fast camera movement.
- Efficient for the scene having moving objects.
- Target number of key frames needs not to be specified.

The watermark embedding algorithm will be executed on the output key-frames of video bookmark algorithm with the scrambled watermark. The proposed DCT based invisible watermarking framework performs 8 × 8 block DCT on the video key-frames to embed a binary watermark into it. The binary watermark is scrambled with a secret key to employ additional security. The watermarking information is embedded into one of the seven low

✉ Soumik Das
soumikdas.techno@yahoo.co.in

Monalisa Banerjee
monalisa_erjee@yahoo.com

Atal Chaudhuri
atalc23@gmail.com

[1] Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, India

[2] Master of Computer Application, Techno India Salt Lake, Salt Lake, West Bengal, India

frequency coefficients of each 8 × 8 block depending on another secret key. Instead of embedding watermark bits, we have proposed a new phenomenon called scaled average.

After embedding watermark into the key-frames no significant change will be observed as we've exploited the low frequencies which are less sensitive to the human psycho-visual system. Now all the frames of the video sequence including key-frames will be collaborated in the same order of their appearance for the uncompressed video reconstruction. At the extraction end again the key-frame extraction algorithm will be applied on the watermarked video to identify the key-frames and the same pair of secret keys should be provided to extract the watermarks from the respective key-frames.

In the next section we've reported the literature survey on the related works. Sections 3, 4 and 5 elaborates the proposed "video bookmark", watermark embedding and extraction framework, respectively. Result set analysis is depicted in Sect. 6 and conclusions are being made in Sect. 7.

## 2 Literature survey on related work

Different methods can be adopted for identifying the key frames. Pixel-Compare method is one of them. In this method every consecutive frames is compared pixel wise and when the comparison difference crosses a given threshold, system identifies that frame as key frame. But this method is highly time consuming and over sensitive to motion of objects in the frame. A video is a collection of different scenes. The most common approach is to select first frame of each scene as key-frame [1]. Ueda et al. [2] and Rui et al. [3] have considered first and last frame of each scene as key-frames. Pentland et al. [4] identified the key-frames after a specific time interval within the shot. These approaches rely only on the information regarding the video sequence but do not consider the visual dynamics of the scene. They often extract a fixed number of key-frames out of a scene. Zhonghua et al. [5] prescribed a method where a single frame is taken as key-frame out of a scene. The frames are segmented first into objects and background and the frame having highest ratio of object to background is chosen as the key-frame of that scene assuming that frame conveys the most information about the shot.

If we take visual dynamics of scene into consideration one simpler approach is to calculate the difference between two frames in terms of some visual characteristics like—histograms, motion, pixels etc. Zhao et al. [6] described a simple method called "Simplified Breakpoints" where a frame is selected as key-frame if its colour histogram differs by a given threshold from previous frame. But this method doesn't sound good because two completely different frames may result into similar histograms. Hanjalic et al. [7] proposed an algorithm called "Flexible Rectangle" (FR) algorithm where they have taken frame difference to build a "content development" curve composed of a defined number of rectangles which are approximated using error minimization algorithm. Hoon et al. [8] have selected the key-frames by "Adaptive Temporal Sampling" (ATS) algorithm which uniformly samples the cumulative frame differences along y-axis of the curve and non-uniform sampling along x-axis of the curve represents the key-frames.

In compressed domain the development of key-frame extraction algorithms are often considered as they allow expressing the motion dynamics of a scene through motion vector analysis. Narasimha et al. [9] have proposed a fuzzy system for MPEG video that classifies the motion intensity of frames into five categories and the frames exhibit high intensities are considered as key-frames. Liu et al. [10] have developed "Perceived Motion Energy" (PME) algorithm that computes the PME of the motion vectors used to describe the video content.

The main drawback to most of these methods is the number of representative frames must be set depending on the length of video sequence. This approach doesn't guarantee that representative frames (key-frames) will not be highly correlated. It is also difficult to set a suitable interval of time and as a result of that small intervals may not extract enough representative frames [11]. A comparative study in this regard has been reported at Table 2 of this paper.

With respect to identify a robust watermarking framework, Cox et al. have proposed a global DCT based watermarking technique in 1997 [12]. They first suggested that the watermark could be embedded in the low frequency bands. Although low frequency coefficients are very sensitive to HVS but at the same time it is also true that most compression techniques reduce the insubstantial parts like—LSB in spatial domain and high frequencies in frequency domain. They proposed a spread spectrum watermarking technique where NxN DCT is performed on a NxN image to obtain NxN coefficients (Global DCT). Their novelty in proposing the low frequency is more robust than high frequencies is well accounted because, in compression techniques the high frequencies are being discarded. So if we embed the watermark data into the high frequencies those will be lost if a compression is performed even with a high quality factor (i.e., less compression). But their approach of performing Global DCT is surely reducing the watermarking capacity of the proposed technique because, performing NxN DCT results only NxN coefficients. Out of those a few will be of low frequencies.

So Global DCT is undoubtedly decreasing the capacity of embedding watermark.

In the year 2006, Yuan et al. proposed a multipurpose watermarking algorithm for copyright protection and authentication [13]. The main idea is to embed the robust and fragile watermarks into different color components simultaneously. The fragile watermark is embedded in spatial domain of Blue component using conventional LSB algorithm to achieve the excellence in content authentication whereas the robust watermark is embedded in the frequency domain of Green component to obtain the goal of copyright protection by modifying the Discrete Cosine Transform (DCT) coefficients. The idea of embedding watermark in spatial and frequency domain altogether really sounds well. Watermarking in 'Blue' channel is also well counted because 'Blue' is least sensitive to HVS if spatial domain is concerned. If spatial domain watermark is extracted intact then it would be understood that there is no attack is performed because, spatial watermarks are very fragile in nature. But embedding another watermark at frequencies of 'Green' using DCT would not be efficient because, RGB color space is highly correlated and that's why it is not considered in frequency domain watermarking. Instead YCbCr space is more suitable for such coding.

In 2009 Lin et al. have proposed another idea of watermarking that was claimed robust against compression [14]. They have also accepted the idea proposed by Cox et al. [12]—Low frequency coefficients can offer more robustness than high frequencies against compression algorithms. According to their framework the host is transformed from RGB to YCbCr color space before frequency domain coding and Y part is considered for watermarking. They've performed 8 × 8 block DCT and then quantized by standard quantization matrix. They claimed quantizing blocks prior to watermarking gives additional robustness against compression. But quantization at the watermarking end is unacceptable because, quantization process discards many high frequencies (i.e., loss of information) so deteriorating the quality of content at the time of watermarking is not at all accepted. They've identified the low frequency DCT coefficients at positions C(2, 0), C(1, 1), C(0, 2), C(0, 3), C(1, 2), C(2, 1), and C(3, 0). Out of these only two coefficients C(0, 2), C(2, 0) are considered for embedding watermark bits. Choosing only two coefficients may lead damage to the watermark by any counterfeiter because, even if these two frequencies are scaled (up or down) by a minimum amount, the watermark will be severely damaged.

In 2012 Deb et al. have proposed a combined DWT and DCT based watermarking technique with low frequency watermarking with weighted correction [15]. DWT has excellent spatial localization, frequency spread and multiresolution characteristics, which are similar to the theoretical models of the human visual system (HVS). DCT based watermarking techniques offer compression while DWT based watermarking techniques offer scalability. The proposed method embeds of watermark bits are in the low frequency band of each DCT block of selected DWT sub-band. The weighted correction is also used to improve the imperceptibility. Choosing low frequency band for watermark embedding like [12, 14] surely enhances the robustness of the scheme under various attacks but the quality of watermarked content is not that well obtained with respect to the PSNR reported. The security issues are not being taken into consideration.

In the year of 2013 Raval et al. have proposed another approach of frequency domain watermarking through combined DWT-DCT [16]. They perform DWT on the host then again DCT is applied on the decomposed sub bands. To make their framework robust against compression they passed the FDCT data into EBCOT (embedded block coding optimal truncatation) algorithm. After receiving, the algorithm outputs the binary watermark bits which are embedded into the frequencies of original content. They did not mention the desired frequency region for watermarking. Considering high frequencies for watermarking surely decreases the robustness.

In 2016 Zong et al. [17] have proposed DCT based method for watermarking. In the watermark embedding process, the host is divided into blocks, followed by the 2-D DCT. A secret key is applied to each block to randomly select a set of DCT coefficients of middle frequency for watermark embedding. Watermark bits are inserted into an block by modifying the set of DCT coefficients with the help of an error buffer to deal with errors caused by attacks. Since the proposed watermarking method only uses two DCT coefficients (of middle frequencies) to hide one watermark bit, it has a limitation in hiding the watermark of bigger size. That means the size of watermark and host ratio should be moderate enough. Though they have claimed that their method is robust against compression but using middle frequencies for watermark embedding may not sound good against higher degrees of compression. As they've used only two coefficients per block to embed watermark, the low frequency components are suggested to be good in the trade-off between imperceptibility and robustness.

Keeping all the aforementioned limitations in account, a DCT based invisible watermarking framework is proposed hereinafter which is robust against compression algorthms, other leading attacks. In this work we have performed 8 × 8 block DCT on the extracted key-frame to embed a binary watermark into it. The binary watermark is scrambled with a secret key to employ additional security. The watermarking information is embedded into one of the seven low frequency coefficients of each 8 8 block depending on another secret key. Instead of embedding watermark bits, we have proposed a new phenomenon called scaled average. The watermark extraction is blind and the same set of secret keys is needed to extract the watermark.

In current communication we propose a different approach to identify the key-frames that do not suffer from any known constraint like—predetermination of number of key-frames, length of the video sequence, over sensitive to moving objects of frame, scene change due to slow camera pan etc., along with a robust DCT based watermarking framework.

## 3 Video bookmark framework

In the current context we are going to discuss "video bookmark" algorithm which has to be executed first. It takes a video sequence as input and extracts the key-frames (video bookmarks) as output. The second algorithm takes key-frames and the binary watermark as input and produces the watermark embedded uncompressed video sequence. Figure 1 shows the schematic diagram of the entire framework.

In the current communication a novel framework is proposed which will perform the key frame extraction more accurately than other available frameworks and less consumable regarding time—oundary Luminosity Analysis. According to proposed framework the mean luminosity of boundary regions of the video frames will be calculated and analyzed. Because for two consecutive frames from a same scene the boundary regions are likely to be identical. To demonstrate the proposed algorithm a 720 × 405 (Aspect Ratio 16:9) resolution video sequence of 35 s from a movie named "Cast Away" is considered and 840 frames have been extracted at 24 frame per second (fps) [18].

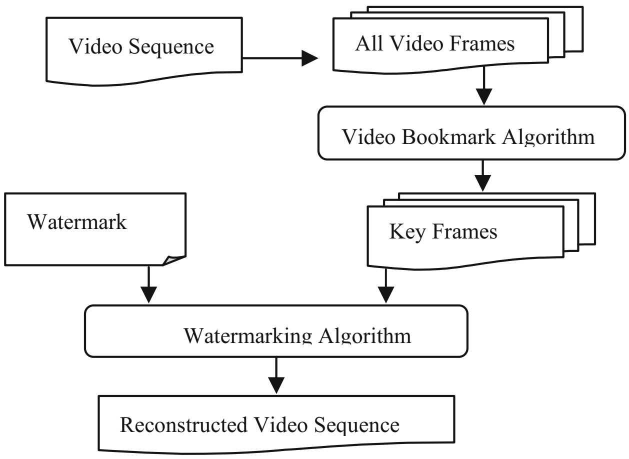


**Fig. 1** Schematic diagram of the proposed framework

For calculating the mean luminosity of boundary regions of video frames n × n pixel blocks are considered say $\Delta b_i$ from the top-left corner of the frame to rightwards like—$\Delta b_1, \Delta b_2, \Delta b_3, \ldots, \Delta b_n$. Figure 2 demonstrates how the blocks $\Delta b_i$ are being taken at the boundary regions. The luminosity of each $\Delta b_i$ is calculated with Eq. 1 and finally through Eq. 2 the mean boundary luminosity (mBL) of a video frame is obtained. The luminance Y can be calculated through weighted Red (R), Green (G) and Blue (B) values as described in Eq. 1 [26].

$$Y_{\Delta \mathrm{bi}} = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} (0.299R + 0.587G + 0.114B) \tag{1}$$

Where, each pixel is $P(x,y)$ and there are $n \times n$ number of pixels in each $\Delta b_i$.

$$mbl = \frac{\sum_{i=1}^{n} luminosity\Delta b_i}{28} \tag{2}$$

Some of the selected frames of the said video sequence are depicted in Fig. 3. And Fig. 4 shows how the mean boundary luminosity (mBL) varies frame to frame. Frame 5 and frame 6 are two consecutive frames from a same scene having difference between mean of boundary luminosity ($\mathrm{mBL}_{\mathrm{DIFF}}$) of 0.07 whereas Frame 6 and Frame 7 having mean boundary luminosity difference of 7.23. So in the continuous sequence of these 3 frames, frame 3 will be extracted as the key frame. In the proposed framework the difference of mean boundary luminosity will be analyzed in two ways considering two different situations:

*Situation 1:* When a key frame appears as a result of cut to a new scene the key frame can be identified by analyzing two consecutive frames. For example as stated above frame 7 appeared as a result of cut to a new scene from the immediate previous frame 6. A threshold $\alpha_1$ is chosen in order to identify the key-frame.

That means if two consecutive frames having difference of mean boundary luminosity ($\mathrm{mBL}_{\mathrm{DIFF}}$) less than $\alpha_1$ then they will be considered as frames of same scene. The

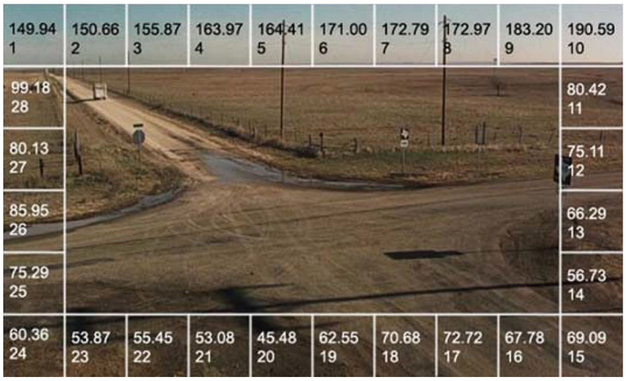


**Fig. 2** Allocation of $\Delta b_i$ to calculate luminosity of each $\Delta b_i$

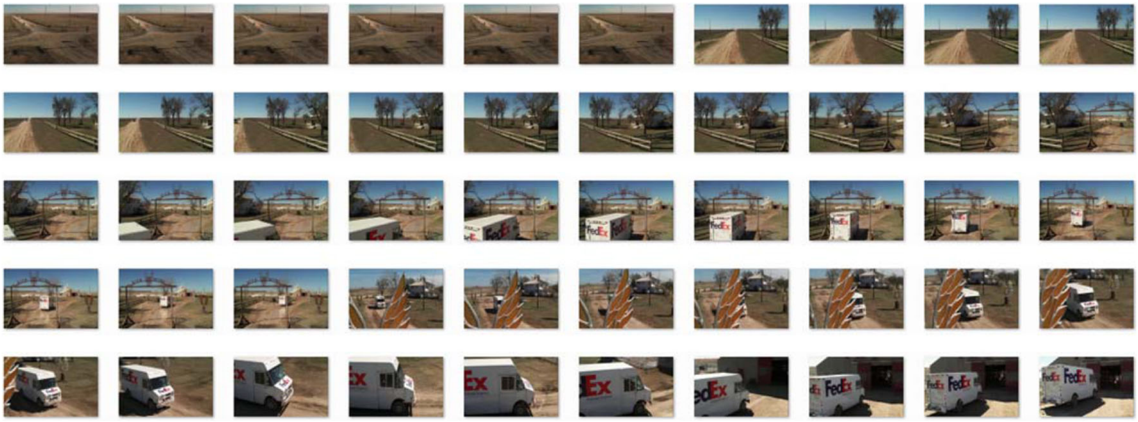
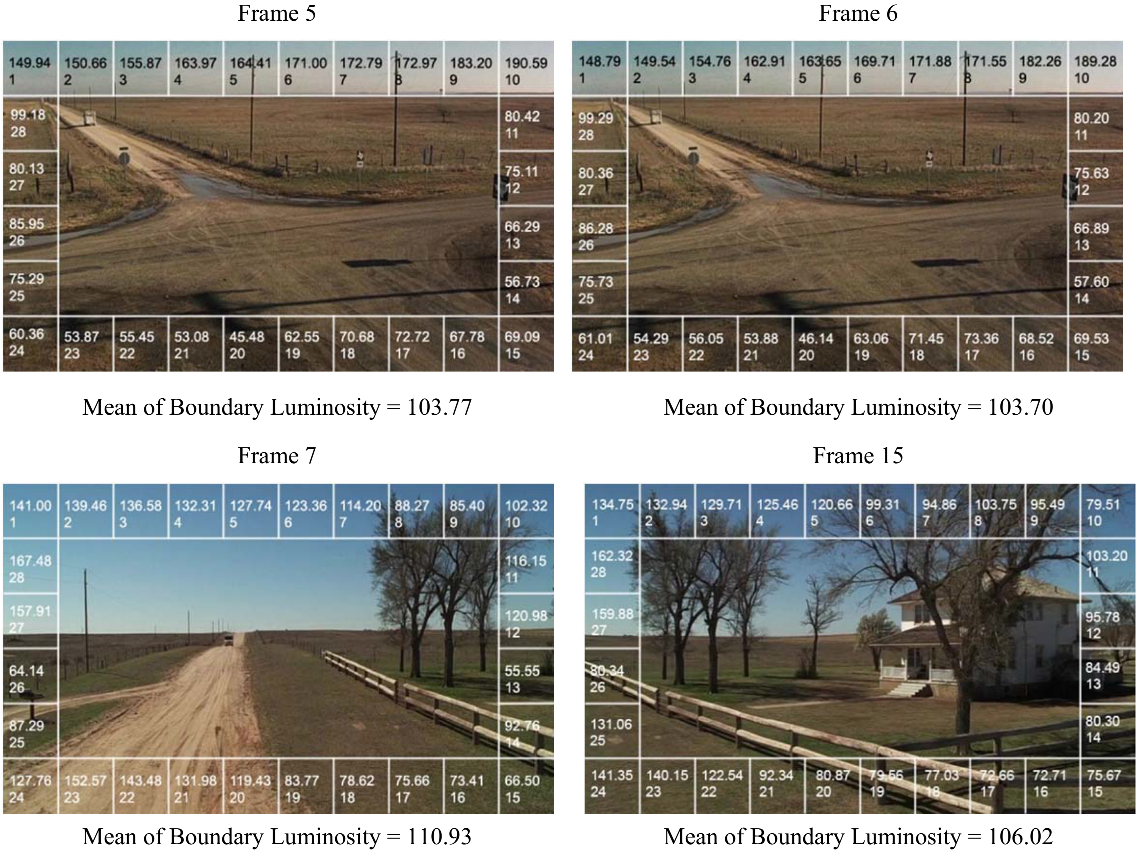


**Fig. 3** Some selected continuous frames of video sequence from the movie “Cast Away”



Frame 5

Mean of Boundary Luminosity = 103.77

Frame 6

Mean of Boundary Luminosity = 103.70

Frame 7

Mean of Boundary Luminosity = 110.93

Frame 15

Mean of Boundary Luminosity = 106.02

**Fig. 4** Variation of mean boundary luminosity (mBL) from frame to frame

second frame will be considered as key-frame (video bookmark) if it is greater than or equal to $\alpha_1$. After performing a number of experiments $\alpha_1 = 2$ is suggested.

*Situation 2:* The previous method of analysis will not sound good if a slow camera pan occurs. In this case key frames will be appeared without varying the $\mathrm{mBL_{DIFF}}$ more than $\alpha_1$. For example consider frames from 7 to 15. If we analyzed $\mathrm{mBL_{DIFF}}$ of 7–8, 8–9, 9–10 so on till frame 15; the $\mathrm{mBL_{DIFF}}$ will not cross the threshold of 2. But clearly we can see the frame 15 is a key-frame after key frame 7 as a house and a tree have framed in at right side of frame 15. In order to remove this anomaly from the proposed system a new method of cross checking is adopted. After identifying a key-frame the mean boundary luminosity (mBL) of the key frame is stored till the next key frame is detected. The mBL of next key frame will replace the value of previous stored one. Every time the proposed system will calculate $\mathrm{mBL_{DIFF}}$ of two consecutive frames as per the 1st method and at the same time a calculation of $mBL_{DIFF}$ will be performed between the current frame and last identified key-frame. If the difference results greater than $\alpha_2$ (a value 4 is suggested for $\alpha_2$ after result set analysis of thousands of video streams) then the current frame will be identified as key frame and its mBL will be stored in the proposed analytical system. Continuing with the aforesaid example 1st method is failed to identify frame 15 as key frame but 2nd method is able to identify that as follows—

$$|mBL_{DIFF}|(\text{frame 7} - \text{frame 15}) = (110.93 - 106.02) = 4.91 > \alpha_2$$

### 3.1 Video bookmark algorithm

```
STEP 1:   Initialize
          KF = frame1, PF = frame1,
          CF = frame2, LF
STEP 2:   Declare $mBL_{KF}$, $mBL_{CF}$, $mBL_{PF}$, $mBL_{DIFF}$, $mBL_{KF\_DIFF}$
STEP 3:   FIND (LF)
STEP 4:   $mBL_{PF} = CALC_{MBL}$ (PF)
          $mBL_{KF} = CALC_{MBL}$ (KF)
STEP 5:   LOOP (CF=2; CF=< LF, CF++)
          {
          $mBL_{CF} = CALC_{MBL}$ (CF)
          $mBL_{DIFF} = | mBL_{CF} - mBL_{PF} |$
          $mBL_{KF\_DIFF} = | mBL_{CF} - mBL_{KF} |$
                IF ($mBL_{DIFF} >= \alpha_1$)
                {
                $mBL_{KF} = mBL_{CF}$
                $mBL_{PF} = mBL_{CF}$
                }ELSE IF ($mBL_{KF\_DIFF} > \alpha_2$)
                {
                $mBL_{KF} = mBL_{CF}$
                $mBL_{PF} = mBL_{CF}$
                }ELSE
                {
                $mBL_{PF}=mBL_{CF}$ }
          } END OF LOOP
```

Definitions of functions and variables used:

KF: key frame

PF: previous frame

CF: current frame

LF: last frame

$mBL_{KF}$: mean boundary luminosity of immediate KF

$mBL_{PF}$: mean boundary luminosity of PF

$mBL_{CF}$: mean boundary luminosity of CF

FIND (LF): this function finds the last frame of the video stream

$CALC_{MBL}$: this function calculates the mean boundary luminosity (mBL) of the desired frame

$mBL_{DIFF}$: this function calculates the modulus value of the difference mBL of previous and current frame

$mBL_{KF\_DIFF}$: this function calculates the modulus value of the difference mBL of immediate key frame and current frame

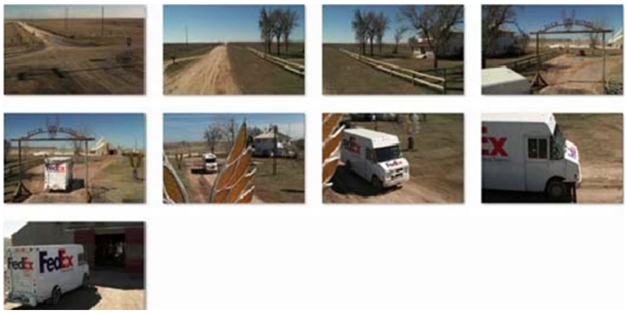


**Fig. 5** Key-frames extracted with video bookmark algorithm

Stepping in with our video bookmark algorithm 9 key-frames (video bookmark) has been extracted out of 840 video frames. See Fig. 5.

The next section demonstrates how we embed the binary watermark in these key-frames.

## 4 Watermark embedding framework

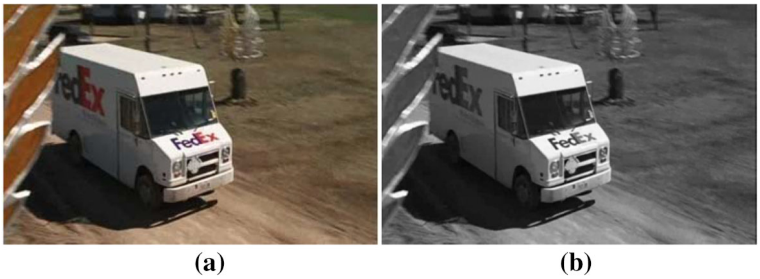
The steps of proposed watermark embedding algorithm are described below:

Input: key-frame (obtained from video sequence), Watermark, Secret key-1, Secret key-2
Output: watermarked key-frame

### 4.1 Color space transformation of key-frame—step 1

The Key-frame (H) will be transformed from RGB color space to YCbCr color space because RGB color space is highly correlated and not suitable for frequency domain watermarking such as DCT [19]. Y part is called the luminance component whereas the Cb and Cr parts are called blue chrominance and red chrominance, respectively. Although the luminance is much sensitive to HVS than the chrominance still the luminance (Y) channel of key-frame is considered for embedding watermark because JPEG compression discards a lot of chrominance information during chroma subsampling. So the watermark will not sustain against compression techniques if watermark is embedded at chrominance part. The transformation from RGB color space to YCbCr color space is done with following Eq. 3 [14]. Figure 6a shows the key-frame in RGB color space and Fig. 6b shows the luminance (Y) of the key-frame.

**Fig. 6** **a** RGB Color space of key-frame, **b** Y part of key-frame


(a)
(b)

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3}$$

### 4.2 Watermark scrambling—step 2

The binary watermark (w) is taken and scrambled by applying Secret key-1. The watermark is scrambled to employ enhanced security to the proposed watermarking system. Even if a counterfeiter able to extract the watermark from a watermarked key-frame, the scrambled watermark will be retrieved, not the original one. The binary watermark of size 256 × 256 is divided into sixty-four 32 × 32 non overlapping blocks. Depending on the 24-byte long Secret key-1 these sixty-four blocks get shuffled their positions as per our scrambling algorithm. With two different value of Secret key-1 such as $\beta_1$ and $\beta_2$ the watermark gets scrambled in absolutely different manner as shown in Fig. 7.

24 byte key will be divided into 64 groups where each group contains consecutive 3 bits as follows—

101| 010 | 001|111|110 |101|001|111|………

5|2|1|7|6|5|1|7|………

Each consecutive three bits can represent a range of 0–7 as above. Therefore, consecutive two numbers able to represent a particular block position as follows—

$(5,2)\,(1,7)\,(6,5)\,(1,7)\ldots$[Ranging from $(0.0)$ to $(7,7)$]

Now every pair of 2 consecutive blocks will be swapped provided either of the blocks is not swapped earlier. In the above example the block of position (5, 2) is swapped with (1, 7) but next pair (6, 5) will not be swapped with (1, 7) as because (1, 7) is already swapped with (5, 2). Continuing in this manner the logo will be scrambled.

#### 4.2.1 Scrambling algorithm

```
Input:     Watermark (256x256),Secret Key-1(β of 24 byte)
Output:    Scrambled Watermark (256x256)
STEP1:
LOOP (i=0; i<64; i++)
{
 xpos = i * 3;
 LOOP (a=0; a<3; a++)
  {
   Array1[a]= β[xpos+a];
   Array2[i] = decimal equivalent of
    Array[a] Array1[a+1] Array[a+2];
  }
}
STEP2:
LOOP (j=0; j<32; j++)
{
 ypos = j * 4;
 LOOP (b=0; b<4; b++)
  {
   p = Array2[ypos+b];
   q = Array2[ypos+b+1];
   r = Array2[ypos+b+2];
   s = Array2[ypos+b+3];
 If (flag[p][q]!=TRUE||flag[r][s]!=TRUE)
  {
   temp[0][0]  =  Block [p][q];
   Block[p][q] =  Block[r][s];
   Block[r][s] =  temp[0][0];

   flag[p][q] == TRUE;
   flag[r][s] == TRUE;
  }
 }
}
```

### 4.3 Texture localization—step 3

The scrambled binary watermark can have two possible pixel values—255 and 0. The pixels of scrambled

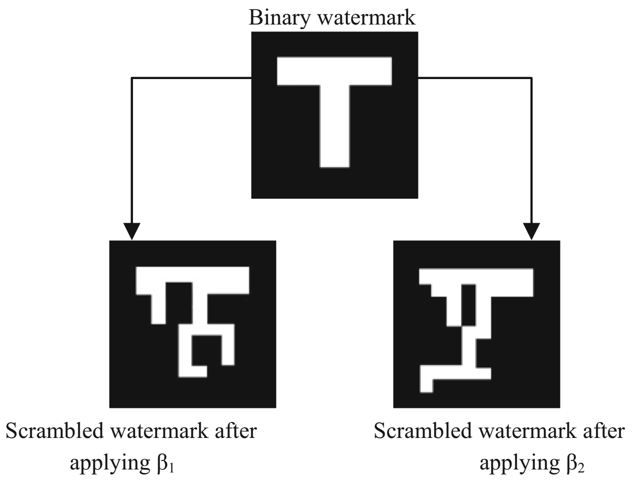

Binary watermark
Scrambled watermark after applying β₁
Scrambled watermark after applying β₂

**Fig. 7** Watermark scrambling with two different values of Secret key-1

watermark having value 255 are substituted with 0. On the other hand the pixels having value 0 are substituted by the Y values of the key-frame. Figure 8a–c is provided in this regard.

### 4.4 DCT of luminance of key-frame—step 4

Discrete Cosine Transform is a well-known method for signal decomposition that transforms an image from spatial to frequency domain. The DCT works by separating image into parts of differing frequencies. The forward DCT of an image will be achieved from Eq. 4 [20].

$$\mathrm{DCT}\ (\mathrm{i},\ \mathrm{j}) = \mathrm{C}(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\mathrm{pixel}\ (\mathrm{x,y}) \times \cos\left[\frac{(2\mathrm{x}\ +\ 1)\mathrm{i}\pi}{2\mathrm{N}}\right]\cos\left[\frac{(2\mathrm{y}\ +\ 1)\mathrm{j}\pi}{2\mathrm{N}}\right] \tag{4}$$

where,

$$C(i), C(j) = \begin{cases} \sqrt{\dfrac{1}{N}} & \mathit{for}\ i,j = 0 \\ \sqrt{\dfrac{2}{N}} & \mathit{for}\ i,j = 1,2,3\ldots N-1 \end{cases}$$

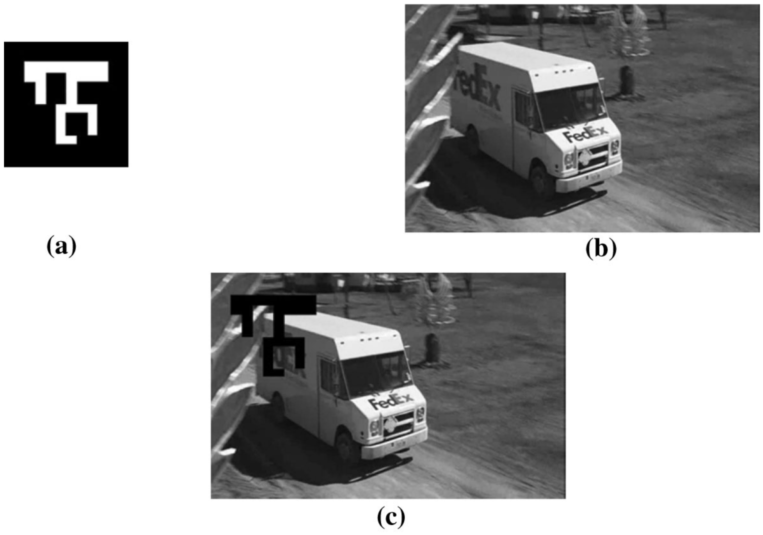
p(x,y) is the x, yth element of the key-frame represented by matrix p. N is the size of the block on that the DCT is done. Equation 4 determines one entry (i, jth) of the transformed key-frame from the pixel values of the original image matrix. In proposed framework the luminance part Y of the key-frame is divided into 8 × 8 (N = 8) non-overlapping blocks (equals to the size of the watermark) from the upper-left corner of the key-frame and forward DCT is performed on each individual block.

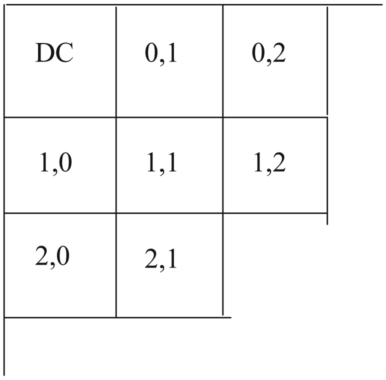
### 4.5 Encoding—step 5

Each 8 × 8 block is having 64 coefficients, out of these the (0,0) element is known as DC coefficient that has most significant information of that block. Other 63 coefficients are called AC coefficients where typically 7 coefficients from top-left corner of the block are considered as low frequency coefficients [14] such as—(0,1) (0,2) (1,0) (1,1) (1,2) (2,0) (2,1). The higher frequency coefficients are obtained traversing towards right-bottom corner of the block. In all compression technique, the high frequencies are being discarded because our psycho-visual system is less sensitive towards high frequencies [14, 19]. So choosing high frequency band for watermarking will surely lack robustness against compression techniques. That's why low frequency band is considered for hiding the watermark in current context.

Scaled average of two low frequency coefficients is calculated and another low frequency coefficient is

**Fig. 8** **a** Scrambled watermark, **b** Y part of key-frame, **c** texture localized watermark





| | | |
|---|---|---|
| DC | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | |

**Fig. 9** Low frequency coefficients of 8 × 8 block

substituted with the averaged value for such blocks where the value of DC coefficient is different in texture localized watermark and key-frame, i.e., the key-frame blocks where the watermark blocks are superimposed. On the other hand for the blocks where the value of DC coefficient doesn't differ will remain same. Figure 9 shows the low frequency coefficients of 8 × 8 block. A 384 byte long Secret key-2 is taken (formed by repeating Secret key-1 for 16 times) and applied to determine which coefficients are to be averaged and which one will be substituted by averaged value. Proposed operations are as follows depending on different values of Secret key-2. Now consider the following key—

101001100000111101110 011000110110. . .. . .. . .

This key will be divided into 1024 groups where each group is having three bits.

101| 001|100| 000|111|101|110| 011|. . ..

5 | 1 | 4 | 0 | 7 | 5 | 6 | 3 |. . .

Each of the 1024 values will be assigned to 1024 nos. of 8 × 8 blocks of the key-frame.

$$5 \rightarrow (0,0)\, 1 \rightarrow (0,1)\, 4 \rightarrow (0,2)\, 0 \rightarrow (0,3)\, 7 \rightarrow (0,4)\, 5 \rightarrow (0,5)\, 6 \rightarrow (0,6)\, 3 \rightarrow (0,7)$$

and so on.

Now consider, the DC coefficient of (0,0) block is different from DC coefficient of the same block of texture localized watermark. The (0,0) block will be encoded with the 5th rule as the key value 5 is assigned to (0,0) block. The set of rules are as follows—

$$\text{For assigned value } = 0/1(0,1) \leftarrow \frac{(1,0)+(1,1)}{1}$$

$$\text{For assigned value } = 2(1,0) \leftarrow \frac{(0,1)+(2,0)}{2}$$

$$\text{For assigned value } = 3(1,1) \leftarrow \frac{(0,2)+(1,0)}{3}$$

$$\text{For assigned value } = 4(0,2) \leftarrow \frac{(0,1)+(1,1)}{4}$$

$$\text{For assigned value } = 5(2,0) \leftarrow \frac{(0,1)+(1,0)}{5}$$

$$\text{For assigned value } = 6(1,2) \leftarrow \frac{(0,1)+(2,1)}{6}$$

$$\text{For assigned value } = 7(2,1) \leftarrow \frac{(1,2)+(1,0)}{7}$$

The following key-frame block analysis illustrates the encoding technique—

Let us assume that, block (6, 6) is such a block where DC value of the block is different in texture localized watermark and key-frame i.e., the block contains watermark information.

The luminance of aforesaid block be as follows—

| 166 | 192 | 160 | 119 | 94 | 73 | 43 | 27 |
|---|---|---|---|---|---|---|---|
| 194 | 190 | 139 | 100 | 67 | 44 | 33 | 26 |
| 189 | 157 | 115 | 81 | 47 | 30 | 30 | 26 |
| 162 | 122 | 93 | 55 | 35 | 33 | 31 | 25 |
| 127 | 101 | 70 | 38 | 34 | 34 | 27 | 24 |
| 113 | 79 | 44 | 30 | 32 | 32 | 27 | 25 |
| 93 | 49 | 28 | 30 | 30 | 28 | 28 | 29 |
| 62 | 28 | 32 | 37 | 32 | 26 | 27 | 31 |

After performing FDCT the coefficients are as follows—

| − 483 | 295 | 87 | 30 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 206 | 154 | − 7 | − 24 | − 30 | − 16 | − 10 | − 5 |
| 26 | − 21 | − 46 | − 14 | 0 | − 5 | − 5 | − 4 |
| 11 | − 9 | − 14 | 8 | − 13 | − 15 | − 2 | − 2 |
| − 1 | − 20 | − 22 | − 8 | − 10 | − 2 | 7 | 3 |
| 1 | − 6 | − 1 | 5 | 2 | 5 | 3 | 0 |
| 1 | − 3 | − 6 | − 2 | 0 | 0 | 0 | 0 |
| − 1 | − 2 | − 3 | − 2 | 1 | − 1 | − 1 | 0 |

Say the key value 2 is assigned for (6, 6), so according to our framework the values of coefficient (0, 1) and (2, 0) will be summed up and divided by the value 2. The resultant value will substitute the value of coefficient (1, 0). The following equation is used to perform the operation—

$$\text{For assigned value } = 2(1,0) \leftarrow \frac{(0,1)+(2,0)}{2}$$

$$(1,0) \leftarrow \frac{295+26}{2} = 160.5$$

After the encoding the watermarked block the coefficients are as follows —

| − 483 | 295 | 87 | 30 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 160.5 | 154 | − 7 | − 24 | − 30 | − 16 | − 10 | − 5 |
| 26 | − 21 | − 46 | − 14 | 0 | − 5 | − 5 | − 4 |
| 11 | − 9 | − 14 | 8 | − 13 | − 15 | − 2 | − 2 |
| − 1 | − 20 | − 22 | − 8 | − 10 | − 2 | 7 | 3 |
| 1 | − 6 | − 1 | 5 | 2 | 5 | 3 | 0 |
| 1 | − 3 | − 6 | − 2 | 0 | 0 | 0 | 0 |
| − 1 | − 2 | − 3 | − 2 | 1 | − 1 | − 1 | 0 |

Modified: (1, 0) = 160.5

### 4.6 IDCT—step 6

Inverse DCT needed to be performed on individual blocks after encoding; IDCT will be done according to Eq. 5 [20].

$$pixel(x,y) = C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} DCT(i,j) \cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{5}$$

where

$$C(i),C(j) = \begin{cases} \sqrt{\frac{1}{N}} & for\ i,j=0 \\ \sqrt{\frac{2}{N}} & for\ i,j = 1,2,3\ldots N-1 \end{cases}$$

The Luminance of watermarked block (6, 6) will be as follows—

| 158 | 184 | 152 | 112 | 86 | 65 | 35 | 19 |
|---|---|---|---|---|---|---|---|
| 187 | 183 | 133 | 93 | 60 | 38 | 27 | 19 |
| 184 | 153 | 111 | 77 | 43 | 26 | 26 | 21 |
| 160 | 120 | 92 | 53 | 33 | 32 | 29 | 23 |
| 128 | 103 | 72 | 39 | 36 | 35 | 29 | 26 |
| 118 | 84 | 49 | 35 | 36 | 37 | 32 | 30 |
| 100 | 56 | 35 | 37 | 36 | 35 | 35 | 36 |
| 69 | 36 | 39 | 45 | 40 | 34 | 35 | 39 |

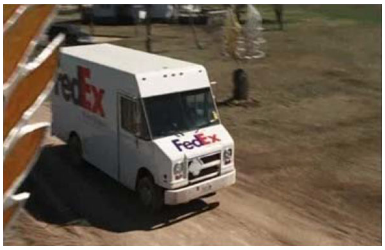


**Fig. 10** Watermarked key-frame

### 4.7 Color space re-transformation—step 7

Finally the watermarked key-frame will be obtained by transforming from YCbCr color space to RGB color space using Eq. 6 [14]. Figure 10 shows the final watermarked key-frame.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{pmatrix} \times \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} \tag{6}$$

All the watermarked key-frames along with the un-watermarked frames of the video sequence will be collaborated in the order of their appearance to form uncompressed watermarked video sequence.

## 5 Watermark extraction framework

At the extraction end, the watermarked key-frames will be extracted from the watermarked video sequence through the proposed “video bookmark” algorithm. The steps of proposed watermark extraction algorithm are described below:

Input: watermarked key-frame, secret key-1, secret key-2
Output: watermark

### 5.1 Color space transformation—step 1

The watermarked key-frame will be transformed from RGB color space to YCbCr color space using Eq. 3 and only Y part is taken for consideration. Figure 11a and b show the watermarked key-frame in RGB space and luminance part (Y) of watermarked key-frame, respectively.

### 5.2 DCT of luminance of watermarked key frame—step 2

Watermarked key-frame is divided into 8 × 8 non-overlapping blocks and forward DCT is performed on the Y

(a) (b)

**Fig. 11** **a** RGB of watermarked key-frame, **b** Y part of watermarked key-frame

part of watermarked key-frame for all such blocks using Eq. 4.

### 5.3 Decoding—step 3

Each block is examined thoroughly after applying Secret key-2 (obtained by repeating Secret key-1 for 16 times) that is used at the time of encoding. A particular block is considered to be watermarked if a particular low frequency coefficient (derived from Secret key-2) holds the frequency of scaled average of other two low frequency coefficients (derived from Secret key-2). The following example is provided to illustrate the decoding technique. The block (6, 6) is taken to examine it is watermarked or not. After performing FDCT the coefficients are as follows—

| − 483 | 295 | 87 | 30 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 161 | 154 | − 8 | − 24 | − 30 | − 16 | − 10 | − 5 |
| 26 | − 21 | − 46 | − 14 | 0 | − 5 | − 5 | − 5 |
| 11 | − 9 | − 13 | 8 | − 13 | − 15 | − 2 | − 2 |
| − 2 | − 20 | − 22 | − 8 | − 10 | − 2 | 7 | 3 |
| 1 | − 6 | − 1 | 5 | 2 | 5 | 3 | 0 |
| 1 | − 3 | − 6 | − 2 | 0 | 0 | 0 | 0 |
| 0 | − 2 | − 3 | − 2 | 1 | − 1 | − 2 | 0 |

Same assigned secret key value 2 that is used at the time of encoding is applied. According to our framework following calculation is performed—

For assigned value = 2,

$$(1,\,0) = \frac{(0,1)+(2,0)}{2} = \frac{295+26}{2} = 160.5 \pm \delta$$

where, ‘δ’ is a marginal threshold.

Here the block (6, 6) is considered as watermarked block because the scaled average of coefficients (0, 1) and (2, 0) (i.e., 160.5 + 0.5 = 161 where, δ =+0.5) is found at coefficient (1, 0).

### 5.4 Frequency substitution—step 4

The frequency of DC coefficient of a watermarked block is substituted by very high frequency (e.g. 2000) and the frequency of DC coefficient of an un-watermarked block is substituted with very low frequency (e.g. − 2000). DC coefficients are holding the most significant information of every DCT block and as binary watermark is considered in current context, substituted DC frequencies will be good enough to reconstruct the binary watermark. After high frequency substitution at DC coefficient, the block (6, 6) is as follows—

| 2000 | 295 | 87 | 30 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 161 | 154 | − 8 | − 24 | − 30 | − 16 | − 10 | − 5 |
| 26 | − 21 | − 46 | − 14 | 0 | − 5 | − 5 | − 5 |
| 11 | − 9 | − 13 | 8 | − 13 | − 15 | − 2 | − 2 |
| − 2 | − 20 | − 22 | − 8 | − 10 | − 2 | 7 | 3 |
| 1 | − 6 | − 1 | 5 | 2 | 5 | 3 | 0 |
| 1 | − 3 | − 6 | − 2 | 0 | 0 | 0 | 0 |
| 0 | − 2 | − 3 | − 2 | 1 | − 1 | − 2 | 0 |

After low frequency substitution at DC coefficient, an unwatermarked block say (1, 1) is as follows—

| − 2000 | − 51 | 16 | 3 | 0 | 0 | 1 | − 1 |
|---|---|---|---|---|---|---|---|
| − 151 | − 30 | − 17 | − 15 | − 1 | − 2 | − 4 | 2 |
| − 160 | − 58 | 46 | − 1 | 1 | − 5 | 3 | 0 |
| − 77 | 101 | − 35 | − 18 | − 7 | − 6 | − 1 | 3 |
| 20 | − 99 | − 50 | 25 | 1 | 6 | 2 | 3 |
| − 36 | − 14 | 46 | 18 | 10 | 4 | 2 | 0 |
| 12 | 9 | 16 | 17 | 0 | 1 | 0 | 0 |
| 3 | 14 | 24 | − 10 | − 4 | 0 | 1 | 1 |

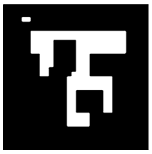


**Fig. 12** Extracted scrambled watermark

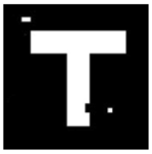


**Fig. 13** Descrambled watermark

### 5.5 IDCT—step 5

Inverse DCT is performed using Eq. 5 to obtain the scrambled watermark in spatial form. Figure 12 shows the extracted watermark in scrambled form.

### 5.6 Descrambling—step 6

Same Secret key-1 that is used at the time of scrambling the watermark is applied to descramble the extracted watermark to obtain it in its original form. Figure 13 shows the final extracted watermark after descrambling.

## 6 Resultset analysis

We have executed our "video bookmark" algorithm on thousand of video stream of different nature and variety of length because samples taken from different type of video will posses different characteristics, like—the first video sample "Cast Away" is a Hollywood movie that is of 16:9 aspect ratio and has 24 fps. In this video new scene arrives due to possible camera movements whereas the second sample taken is a popular cartoon "Tom and Jerry" that is of 4:3 aspect ratio and has 15 fps. In this sample mostly new scene arrives as cut to next scene rather than camera movements. Third sample is from a "Copa America" football match where the scene duration is long with rapid camera movements. The last sample is news from "NDTV" that has long scenes with minimum camera movements. Considering all these samples our algorithm is proven its excellence. Table 1 is provided in this regard where TNF and KF mean total no. of frames and key-frame, respectively.

A comparative study of our algorithm with some existing algorithm has also been made and for each algorithm some important characteristics are reported hereinafter at Table 2.

The leading features of proposed watermarking framework are stated below:

- Frequency domain watermarking offers more robustness than spatial domain.
- Using luminance component (Y) for watermark embedding makes the framework more robust because, human visual system is more sensitive to luminance than the chrominance and that's why most filter based attack does not involve the luminance as such. The luminance part is also not down-sampled in compression algorithms [19]. Therefore, the watermark can still be extracted in recognizable form even after performing filter based attacks or compression.
- Considering low frequency coefficients of individual DCT blocks of the luminance part (Y) of key-frames for embedding watermark information resist the effect of different compression techniques. Different compression algorithms discarded the high frequencies at the time of quantization and low frequencies are not modified as such because, they carry significant information. So, watermark information can be retrieved

**Table 1** Analysis of extraction of key-frames for different video type varying in length

| Video name | Video length (s) | Video type | Aspect ratio | fps | TNF | KF |
|---|---|---|---|---|---|---|
| Cast away | 35 | Movie | 16:9 | 24 | 840 | 9 |
| Tom and Jerry | 10 | Cartoon | 4:3 | 15 | 150 | 11 |
| Copa America | 25 | Sports (single take) | 4:3 | 24 | 600 | 3 |
| NDTV | 15 | News | 4:3 | 20 | 300 | 8 |

**Table 2** Comparative study of algorithms

| Characteristic | ATS [7] | FR [7] | PME [9] | Video bookmark |
|---|---|---|---|---|
| Automatic key frame selection | No | No | Yes | Yes |
| Variable no. of key frames | Yes | Yes | Yes | No |
| Requires motion vectors | No | No | Yes | No |
| Uses an optimization algorithm | No | Yes | No | No |
| Shot length sensitivity | Yes | Yes | Yes | No |

**Table 3** Analysis of PSNR

| Video name | Frame resolution | PSNR |
|---|---|---|
| Cast away | 720 × 405 | 73.82981 |
| Tom and Jerry | 640 × 480 | 70.66227 |
| Copa America | 640 × 480 | 71.42952 |
| NDTV | 640 × 480 | 70.81926 |

**Table 4** Analysis of nc

| Video name | Frame resolution | nc |
|---|---|---|
| Cast away | 720 × 405 | 0.94 |
| Tom and Jerry | 640 × 480 | 0.92 |
| Copa America | 640 × 480 | 0.91 |
| NDTV | 640 × 480 | 0.91 |

from low frequency coefficients even after higher level of compression.

- There is no fixed block (8 × 8 DCT block) for embedding watermark information. The watermark embedding blocks are identified depending on the watermark itself. That employs additional robustness to the framework against re-watermarking. It empowers the owner to extract the watermark if the watermarked key-frame is re-watermarked with a different logo by any counterfeiter.
- Security is well considered in the proposed framework. Two secret keys are used—the first one is used to scramble the watermark before embedding. And the second one is used to identify one of seven low frequency coefficients within an 8 × 8 DCT block for watermark encoding. Without knowing these two secret keys a counterfeiter will not be able to extract the watermark.
- The extraction algorithm is blind that means, neither the key-frames nor the watermark is required at decoding end.

The perceptual invisibility of watermark to HVS is established with PSNR. It is most commonly used as a measure of quality of watermarked content and could be defined via root mean square error (RMSE) as described in Eq. 7 [21, 22]. Detailed experiment has been carried out and we have achieved a set of nice PSNR values which are more than 70 dB. Hence no difference between host and watermarked key-frame can be noticed in bare eyes. Table 3 is provided in this regard.

$$PSNR = 20\log_{10}\left(\frac{MAX}{RMSE}\right) \quad (7)$$

If a pixel in the key-frame is defined as Y (i,j) and that in the watermarked key-frame is defined as y (i,j), then the root mean square error (RMSE) of the watermarked key-frame is computed with Eq. 8 [23] [24].

$$RMSE = \sqrt{\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[Y_{(i,j)} - y_{(i,j)}\right]^2}{M \times N}} \quad (8)$$

The quantitive similarity measurement between the referenced watermark and extracted watermark is computed by normalized correlation (nc). The nc calculation is done with Eq. 9 [25].

$$nc = \frac{\sum\sum(I_w[i][j] * I_o[i][j])}{\sqrt{\sum\sum(I_w[i][j] * I_o[i][j])^2}} \quad (9)$$

The proposed watermarking framework is being tested on a number of video frames with different binary watermarks. But it is observed that the watermark is sustained and extracted well. Some of the tested results are reported at Table 4.

Experimental analysis has also been carried out against different attacks on watermarked frames. A comparative study is given below with the proposed watermarking algorithm and the most efficient DCT based watermarking method of recent days proposed by Lin et al.

## 7 Conclusion

Detailed experiment has been carried out and found that the proposed "video bookmark" algorithm is able to extract key-frames efficiently from different type of videos varying in length and resolution by analyzing mean boundary luminosity of video frames. A DCT based invisible watermarking framework is also being proposed. The watermark is embedded in the low frequency coefficients of the luminance of key-frames. Embedded watermark can be extracted even after higher degree of compression and filter based attacks. Another leading aspect of the proposed framework is, the blocks selected in the key-frame for embedding watermark are function of the watermark itself. The watermark extraction framework is blind that guarantees except the pair of secret keys nothing are needed.

## References

1. Tonomura Y, Akutsu A, Otsugi K, Sadakata T (1993) VideoMAP and VideoSpaceIcon: tools for automatizing video content. In: Proc. ACM INTERCHI'93 Conference
2. Ueda H, Miyatake T, Yoshizawa S (1991) IMPACT: an interactive natural-motion-picture dedicated multimedia authoring system. In: Proc. ACM CHI'91 conference
3. Rui Y, Huang TS, Mehrotra S (1998) Exploring video structure beyond the shots. In: Proc. IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS)

4. Pentland A, Picard R, Davenport G, Haase K (1994) Video and image semantics: advanced tools for telecommunications. IEEE MultiMedia 1(2):73–75
5. Sun Z, Ping F (2004) Combination of color and object outline based method in video segmentation. In: Proc. SPIE Storage and Retrieval Methods and Applications for Multimedia
6. ZhaoL, Qi W, Li SZ, Yang SQ, Zhang HJ (2000) Key-frame extraction and shot retrieval using nearest feature line (NFL). In: Proc. ACM Int. Workshops on Multimedia InformationRetrieval
7. Hanjalic A, Lagendijk RL, Biemond J (1998) A new method for key frame based video content representation. In: Smeulders A, Jain R (eds) Image databases and multimedia search. World Scientific, Singapore pp 97–107
8. Hoon SH, Yoon K, Kweon I, (2000) A new technique for shot detection and key frames selection in histogram space. In: Proc. 12th workshop on image processing and image understanding
9. Narasimha R, Savakis A, Rao RM, De Queiroz R (2004) A neaural network approach to key frame extraction. In: Proc. of SPIE-IS&T electronic imaging storage and retrieval methodsand applications for multimedia
10. Liu TM, Zhang HJ, Qi FH (2003) A novel video key-frame-extraction algorithm based on perceived motion energy model. IEEE Trans Circuits Syst Video Technol 13:1006–1013
11. Ciocca G, Schettini R (2006) An innovative algorithm for key frame extraction in video summarization. J Real-Time Image Process 1(1):69–88
12. Cox IJ, Kilian J, Leighton FT, Shamoon T (1997) Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 6(12):1673–1687
13. Jun Y, Guo-hua C, Yi-jia Z (2006) A practical multipurpose color image watermarking algorithm for copyright protection and image authentication. In: Proc. digital telecommunications, ICDT '06, Digital Object Identifier, p 72–77 https://doi.org/10.1109/ICDT.2006.10
14. Lin SD, Shie S-C, Guo JY (2009) improving the robustness of DCT-Based image watermarking against JPEG compression. Elsevier. J Comp Stand Interfaces 32(1–2):54–60
15. Deb K, Al-Seraj MS, Hoque MM, Sarkar MI (2012) Combined DWT-DCT based digital image watermarking technique for copyright protection. In: Proc. 7th International conference on electrical and computer engineering, IEEE
16. Raval K, Zafar S (2013) Digital watermarking with copyright authentication for image communication. In: Proc. IEEE international conference on intelligent systems and signal processing (ISSP-13). p 111–116 (ISBN: 978-1-4799-0316-0)
17. Zong T, Xiang Y, Guo S, Rong Y (2016) Rank-based image watermarking method with high embedding capacity and robustness. IEEE Access 4:1689–1699
18. http://en.wikipedia.org/wiki/Frame_rate
19. Wallace GK (1991) The JPEG still picture compression standard. Commun ACM 34(4):30–44
20. Rao KR, Yip P (1990) Discrete cosine transform: algorithms, advantages, applications. Academic Press, Boston MA
21. Das S, Bandyopadhyay P, Paul S, Chaudhuri A, Banerjee M (2010) An invisible color watermarking framework for uncompressed video authentication. Int J Comput Appl (IJCA) 1(11):22–28
22. Dixit A, Dixit Rahul (2017) A review on digital image watermarking techniques. Int J Image Graph Signal Process 2017(4):56–66
23. Das S, Bandyopadhyay P, Banerjee M, Chaudhuri A (2011) A chip-based watermarking framework for color image authentication for secured communication. Communications in computer and information science, 1, Vol 125, Springer. Advances in Computing, Communication and Control, Part 2, Springer
24. Banerjee M (2007) Theory and application of cellular automata for authentication and watermarking. Ph.D. thesis, Jadavpur University
25. Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker Ton (2008) Digital watermarking and steganography, 2nd edn. Morgan Kaufmann, San Francisco. ISBN 978-0-12-372585-1
26. Das S, Banerjee M, Chaudhuri A (2017) An improved DCT based image watermarking robust against JPEG compression and other attacks. Int J Image Graph Signal Process 2017(9):40–50