WILEY

An educational tool for understanding MPEG-2 video compression and coding techniques

Unai Fernández-Plazaola | Gerardo Gómez ⓘ | F. Javier López-Martínez

Department of Ingeniería de Comunicaciones, ETSI Telecomunicación, Málaga, Spain

Correspondence
Unai Fernández-Plazaola, Department of Ingeniería de Comunicaciones, ETSI Telecomunicación, 29010 Málaga, Spain.
Email: unai@ic.uma.es

Abstract
The understanding of the video encoding/decoding processes is a key skill for students of computer science, engineering, or telecommunications, especially those specialized in multimedia. The special characteristics of the video signals, and how such features are leveraged to compress the video frames using coding theory, are hard to understand for engineering students. In this paper, we describe MPEG-2 analyzer, an educational tool designed to provide a comprehensive and interactive way to understand MPEG-2 video compression and encoding techniques. We have used our tool to evaluate its usefulness in reinforcing the knowledge about the MPEG-2 video (de)coding process to Telecommunication Engineering students. Results show that the evaluation of the tool is very positive, and it has helped them to understand many of the complex concepts covered in the course.

KEYWORDS
analyzer, coding, compression, MPEG-2, video

1 | INTRODUCTION

According to the latest Ericsson Mobility Report [5], video traffic currently accounts for 66 percent of all mobile data traffic, a share that is forecast to increase to 77% in 2026. In this context, the development of efficient video compression mechanisms that reduce the sizes of videos transmitted over the internet is key to meet stringent requirements such as minimal buffering and smooth playback, that ensure a certain quality of experience (QoE) to end users.

The well-known Motion Picture Expert Group (MPEG) standards have provided in the last 30 years well-established solutions for video and audio compression, coding, and distribution. While MPEG standards have largely evolved in this period, most of these improvements have been enabled

by the increased computing capabilities of coding/decoding devices thanks to Moore's law. In this sense, key features such as motion estimation and compensation, the use of temporal correlation between video frames, chroma sub-sampling, macroblock arrangements, or the use of discrete cosine transforms have been incorporated since the first releases in the early 90s. One of the most important MPEG standards is MPEG-2 (ISO/IEC 13818) [11], which focused on generic coding of moving pictures and associated audio information for digital television. MPEG-2 standard is the basis of video encoding techniques, which any student of computer science, engineering, or telecommunications should know in detail.

The study of multimedia transmission plays an important role in the curricula of a number of Engineering Degrees in the contexts of telecommunications,

electrical engineering, and computer science. Specifically, a strong emphasis is made on the systems of generation, transmission, processing, and recording of audio and video signals. These concepts are covered in different degrees throughout different courses such as *Fundamentals of Video*, *Audio and Video Systems*, or *Transmission of Multimedia Information*. All these courses are framed in the third year of the 4-year curricula of different Telecommunication Engineering degrees so that students have already completed several courses in mathematics, signal processing, communications and networking. Some of the learning outcomes defined in these subjects are: (i) To know, understand and handle the basic digital video and image processing tools necessary to understand the processes of compression; (ii) to analyze in a qualitative and quantitative way the digital video and image compression processes; and (iii) to identify the parameters that control their operation and establish its relationship with the objective and subjective impairments introduced by the compression process.

For some years, the faculty members in charge of these subjects at the School of Telecommunication Engineering in our institution have detected a certain difficulty for the students to successfully complete the learning process of this type of skills. This is due to a number of reasons, which relate to the complexity of understanding the impact of each of the algorithmic steps associated to video compression on how the video sequence is actually compressed.

The use of commercial video coding analyzers for practical activities is a plausible option to mitigate this issue. Some alternatives used for this purpose (including the authors themselves) have been VisualMPEG [17], Qualify [13], or Telestream [15]. However, commercial video analyzers often have nonintuitive user interface for students and, in addition, require the purchase of licensed versions, which limits their use in public institutions. Another alternative is the development of educational tools with similar functionality as commercial ones, but which can be used freely by students throughout their career. This was suggested in [3], although with a different focus: it was the bulk of students following the course who had to take care of the implementation of the MPEG-2-based video coding analyzer, instead of using the developed tool for learning purposes.

To the best of the author's knowledge, there is no free tool devoted to video coding for learning purposes. For these reasons, we decided to design and develop *MPEG-2 Analyzer*, an educational tool that can help the students understand each and every one of the details of the video encoding and decoding process in the MPEG-2 standard.

A key objective when designing this tool is to provide a simple Graphical User Interface, that allows the students to easily interact with the video content to understand how the encoded information is structured.

The remainder of this paper is organized as follows. In Section 2, some background and related work related with MPEG-2 technology is briefly summarized, with a special focus on the video compression fundamentals. The MPEG-2 Analyzer tool is described in Section 3, and compared with other commercial video coding analyzers. A use case description is then detailed in Section 4. Analysis and discussion from the students' feedback are provided in Section 5. Finally, the key conclusions are presented in Section 6.

2 | PRELIMINARIES

2.1 | Background and related work

The development of educational software tools in the context of engineering education is a common approach to improving the students' learning skills when covering complicated subjects in the coursework.

This is the case, for instance, when teaching telecommunication standards [12, 8], and multimedia teaching is no exception [2]. Some examples of educational tools for engineering education (in general) are described next. In Gerson et al. [7], new multimedia software and a workbook for the improvement of 3-D spatial *visualization skills* for engineering students is described. In Tepe and Savaştaer [16], an expansion board for Raspberry Pi is presented as a tool for the acquisition, processing, and displaying of biomedical signals. A new educational simulation tool designed for the generic study of wireless networks is presented in Sanguino et al. [14]. In Dogmus et al. [4], an interactive educational tool for artificial intelligence planning for robotics is described. In Akgul et al. [1], a novel interface was designed for chaos-based encryption to be used as an educational tool in cryptology. A software tool (SEFRI) to improve the learning process of the representation and compression of multimedia *images* in shown in Fresnedo et al. [6].

Most previous tools have been used by students of computer science, engineering, or telecommunications, and evaluated with quantitative student surveys. However, existing commercial video coding tools [17, 13, 15] require the purchase of licenses for their operation. Hence, this makes it difficult that students can use them in their personal computers, and beyond the context of the multimedia course. Besides, they are often not easy to handle for non-experienced users. For these reasons, in

this article, we present a new educational and free tool called *MPEG-2 Analyzer*, which has been used and evaluated for engineering students to help them understand easily the video coding and decoding processes. This tool is later described in Section 3, and and compared to other existing commercial video coding analyzers.

2.2 | MPEG-2 technology overview

MPEG-2 was specified in ISO/IEC 13818 as a standard for compressing video and audio bit streams [11]. MPEG-2 supports a wide variety of coding rates and formats, although it is mainly targeted for high-quality video traffic at bit rates greater than 2 Mbps. A video stream is composed of a succession of consecutive video frames; however, it is not feasible to compress such video frames as individual images, since this would require an unaffordable throughput. Next, some details about the structure of video streams as well as video compression fundamentals, are given.

MPEG-2 video streams are hierarchically structured into the following elements (see Figure 1):

- **Block**: A basic image unit formed by 8×8 pixels.
- **Macroblock**: A group of 16×16 pixels that contains four luminance blocks (i.e., the bright information of the image) and a variable number of chrominance blocks (i.e., the color information of the image). Five types of macroblocks are defined: 1) *intra-coding*, with no motion compensation; 2) *forward prediction*, for which a previous reference image is used for motion compensation; 3) *backward prediction*, for which a future reference image is used for motion compensation; 4) *bidirectional prediction*, for which *both* a previous and a future image are used for motion

compensation; 5) *skipped macroblock*, that is, a macroblock for which no information is sent to the decoder.
- **Slice**: Defined as a consecutive and ordered number of macroblocks. Each slice is independently encoded with specific coding parameters.
- **Picture**: A sequence of slices that represents either a complete frame picture or one of the two field pictures used in interlaced mode. MPEG-2 defines three types of pictures: 1) stand-alone pictures, referred to as intra-coded or I pictures, 2) Predicted pictures (or P pictures), which use forward prediction from previous reference pictures (I or P), and 3) Bidirectional pictures (or B pictures), which combine forward and backward prediction from past or future references (I or P) to encode the information.
- **Group of Pictures (GoP)**: It includes a set of I/P/B pictures. All GoPs in MPEG-2 start with an I picture, which is independently encoded and decoded. For example, let us consider a typical I:P:B picture ratio in a GoP of 1:3:6 and a picture pattern following the logical sequence $I_1 B_2 B_3 P_4 B_5 B_6 P_7 B_8 B_9 P_{10}$, being the subscripts the positions of the video frames in the original video sequence. Taking into account that B pictures require future (I or P) pictures to be encoded, the encoder must reorder the pictures before coding (to simplify the decoding process) to produce the following pattern: $I_1 P_4 B_2 B_3 P_7 B_5 B_6 P_{10} B_8 B_9$.
- **Video sequence**: It includes a sequence of GoPs, which may be associated to a progressive or interleaved image format.

During the video compression process, video blocks are translated into a transformed domain so that compression quality may be better adjusted. To do that, the processing chains described in Figure 2 are applied for compression and decompression, where DCT denotes Discrete Cosine Transform. At the decoder side, the inverse functions are required to recover the video content, that is, temporal and spatial decompression via an inverse DCT (IDCT) and inverse quantization (Q^{-1}), as well as chroma upsampling. Besides, since the key compression benefits are obtained when leveraging spatial and temporal correlation, the spatial and temporal compression processes also need to be well understood. The key processes applied over the video stream are briefly explained next:

1. **Chroma subsampling:** Red, green and blue (RGB) intensity information associated with each pixel are translated into luminance (Y) and color vectors (C_B, C_R), so that they can be treated separately and incorporate the human perception features previously described. Originally, chrominance and luminance
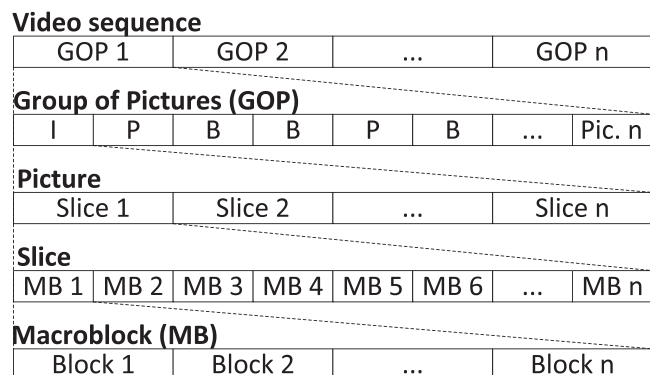
Video sequence

GOP 1	GOP 2	...	GOP n

Group of Pictures (GOP)

I	P	B	B	P	B	...	Pic. n

Picture

Slice 1	Slice 2	...	Slice n

Slice

MB 1	MB 2	MB 3	MB 4	MB 5	MB 6	...	MB n

Macroblock (MB)

Block 1	Block 2	...	Block n

FIGURE 1 Hierarchical structure of MPEG-2 elementary video streams.
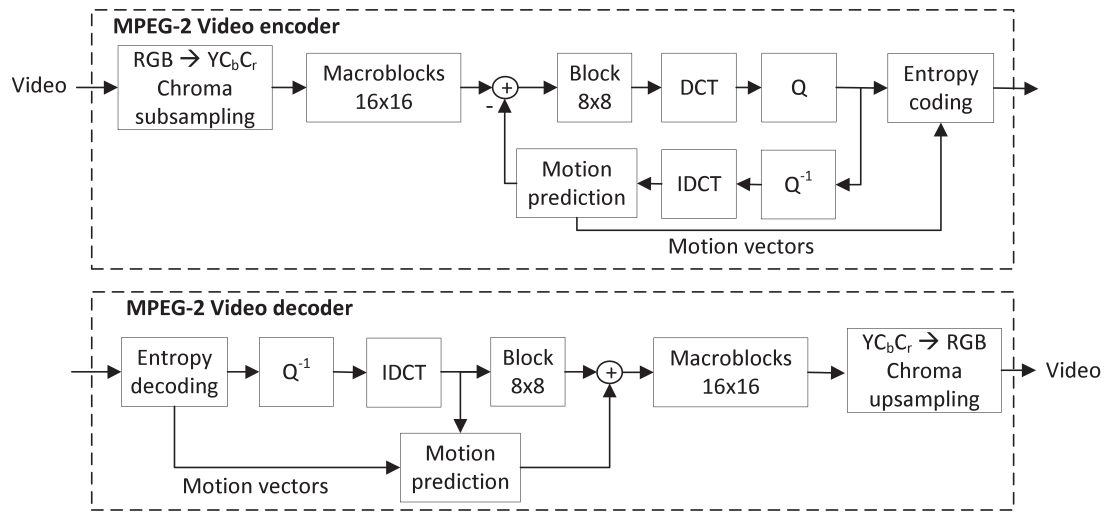
FIGURE 2 MPEG-2 video encoder/decoder architecture.

information are sampled at the same resolution, which is usually denoted as 4:4:4. Depending on how many chrominance blocks are dropped to improve compression performance, different subsampling formats can be used: 4:2:2 format uses half resolution in the horizontal direction for the chrominance, and 4:2:0 format uses half resolution both in the horizontal and vertical directions.

2. **Spatial compression:** A bi-dimensional DCT is applied to each block (8×8 pixels) to convert it from the spatial domain to the frequency domain. Mathematically, this is accomplished through the following expression:

$$F(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (1)$$

where x and y represent the horizontal and vertical coordinates or original samples within a block, and u and v represents the coordinates in the transformed domain; C_u and C_v are constants of value $1/\sqrt{2}$ if u or v are null; otherwise $C_u = C_v = 1$. The output of this DCT results in a series of coefficients indicating the magnitude of cosine functions at increasing frequencies. Specifically, the resulting matrix $\mathbf{F}$ has 64 coefficients, so that element $\mathbf{F}_{1,1}$ represents the DC component, whereas the other coefficients represent higher frequency components (AC) in the horizontal and vertical axis, being $\mathbf{F}_{8,8}$ the highest frequency component in both directions. It is important to note that for most images, and thanks to the inherent spatial correlation of consecutive pixels within a block, the DCT operation allows the key features of

each block are captured by a small number of coefficients, while the remainder coefficients (especially those of higher frequencies) have small magnitudes. Hence, quantization (denoted as Q) can be used to drop these small-magnitude coefficients, thus reducing the amount of coded information using an entropy-based compression method. Interestingly, quantization can be designed individually for I/P/B frame types; intuitively, it is recommended that strong quantization is avoided in frames that can act as references. Similarly, different quantization scales may also be defined at a macroblock level. Once the DCT coefficients are quantized, they need to be encoded. First, coefficients are arranged according to a zigzag pattern specified in the standard, after the last non-zero coefficient is reached. By doing so, the remainder zero coefficients are dropped before entropy compression, which reduces the overall bitrate. Finally, entropy coding is applied using Huffman encoding and taking into account the existence of long runs of zero coefficients.

3. **Temporal compression:** Inter-frame compression takes advantage of the correlation between consecutive video frames along time. In this sense, MPEG-2 uses three types of frames (or pictures) as indicated before: I-frames, P-frames, and B-frames. The key idea is to encode only those macroblocks which are new compared to the references, as well as motion vector information for those macroblocks associated to moving objects in the video sequence. Hence, we avoid to encode again those macroblocks that remain unaltered. Note that I frames cannot take advantage of temporal compression, since they are encoded as standalone pictures by definition. During this process, it is required to estimate both the motion vector associated to the macroblock under consideration, as

well as the differences with respect to the reference macroblock, which is referred to as prediction error. Depending on whether a such prediction is done from previous or posterior frames, it is called backward or forward prediction, respectively. Bi-dimensional DCT transformation is then applied to the prediction error matrices, which are quantized and coded as if they were regular macroblocks. However, such error matrices can be much more compressed than the original macroblock, which allows to further reduce the coding bitrate. In case the search process does not provide any matching with any macroblock in the reference picture (i.e., new objects in the video scene), the macroblock is encoded as intra-coded.

3 | MPEG-2 ANALYZER TOOL

3.1 | Design and implementation

The first version of the MPEG-2 Analyzer software was developed in 2013 by the student Isaac Lozano Rey as his B. Sc. thesis in Telecommunication Engineering (Sound and Image), supervised by one of the authors of this paper. The current version of the application is the result of several improvements over the such original version, and it is freely available upon request at the analyzer web page [9]. During these years, MPEG-2 analyzer has been used by numerous Universities and institutions, such as: Dept. Teoria de la Señal y Comunicaciones (Universidad de Vigo), Dept. Teoria de la Señal y Comunicaciones (Universidad Politecnica de Barcelona), Dept. Comunicaciones (Universidad Politécnica de Valencia), Dept. Teoria de la Señal y Comunicaciones (Universidad Carlos III), Dept. Ingeniería Eléctrica y Electrónica (Universidad Pública de Navarra), Servicio Nacional de Aprendizaje (Medellín, Colombia), as well as by students from Escuela Politécnica Nacional (Ecuador).

The MPEG-2 Analyzer has been developed using a reference implementation provided by the MPEG Software Simulation Group (MSSG). Therefore, the different algorithms implementing the MPEG standard are those provided by the MSSG. The implementation of the MPEG Analyzer has consisted of the design of the graphical user interface (GUI) and the interface between the routines provided by MSSG and the GUI. For the latter interface, it has been necessary to choose a data structure containing all the parameters to be presented in the GUI.

The MSSG implementation was originally developed in C and was made available with all the source code for academic purposes in reference [10]. The MSSG software basically consists of two command-line applications without graphical interface: *mpeg2enc*, which compresses
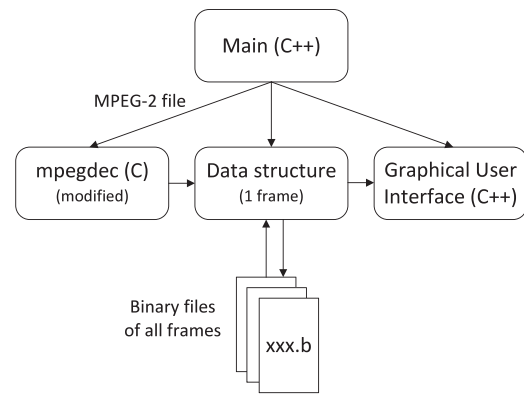

FIGURE 3 Operation diagram of MPEG-2 analyzer.

video frames into MPEG-2, and *mpeg2dec*, which makes the inverse operations. It is the latter, *mpeg2dec*, which was used to develop the MPEG-2 Analyzer tool.

The main routines of *mpeg2dec* application have been modified to obtain all the relevant information from a MPEG-2 video stream, and put together with a graphical interface to develop the MPEG-2 Analyzer tool in C++. Figure 3 represents how MPEG-2 Analyzer is designed to use the information obtained from *mpeg2dec* and present it in a friendly graphical interface.

A data structure was designed to save all MPEG-2 parameters for one video frame. The source code of *mpeg2dec* was modified to allow storing the different parameters of the frame as they were decoded on that data structure. Once a frame is completely decoded, its data structure is stored in several binary files. The *prediction error* or *decoded image* (i.e., part of the video frame not available in the reference), *predicted image* (i.e., part of the video frame available in the reference), and *reconstructed image* (i.e., decoded image + predicted image) are stored separately for the luminance and chrominance components.

An additional file contains information on all macroblocks and blocks of the frame, including motion vectors, block coefficients, and the rest of the information needed in the application. Once the entire MPEG-2 video is successfully decoded, all the relevant files generated for each frame are stored in the disc. Then, the Analyzer can retrieve the information of any frames of the video, by loading the corresponding files to the data structure and showing the MPEG-2 information in the graphical interface.

3.2 | GUI description

Since the ultimate purpose of developing the MPEG-2 Analyzer tool is educational, a Graphical User Interface

(GUI) has been implemented. As it will be later covered in detail in Section 3.3, the key design choices for the GUI have been made in coherence with those of other available commercial tools previously used in the multimedia courses. With this in mind, the GUI is organized into 8 panels, as shown in Figure 4.

1. **General information**: It contains the most important video parameters (described in Table 1).
2. **Image selection list**: It contains a numbered list of all pictures/frames within the video. For each frame, the following information is given: display order, coding order, image type (I, P, B), storage size (in bits). The user may click on a particular image to visualize it in the display and analyze its (macro)blocks. The table also presents the order used for coding the video sequence, which is usually different from the display order. Note that reference frames such as I, but also P, must be encoded before B frames so that they can be used as references.
3. **Image type selection**: It allows the user to select how the image is displayed: i) Y (only luminance component is shown); ii) C_b (only blue-difference chrominance component is shown in grayscale); iii) C_r (only red-difference chrominance component is illustrated in

TABLE 1 List of parameters and description included in the *General Information* panel of the GUI.

Parameter	Description
Number of frames	Total number of frames in the video sequence
Aspect ratio	Ratio of the video width to its height (x:y)
Luma	Number of pixels (width x height) for luminance information
Chroma	Number of pixels (width x height) for chrominance information
Mb	Number of macroblocks (width x height) in a picture
Bits rate	Average number of bits per seconds of the video sequence
Frame rate	Number of frames per second
Chroma format	YC_bC_r format {4:2:0, 4:2:2, 4:4:4}
Video format	Video format standard {NTSC, PAL, SECAM}
Intra	Average number of bits for I frames
Pred	Average number of bits for P frames
Bidi	Average number of bits for B frames

Abbreviation: GUI, graphical user interface.
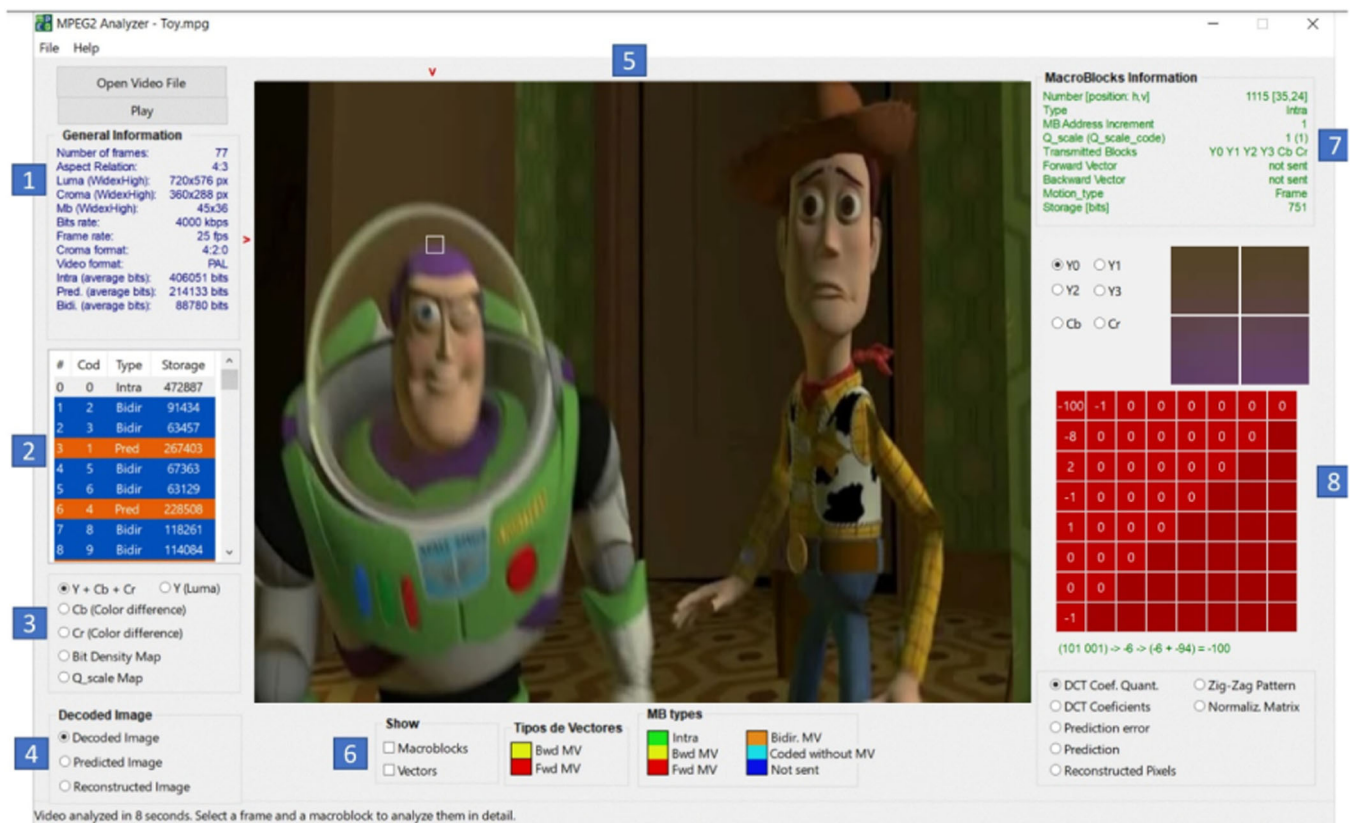
FIGURE 4 MPEG-2 analyzer screenshot.

grayscale); iv) bits density map, which represents the number of bits (in grayscale) per macroblock; v) Q_{scale} map, indicating the Q_{scale} value used for quantization of each macroblock (in grayscale).

4. **Decoded image type selection**: It allows the user to visualize different image types (decoded, predicted and reconstructed). Note that I frames will not provide any predicted information, as they do not depend on any other frames.

5. **Image/video display**: It shows the image/video content, allowing the user to select a particular macroblock to be analyzed.

6. **Macroblock visualization and motion vectors**: It allows the user to visualize motion vectors and the macroblock type in different colors.

7. **Macroblock information**: It contains the most important parameters of the selected macroblock (described in Table 2).

8. **Block information and DCT coefficients**: It includes relevant block information like DCT coefficients (before an after quantization), prediction error (i.e., difference with respect to the reference macroblock, if there's one), reconstructed pixels and zig-zag pattern. Figure 5 shows an example. Additionally, it details all the steps for obtaining the selected DCT coefficient value from the binary code used for its encoding. In the case of the AC coefficients, the run length of preceding zero-coefficients and the value of the coefficient are given. In the case of the DC coefficient, the following information is presented: category bits, additional bits, decoded value, and final result obtained by adding the DC value of the previous block, as requested by MPEG-2 standard. In the example of Figure 5, the DC coefficient has the binary code 101 corresponding to category 5 and the
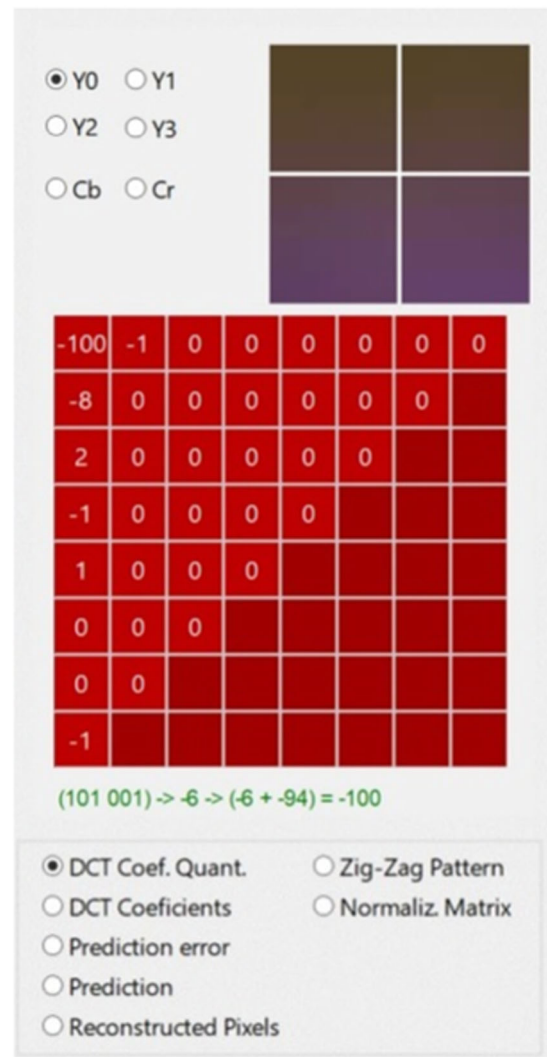

FIGURE 5 Block information and DCT coefficients. The formula shows an explanation of how to compute the selected coefficient. DCT, Discrete Cosine Transform.

TABLE 2 List of parameters and description included in the *Macroblock Information* panel of the GUI.

Parameter	Description
Number [position: h,v]	Macroblock number (in coding order) with [h,v] its horizontal and vertical position
Type	Intra (no motion compensation), Bwd MV (backward motion vectors), Fwd MV (forward motion vectors), Bidir MV (bidirectional motion vectors), Coded without MV, Not sent (skipped)
Macroblock address increment	Needed to know the position of the macroblock
$Q_{scale}(Q_{scale\_code})$	Quantization scale value (quantization scale code). Larger Q_{scale} denotes stronger quantization
Transmitted blocks	Indicates the list of block that transmit non-null DCT coefficients
Forward vector	Motion compensation vector from a previous image
Backward vector	Motion compensation vector from a subsequent image
Motion type	Type of motion prediction
Storage	Macroblock size (bits)

Abbreviation: GUI, graphical user interface.

additional bits 001, which give the decoded value −6. As the value of the previous block DC coefficient is −94, the value finally obtained is −100. The last zero-coefficients after the last non-zero coefficient (row 8, column 1 in the example) are discarded (empty cells in the dark red color).

One of the key aspects of the MPEG-2 Analyzer tool is its simplicity to interact with the video content, allowing to select a particular macroblock of a specific image by just clicking on the image/video display. Once a particular image and macroblock is selected, the user may analyze its DCT coefficients, prediction error, motion vectors, etc.

3.3 | Commercial video coding analyzers

To better assess the performance of our MPEG-2 Analyzer, a comparison with the some state-of-the-art commercial video coding analyzers, VisualMPEG and Qualify, is presented in this section. Specifically, VisualMPEG was the original SW tool used in the context of our multimedia-related teaching, until the MPEG-2 Analyzer tool was developed and used ever since. Later, VisualMPEG was discontinued, and Qualify became the most similar alternative in the market.

Figure 6 shows the VisualMPEG GUI for the same video and the same display selection as used in Figure 4 for the MPEG-2 Analyzer. Figure 7 shows the Qualify GUI also for the same video and the same display selection. The authors are not aware of or do not have access to other commercial MPEG-2 video analyzer applications with similar features.

We consider that the key differences between our analyzer and the two commercial solutions can be summarized in the following three items, which were our motivation to develop the MPEG-2 Analyzer.

First, commercial software requires the purchase of licenses for its use. When the authors used the VisualMPEG tool, they purchased a license valid for 15
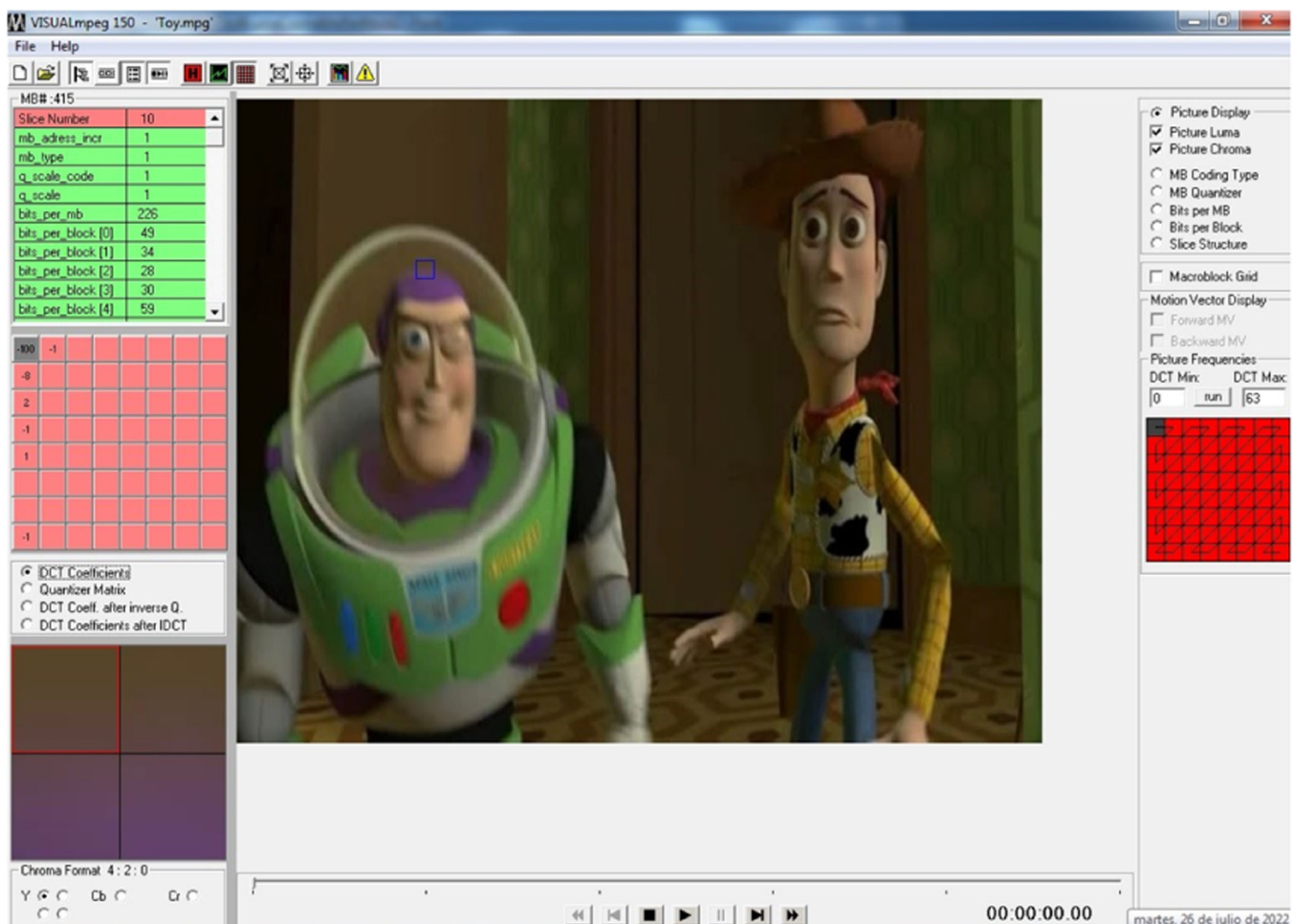
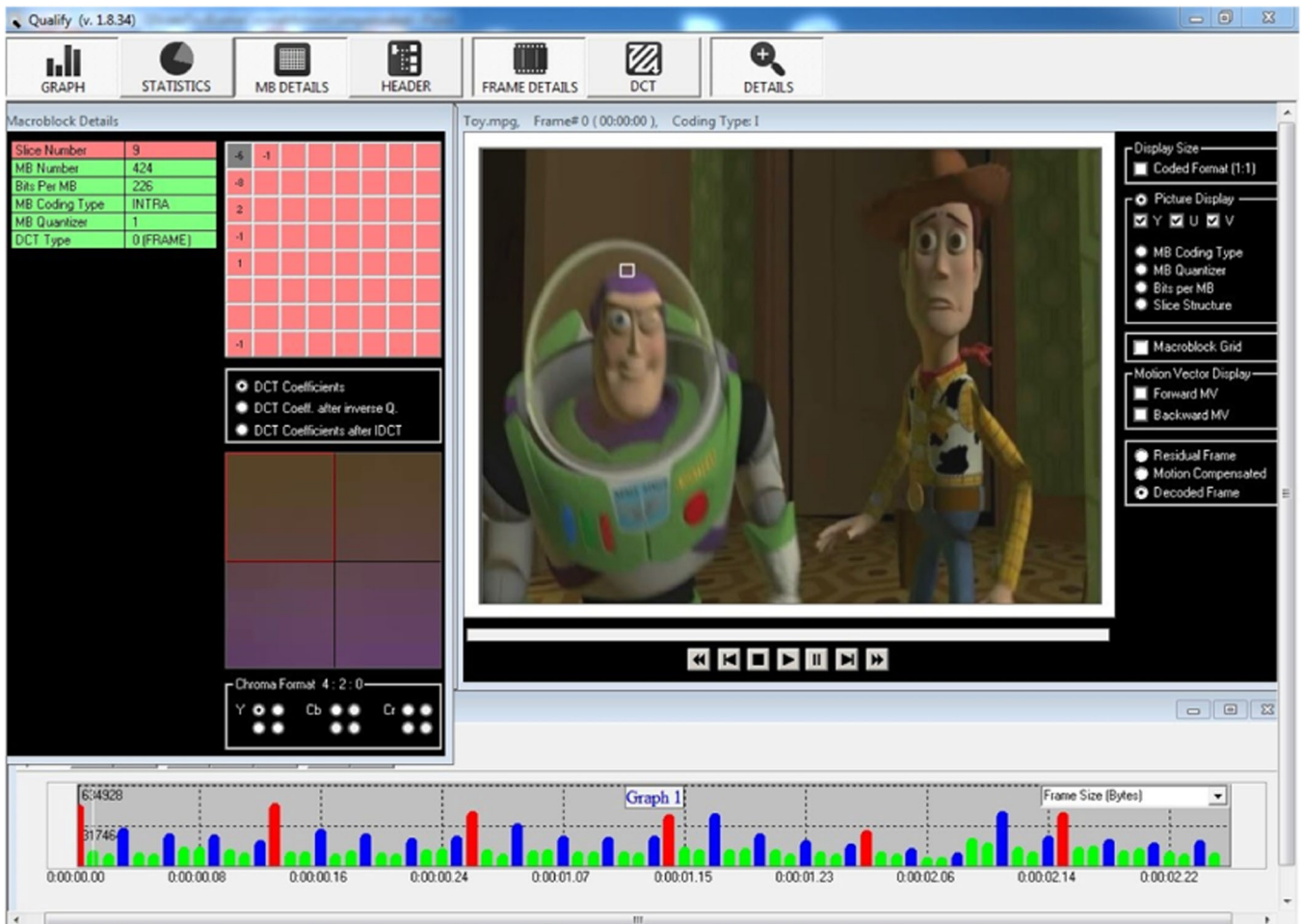

FIGURE 6 VisualMPEG screenshot.

FIGURE 7 Qualify screenshot.

users. This license was installed on a server in the university laboratories, allowing the computers connected to the same network to run the program. Hence, only a reduced number of students could concurrently use the software, requiring instructors to split practical sessions into different slots. The license can not be used by students at home. This lack of flexibility largely limited the learning process of students, preventing them to consolidate the concepts studied in the laboratory at home in a more relaxed way.

Second, we find the MPEG-2 Analyzer GUI easier to use than the other two applications GUIs. All video information can be viewed on a single screen, without having to navigate between different menu options. For example, in VisualMPEG, as can be seen in Figure 6, to see the DCT coefficients of a block, it is necessary to press a button in the tools menu. Something similar happens with Qualify, see Figure 7, which requires pressing the MB details button. In the case of the MPEG-2 Analyzer, the block information is always displayed, and each time a different macroblock is clicked in the image, the

displayed values of the block are updated. This makes it easier for the students to carry out the practical exercises.

Third, the MPEG-2 Analyzer adds more information about the image blocks. In addition, it provides all the details of the decoding process for DCT coefficients, which neither VisualMPEG nor Qualify offer. As can be seen in Figures 5 and 8, the MPEG-2 Analyzer is able to also show the predicted pixels and the reconstructed pixels, in addition to the information about the quantized/non-quantized and inverse DCT coefficients (the block prediction error). On the other hand, by selecting a DCT coefficient from the block with the cursor, all the details for obtaining it are presented below, as described in Section 3.2. Note that in VisualMPEG the value presented for the DC coefficient is −100, the same as in MPEG-2 Analyzer. However, in Qualify, the value −6 is presented instead, which is the one *before* adding the DC value of the previous block. These inconsistencies can lead to misunderstandings among the students and are resolved when the details to obtain the values are shown. It is very instructive for the students to see how
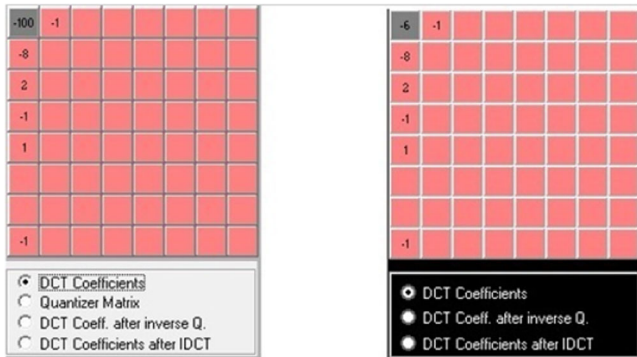
FIGURE 8 Block information and DCT coefficients detail for VisualMPEG on the left and Qualify on the right. DCT, Discrete Cosine Transform.

the coefficients are coded, and all the steps to obtain the pixel values.

FIGURE 9 Motion vectors (forward vectors in red, backward vectors in yellow).

4 | USE CASE DESCRIPTION

Our MPEG-2 analyzer tool is used in several Third-year courses within the B.Sc. in Telecommunication Engineering curricula: *Audio and Video Systems*, *Fundamentals of Video* and *Transmission of Multimedia Information*. One of the practical activities or laboratories of these subjects is focused on the detailed analysis of a video segment using MPEG-2 Analyzer tool. The reference video file used in the activity is called *Toy.mpeg*, composed of 77 frames (see Figure 4).

The activity is guided, intending that students put into practice the knowledge acquired in class. For this purpose, the activity is structured in three main tasks: i) Basic handling of the analyzer; ii) Analysis of I-frames; iii) Analysis of P/B frames. The entire set of activities, which are carried out in the 3-h session, are detailed as follows:

1. **Basic handling of the analyzer:**
   a. *Goal*: Become familiar with the *General Information* panel of the GUI (Panel 1 in Figure 4) and review the concept of dependency between different types of frames within of GoP. *Instructions*: Looking at the *General Information* panel and the table of frame types, determine the length of the GoP. For the first GoP, draw two schematics indicating the dependencies between frames: the first schematic representing the frames ordered as they are compressed, and the second one with the frames ordered as they are stored/displayed.
   b. *Goal*: Become familiar with the *Image Selection* list to select a particular frame, and with the *Image*

Type selector to choose how the image is displayed (Panels 2 and 3 in Figure 4). Review the concept of quantization value for different types of frames. *Instructions*: Focusing on the Bit Density and Q_{scale} maps, select the first 6 images of the video and explain how the bit density and Q_{scale} value change with the frame type (I, P, B).
   c. *Goal*: Become familiar with the *Decoded Image Type* selector (Panel 4 in Figure 4) and relate the concept of prediction error with the different type of frames (I, P, B). *Instructions*: Check the predicted image and the prediction error obtained for different types of frames (I, P, B). Justify for what kind of images the prediction is better, and why in some areas the prediction fails.
   d. *Goal*: Become familiar with the *Macroblock visualization and motion vector* selector (Panel 6 in Figure 4) and identify each macroblock type with those explained in theory. *Instructions*: Select different types of frames (I, P, B) and indicates the type of macroblock.
   e. *Goal*: Review the concept of motion vector, and associate this concept with the different type of frames. *Instructions*: Check and justify the different motion vectors that appear in the different types frames (I, P, or B). Explain why there are long motion vectors in some areas (see Figure 9).
   f. *Goal*: Become familiar with the *Macroblock information* and *Block information and DCT coefficients* (Panels 7 and 8 in Figure 4). *Instructions*: As the video uses a 4:2:0 format, there are four blocks of luminance and two blocks of color difference. Select a specific block to be analyzed. Select

different macroblocks from different images and check the values obtained from the quantized DCT coefficients, de-normalized DCT coefficients, prediction error, prediction, reconstructed pixels, zigzag pattern and normalization matrix. In the case of DCT coefficients (quantized and not), you can select any of the coefficients, and the application indicates below the matrix (in green) how the corresponding value is obtained (see Figures 10 and 5).

2. **Analysis of I frames:**

a. *Goal*: Understand how the DC component of a macroblock is encoded. *Instructions*: Find a complex macroblock of any of the I frames and examine the quantized and de-normalized DCT coefficients for each of its blocks. Select a luminance block and its quantized DC coefficient (i.e., the one in [1,1] position]). Describe how the bit sequence is decoded into a specific number. Repeat the explanation with the same coefficient, now using its de-normalized version.

b. *Goal*: Understand how the rest of the coefficients at the DCT output for a macroblock are encoded. *Instructions*: Select a quantized DCT output coefficient that is preceded by zero coefficients and watch how it is decoded. Check that the sign bit and the number of zero coefficients preceding it match the decoded values. Select the same coefficient, but now in its de-normalized version, and notice that the formula presented to obtain its value is the one studied in theory. Find each value of the formula and check that the final result is correct.
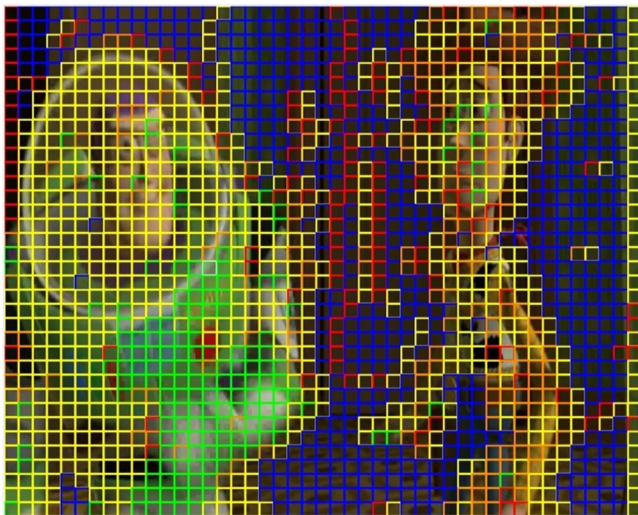

FIGURE 10 Macroblock type information: Intra (green), Bwd MV (yellow), Fwd MV (red), Bidir MV (orange), Not sent (blue). See Table 2 for further information.

c. *Goal*: Understand how the prediction error can be obtained from the DCT coefficients of a macroblock. *Instructions*: Look for a macroblock of reduced complexity (i.e., with only 2 or 3 nonzero DCT coefficients) of any of the I frames. Select one of the luminance blocks and examine its denormalized DCT coefficients, prediction error and reconstructed pixels. Explain how can we obtain the prediction error from the DCT coefficients.

d. *Goal*: Understand the differences in the DCT coefficients for different types of image transitions (horizontal vs. vertical). *Instructions*: Locate, examine, and compare the DCT coefficients of a block in which there is an abrupt horizontal transition (an edge) with those of another block with an abrupt vertical transition. Justify how the DCT coefficients are arranged in the matrix depending on the direction of the transition.

e. *Goal*: Understand the relationship between the DC value and the image brightness level. *Instructions*: Examine the DC component obtained for luminance in blocks of reduced spatial complexity (almost homogeneous if possible) with low, medium and high levels of brightness, respectively. Explain the relation between these DC values and the brightness levels of the blocks.

f. *Goal*: Understand the relation between the DC value and the image colors. *Instructions*: Examine the DCT coefficients of the color difference signals for blocks with homogeneous colors but with different tones. Explain the reason why a block has only one DC component and what is the relation between the DC component value and the image colors.

3. **Analysis of P and B frames:**

a. *Goal*: Understand the different compression ratio for I, P and B frame types. *Instructions*: Write down the average number of bits used for each frame type (I, P, B) of the video sequence. Calculate the compression factor obtained by using motion prediction and differential coding techniques (P_{avg}/I_{avg} and B_{avg}/I_{avg}). Explain why it is possible to compress more with some types of frames than with others.

b. *Goal*: Understand why different types of macroblocks are used in different areas of the image. *Instructions*: Select the image number 3 and look at the predicted image. Look at the type of macroblocks in each area of the image (characters, background...) and explain why different types of macroblocks appear in different areas.

c. *Goal*: Understand the relation between the type of prediction macroblock and its DCT coefficients.

Instructions: Select a B frame and find macroblocks with forward, backward and bidirectional prediction macroblocks. For each type, look at their DCT coefficients that correspond to their blocks. Explain why some macroblocks contain more zero coefficients than others.

d. *Goal*: Understand the use of motion vectors for prediction. *Instructions*: Select an image with long motion vectors and explain why they are so long.

5 | VALIDATION RESULTS

This section presents the experiment and results of several tests applied to groups of students enrolled in the courses *Fundamentals of Video* and *Transmission of Multimedia Information*. The goal is to determine to what extent the use of MPEG-2 Analyzer tool is useful to understand the key concepts related to MPEG-2 compression.

5.1 | Validation

The selected sample for the experiment was a set of 5th-semester students enrolled in B.Sc. in Telecommunications Engineering at Universidad de Málaga, Spain. Since the practical activity using MPEG-2 Analyzer tool is mandatory, all students in the class participated in the experiment, so there is no control group in this work.

We have carried out two types of validation of the developed tool. On the one hand, we aim to determine the usefulness MPEG-2 Analyzer for the purpose of learning multimedia-related concepts, based on the students' own perception. On the other hand, we aim to evaluate the success of the transition between commercial SW tools (i.e., VisualMPEG) and MPEG-2 Analyzer, in the context of our multimedia-related courses.

Regarding the former, the MPEG-2 analyzer tool was used by a group of students to complete the activity described in Section 4. After completing this task, a test was applied to measure the level of understanding provided by the tool. The response scale was 5-point Likert type, where 1 is totally in disagreement, and 5 is totally in agreement. The questions of the test are detailed in Table 3, which are divided into two types of questions: 1) Q2–Q10 are focused on the utility of the tool to help understand technical aspects of the MPEG (de) coding process, and 2) Q1 and Q11–Q15 collect students' feedback about their satisfaction with the use of the tool.

A total of 43 students, 11 females and 32 males, aged 20–25, have participated in this activity. We have noted that age or gender does not influence the results, so the results below do not specify these aspects. None of the students had previous experience with any similar tool for video analysis.

TABLE 3 List of 15 questions included in the student's survey.

#	Question
Q1	The part of the subject associated with the MPEG-2 analyzer is interesting.
Q2	MPEG-2 analyzer has helped you understand the differences between the types of images (I, B, P).
Q3	MPEG-2 analyzer has helped you improve your understanding of forward and reverse image prediction.
Q4	MPEG-2 analyzer has helped you improve your understanding of the use of motion vectors for prediction.
Q5	MPEG-2 analyzer has helped you understand the differences between the types of images (I, B, P).
Q6	MPEG-2 analyzer has helped you understand the different types of macroblocks that exist.
Q7	MPEG-2 analyzer has helped you understand the relationship between the block image and its DCT.
Q8	MPEG-2 analyzer has helped you to understand the effect of the quantization of the DCT coefficients.
Q9	MPEG-2 analyzer has helped you better understand the encoding of series of null coefficients.
Q10	MPEG-2 analyzer has improved your understanding of video compression and encoding techniques.
Q11	The user interface of the MPEG-2 analyzer is attractive.
Q12	The user interface of the MPEG-2 analyzer is easy to use.
Q13	The exercises you have done with the MPEG-2 analyzer have increased your interest in video encoding.
Q14	Your overall evaluation of the part of the subject associated with the MPEG-2 analyzer is positive.
Q15	My evaluation of the tool is positive.

With respect to the latter validation previously described, we provide a comparison between MPEG-2 Analyzer and VisualMPEG based on students' exam marks. They correspond to the course Fundamentals of Video. At the end of practical laboratory activities, the concepts acquired by the students are evaluated with an exam. The last year in which VisualMPEG was used in the course of Fundamentals of Video was 2012/2013. We show the exam marks obtained by the students in 2013/ 2014 and compare them with those obtained in the following year, that is, the first time using MPEG-2 Analyzer. We note that it was not possible for us to carry out this validation with a larger data sample, since the transition between both tools in one specific academic year. However, we believe that making this tool available to the multimedia community will allow to replicate this type of validation when used by a larger set of students across different institutions.

5.2 | Results

The results for the 15-question survey described in Table 3 tests are summarized in Table 4. They show an average evaluation for all the questions between 3.30 (for Q7) and 4.50 (for Q2), being 3.94 the final average among all questions.

Figure 11 shows a box-and-whisker plot of the test results, where the average values have been marked as "x".

Regarding the evaluation of the transition between VisualMPEG and MPEG-2 Analyzer, Table 5 lists the marks obtained by the 14 students enrolled in 2012/2013 (i.e., VisualMPEG) and those obtained in the following year by the 29 students enrolled (i.e., MPEG-2 Analyzer). As can be seen in the table, the average mark achieved with VisualMPEG was 7 and with MPEG-2 Analyzer was 7.71.
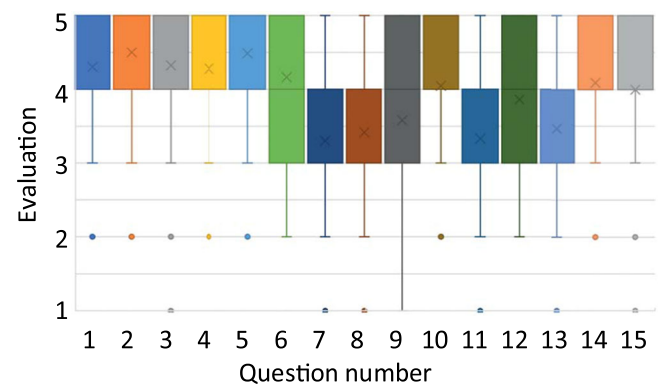

FIGURE 11 Test results (box-and-whisker plot). Average values marked as "x".

TABLE 4 Test results.

#	1	2	3	4	5	Avg.	Dev.
Q1	0 (0%)	1 (2.3%)	6 (14.0%)	15 (34.9%)	21 (48.8%)	4.30	0.80
Q2	0 (0%)	2 (4.8%)	1 (2.4%)	13 (31.0%)	26 (61.9%)	4.50	0.77
Q3	1 (2.3%)	3 (7.0%)	1 (2.3%)	14 (32.6%)	24 (55.8%)	4.33	0.99
Q4	0 (0%)	2 (4.7%)	2 (4.7%)	21 (48.8%)	18 (41.9%)	4.28	0.77
Q5	0 (0%)	1 (2.3%)	3 (7.0%)	13 (30.2%)	26 (60.5%)	4.49	0.74
Q6	0 (0%)	1 (2.3%)	10 (23.3%)	13 (30.2%)	19 (44.2%)	4.16	0.87
Q7	1 (2.3%)	4 (9.3%)	23 (53.5%)	11 (25.6%)	4 (9.3%)	3.30	0.86
Q8	1 (2.3%)	6 (14.0%)	16 (37.2%)	14 (32.6%)	6 (14.0%)	3.42	0.98
Q9	2 (4.7%)	6 (14.0%)	11 (25.6%)	13 (30.2%)	11 (25.6%)	3.58	1.16
Q10	0 (0%)	2 (4.7%)	6 (14.0%)	23 (53.5%)	12 (27.9%)	4.05	0.79
Q11	5 (11.9%)	4 (9.5%)	11 (26.2%)	16 (38.1%)	6 (14.3%)	3.33	1.20
Q12	0 (0%)	6 (14.0%)	8 (18.6%)	15 (34.9%)	14 (32.6%)	3.86	1.04
Q13	2 (4.7%)	6 (14.0%)	14 (32.6%)	12 (27.9%)	9 (20.9%)	3.47	1.12
Q14	0 (0%)	3 (7.0%)	4 (9.3%)	22 (51.2%)	14 (32.6%)	4.09	0.84
Q15	1 (2.3%)	3 (7.0%)	5 (11.6%)	20 (46.5%)	14 (32.6%)	4.00	0.98

Note: Number of answers (and percentage) from 1 to 5, average and standard deviation.

TABLE 5 Student marks obtained with VisualMPEG and MPEG-2 analyzer.

Student	VisualMPEG	MPEG-2 analyzer
Avg.	**7**	**7.71**
Dev.	**2.22**	**2.34**
S1	7	2.5
S2	6	2.5
S3	6	9
S4	8	9
S5	6	10
S6	6	10
S7	4	8
S8	4	5
S9	10	8
S10	9	6
S11	8	3
S12	4	9
S13	10	7.5
S14	10	8
S15		7.5
S16		8
S17		10
S18		10
S19		10
S20		8
S21		10
S22		10
S23		7.5
S24		8
S25		6
S26		6
S27		5
S28		10
S29		10

5.3 | Discussion

If we analyze the individual test results per question (shown in Table 4), we may identify potential aspects of the tool to be improved. For instance, it seems that the students may still have problems on understanding the relationship between the 8x8 block image and its DCT (Q7). Although in the last release of MPEG-2 analyzer we explicitly included a formula below the block to show how to compute the coefficients, we should put more emphasis on clarifying where each of the components comes from. Similarly, the results for Q9 may indicate that the encoding of a series of zero coefficients can be better exemplified. This is probably due to the inherent coding theory behind this process, that needs to be well understood.

Many other important aspects like the type of images, prediction, motion vectors, macroblocks, etc. have been very good evaluated (above 4). We should also notice that the average evaluation for Q15 (*My evaluation of the tool is positive*) is 4.00, which is considered very positive.

Results of the comparison between students' performances (shown in Table 5) should be interpreted with caution, as they are subject to considerable bias. It is common practice for course instructors to change the exam questions every year, and also to keep the overall number of questions limited. Therefore, the questions in the two exams being compared are not the same, nor are the students, nor the number of students. Nevertheless, we think that the results in the table reveal that MPEG-2 analyzer offers similar, even slightly better, results than other VisualMPEG. In this sense, the transition between these tools was smooth, and the students' experience has been improved through the years.

6 | CONCLUSIONS

We have introduced MPEG-2 Analyzer, an educational tool intended to provide a comprehensive and interactive way to understand video compression and encoding techniques. This tool has successfully been used in our institution for 7 years, and has received interest from numerous universities and institutions around the world.

We have used our tool to evaluate its usefulness in reinforcing the knowledge about the MPEG-2 video (de) coding process to 43 students of Telecommunications Engineering. The interactive GUI has been designed ad-hoc to allow students to obtain a better knowledge of compression and encoding techniques.

Results show that the overall evaluation of the tool is very positive, and it has helped students to understand many of the complex concepts explained in the class, especially those skills that include a visual component like the concept of block, macroblock, motion vectors, etc.

While it serves its purpose as an introductory step to multimedia technologies, more advanced features included in later versions of the standard (MPEG-4, or H.264/H.265) are not currently supported. Hence, the design of new modules to extend the functionality of the tool to more advanced video coding standards seems the more promising line of future development activities.

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Gerardo Gómez http://orcid.org/0000-0002-9715-7949

REFERENCES

1. A. Akgul, S. Kacar, I. Pehlivan, and B. Aricioglu, *Chaos-based encryption of multimedia data and design of security analysis interface as an educational tool*, Comput. Appl. Eng. Educ., **26** (2018), no. 5, 1336–1349.
2. W. Burleson, A. Ganz, and I. Harris, *Educational innovations in multimedia systems**, J. Eng. Educ., **90** (2001), no. 1, 21–31. https://doi.org/10.1002/j.2168-9830.2001.tb00563.x
3. M. Dick, S. Lahde, and L. Wolf, *MPEG analyzer a tool for visualizing MPEG encoding characteristics*, In NOSSDAV'08: Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video. 2008, pp. 121–122, New York, NY, USA. Association for Computing Machinery. ISBN 9781605581576. https://doi.org/10.1145/1496046.1496078
4. Z. Dogmus, E. Erdem, and V. Patoglu, *React!: an interactive educational tool for ai planning for robotics*, IEEE Trans. Educ., **58** (2015), no. 1, 15–24. https://doi.org/10.1109/TE.2014.2318678
5. Ericsson. Ericsson mobility report. June 2021.
6. Ó. Fresnedo, F. Laport, P. M. Castro, and A. Dapena, *Educational graphic tool for teaching fundamentals of digital image representation*, Comput. Appl. Eng. Educ., **29** (2021), no. 6, 1489–1504.
7. H. B. P. Gerson, S. A. Sorby, A. Wysocki, and B. J. Baartmans, *The development and assessment of multimedia software for improving 3-d spatial visualization skills*, Comput. Appl. Eng. Educ., **9** (2001), no. 2, 105–113.
8. V. P. G. Jimenez, A. L. Serrano, B. G. Guzman, and A. G.Armada, *Learning mobile communications standards through flexible software defined radio base stations*, IEEE Commun. Mag., **55** (2017), no. 5, 116–123. https://doi.org/10.1109/MCOM.2017.1601219
9. MPEG-2 analyzer web page, available at: http://www.ic.uma.es/contenidos/informacion_web.action?id=19&backurl=/contenidos/profesor_webs.action&request_locale=en
10. MPEG-2 standard implementation by MPEG Software Simulation Group. Initially available but not currently at http://www.mpeg.org/mssg/. Available on October 2021 at; https://github.com/ndavison21/lowrisc-mpeg2decode
11. Mpeg-2 standard, available at: https://www.mpegstandards.org/standards/mpeg-2/
12. Y. Murillo, B. Van Den Bergh, J. Beysens, A. Bertrand, W. Dehaene, P. Patrinos, T. Tuytelaars, R. V. Sabariego, M. Verhelst, P. Wambacq, and S. Pollin, *Multidisciplinary learning through implementation of the DVB-S2 standard.* IEEE Commun. Mag., **55** (2017), no. 5, 124–130. https://doi.org/10.1109/MCOM.2017.1700007
13. Qualify AVC/MP2 video analyzer. Available on 14-02-2023 at: https://www.dektec.com/products/applications/qualify/
14. T. d. J. M. Sanguino, C. S. López, and F. A. M. Hernández, *Wifisim: an educational tool for the study and design of wireless networks*, IEEE Trans. Educ., **56** (2013), no. 2, 149–155. https://doi.org/10.1109/TE.2012.2201157
15. Telestream MPEG Analyzers. Available on 14-02-2023 at: https://www.telestream.net/video/mpeg-analyzers.htm.
16. C. Tepe and E. F. Savaştaer, *A computer-aided educational device and toolbox for biomedical signal processing at the undergraduate level.* Comput. Appl. Eng. Educ., **30** (2022), no. 6, 1725–1738.
17. VisualMPEG analyzer. Originally available at. http://www.mpeg-analyzer.com, now discontinued.

AUTHOR BIOGRAPHIES

Unai Fernández-Plazaola is an Associate Professor and has been working in the Department of Communications Engineering at the University of Málaga since 1997. His main teaching activities include subjects such as Digital Image Processing, Video Fundamentals, and Communications Systems.

Gerardo Gómez is currently an Associate Professor at the Communications Engineering Department, University of Málaga. His main teaching interests include multimedia services, telecommunication networks, digital communications, and quality of service.

F. Javier López-Martínez is an EMERGIA Research Professor at the University of Granada, and an Associate Professor (on leave) at the University of Málaga. His main teaching interests include multimedia systems, wireless communications, and digital signal processing.

How to cite this article: U. Fernández-Plazaola, G. Gómez, and F. J. López-Martínez, *An educational tool for understanding MPEG-2 video compression and coding techniques*, Comput. Appl. Eng. Educ. (2023), 1–15. https://doi.org/10.1002/cae.22636