Macrothink Institute™
Network Protocols and Algorithms
ISSN 1943-3581
2017, Vol. 9, No. 1-2

Autonomous video compression system for environmental monitoring

Irene Mateos-Cañas[1], Sandra Sendra[1,2], Jaime Lloret[2], Jose Miguel Jimenez[2]

[1]Dept. of Signal Theory, Telematics and Communications Department (TSTC), Universidad de Granada, Granada, Spain

C/ Periodista Daniel Saucedo Aranda, s/n. 18071 (Spain)

[2]Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València.

C/ Paranimf nº 1, Grao de Gandía – Gandia, Valencia (Spain)

E-mail: wirnemc@gmail.com, ssendra@ugr.es, jlloret@dcom.upv.es, jojiher@dcom.upv.es

Received: April 5, 2017 Accepted: June 20, 2017 Published: June 30, 2017

DOI: 10.5296/npa.v9i1-2.12386 URL: https://doi.org/ 10.5296/npa.v9i1-2.12386

Abstract

The monitoring of natural environments is becoming a very controversial topic because people are more and more concerned about preserving and monitoring these natural spaces. The monitoring tasks are usually complemented with a network infrastructure composed by cameras and network devices that make easy the remote visualization of the monitored environments. This work presents the design, implementation and test of an autonomous video compression system for environmental monitoring. The system is based on a server in charge of collecting the videos and analyzing the network constraints. As a function of the measured parameters and the predominant color of the requested video, the system determines the best compression codec for transmitting the video through the network. Additionally, the server should run an algorithm developed in Python and MATLAB® in charge of analyzing the RED-GREEN-BLUE (RGB) components of the video and performing the transcoding tasks. The system has been tested with different videos and the results of Quality of Service (QoS) and Quality of Experience (QoE) shows that H264 is a good option when the predominant color of videos are black or white while XVID is one the codecs that offer interesting results when colors as red, green or blue are predominant in the video.

Keywords: Bandwidth, color spectrum, compression codec, decision algorithm, delay, jitter, packet-loss, quality of experience (QoE), quality of service (QoS), RGB level, RTP Transmission, transcoding, video codec, video monitoring.

1. Introduction

Wild fauna and flora threatened or endangered constitute one of the greatest environmental concerns of the past and present century, whether caused by human or natural origin. The problems that arises from this fact affects the whole ecosystem, both the environment where the species find their existence threatened and the consequences derive such as soil fertility, lack of resources to elaborate medicines or the quality of air and water, among many others [1]. The conservation of the environment is the tool for protecting and maintaining the ecosystem and the monitoring of the natural environment is a small help to facilitate conservation. Climate change refers to the global variation of Earth's climate. This disturbance is mainly attributed to human factors. The causes of this change are very diverse but can be encompassed in three major factors related to each other: (1) the increase of polluting emissions, (2) global warming and (3) greenhouse gases. The monitoring of the natural environments from which some of these causes derive helps us to obtain relevant data to record these events and to provide solutions to curb or mitigate possible future damage to the entire ecosystem.

The study of species and natural environments and their conservation can be done through the use of sensors or sensor networks which autonomously operate in a controlled way [2]. We can highlight the use of sensors that monitor environmental parameters such as temperature, humidity, radiation, or water levels of a bounded aquatic environment. In the case of animals, their monitoring can be carried out by means of control and surveillance with recording devices, with localization devices or through the use of sensors. In many cases, these sensors are installed directly on the animals, such as tied to the leg of a bird or the neck of a mammal. The recording devices allow direct observation of the behavior of these species allowing both knowing the status of these and acting in case of need. However, sensors do not allow direct observation but they can provide important data to combine with other monitoring mechanisms.

With the current and novel technology, it is possible to introduce monitoring mechanisms in natural environments. The deployment of a sensor network implies in most cases, the use of a wireless technology as EEE 802.11 [3] or Bluetooth to transmit the data and avoid interfering with the environment and wild fauna with cables and other elements.

When transmitting data through a network, it is important to know the network limitation. The most important factors to take into account in this aspect are the quality of the connection, speed and available bandwidth in order to ensure a sufficient Quality of Service (QoS). The consequence of a low QoS is a poor perception or image quality of the received video. This is known as Quality of Experience (QoE). All these factors are directly conditioned to the economic repercussion that the network deployment entails. The economic investment involved in recording devices such as drones or high definition cameras is considerably high, taking into account that these devices are in constant risk of destruction, loss or deterioration.

Because the nature of videos coming from natural environments is enormous and thanks to previous studies [4], we know that processing them in different ways can improve the network performance, in this paper, we are going to present an autonomous video compression system for environmental monitoring. The system is based on a server that executes our algorithm developed in Python and MATLAB® and analyzing the network constraints and the predominant color of videos provided by cameras. The algorithm analyzes the RED-GREEN-BLUE (RGB) components of the video and performs the transcoding tasks. Finally, the system is tested with different videos and the network performance is measured in terms of consumed bandwidth, packet loss, jitter and delay.

The rest of this paper is structured as follows. Section 2 discusses the related work. The proposal is depicted in Section 3. The results of the performed experiments are analyzed in Section 4. Finally, Section 5 presents the conclusion and future work.

2. Related Work

This section presents some previous works related to the adaptation of video formats according to the recorded content.

J. M. Jiménez et al. present in [4] a study about the most suitable compression codec to recode a video coming from the monitoring of a natural environment. This study takes into account the predominant characteristic color of the video, such as blue in an ocean, the green in a forest or the red in a desert. The rules of the decision algorithm used to select the best codec are based on both the compression time and the quality of the resulting image.

A. M Ferman et al. [5] present a summary of several color descriptors based on histograms to capture and reliably represent the color properties of multiple images or a *Gang of Four* (GoF)or design pattern. The goal of the study is to provide a solution to the challenge that involves the representation of the chromatic spectrum of the frames or images to make a better administration of visual information.

J. Lee, design in [6] a parametric transport layer model to monitor the video quality of IPTV services. Authors present the development of a network monitoring tool to evaluate the QoE depending on the physical characteristics of the IPTV system [11]. Based on this study and through this model it is possible to establish a more effective administration, implementation and design of IPTV services.

T. Zinner et al. [7] propose a method to establish a control mechanism that quantifies the most relevant parameters that influence QoE in video streaming applications based on the H.264 / SVC codecs. Through the obtained results, authors design a system that obtains results regarding to the impact of video encoding over the QoE perceived by the user through the resolution of the video or frame rate of the video, using metrics such as Video Quality Management (VQM).

A. Neogi et al. [8] present a study focused on the compression techniques for interactive video content. Starting from the main goal of offering to the end user additional functionalities while reproducing the video, the authors encounter the problem of storage and transmission costs that are considerably high due to the synchronization of multiple video streams. To solve these limitations, authors analyzed and evaluated different compression techniques.

The exposed systems are studies to determine the best codec for encoding a video. However, this work presents a practical implementation of an autonomous system able to carry out all the process. Additionally, the fact of combining the effect that the predominant color has on the recorded video and the network imitations that its transmission can lead, implies to have complete proposal and a useful tool easily implemented in almost all scenarios.

3. Proposed system

In this section describes the process carried out to encoding the video from environmental surveillance to obtain the best results regarding network performance and video quality. Additionally, this section presents the different algorithms used to analyze the captured video and to automatically transcoding the original video to the optimal videos to be streamed.

Figure 1 shows the full process carried out to capture the video, analyze its content and finally, transcoding it to be analyzed.

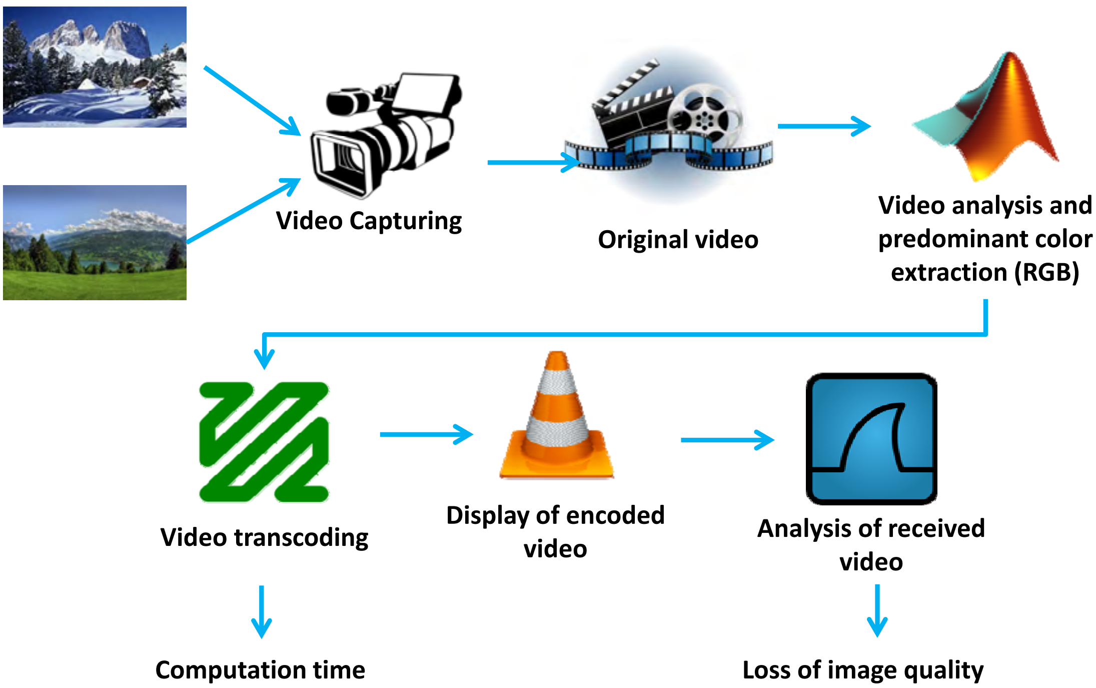

Figure 1. Description of video transcoding process.

3.1 Video analysis and extraction of the predominant color

In order to determine the best codec to encoding a video recorded in a natural environment, taking into account its average RGB level and its predominant color, a short algorithm has been developed. The algorithm quickly and efficiently determines which ones are the color parameters of the captured video. To do this, we have developed the algorithm in MATLAB®. MATLAB® consists of a very powerful and specific tool that allows performing all kinds of data processing.

To develop our proposed algorithm, we have used the integrated applications of image processing *VideoViewer* and the native functions offered by MATLAB®. Through this integrated application, it is possible to perform multiple tasks as viewing a video, extracting relevant information (duration or the frame rate, among others).

The algorithm in charge of determining the predominant color consists of five steps (See Figure 2):

1. ***Read a video stored in the system using the VideoReader function***. This function creates an object capable of being processed and associated with a file that contains a video. Depending on the platform where MATLAB® is running, some formats will be valid. For example, Macintosh supports the following video file formats: MPEG-1 (.mpg), MPEG-4 including video encoded in H.264 (.mp4, .m4v), Apple Quick Time Movie (.mov), 3GPP, 3GPP2, AVCHD and DV.

2. ***Obtaining video parameters to analyze and process the video***. The *Height* and *Width* functions return the resolution video, that is, the height and width of the set of frames measured in pixels. During this task, the number of frames of the video is obtained by applying the *Ceil* function, which returns the approximate integer closest to the obtained value, the product of the duration of the video (obtained with the *Duration* function) and the frame rate per second (obtained with the *FrameRate* function).

3. ***Create Y Matrix.*** An auxiliary matrix 'and' filled with zeros with the size "3·Num.of frames" is created since for each frame we will have a matrix with the red ‘R’ values, a matrix with the green ‘G’ values and a matrix with the blue ‘B’ values.

4. The algorithms’ main loop is in charge of ***analyzing the video file frame by frame***. In each iteration of the main loop, a secondary loop is executed. This secondary loop analyzes the pixels features of each row of pixels. Additionally, the developed algorithm stores the data of pixels characteristics in three auxiliary variables. At the end of the secondary loop, the mean of the RGB level of the analyzed frame is calculated and stored in 'Y'. Finally, the result of this loop is the matrix 'Y' which will contain the average RGB levels of each frame.

5. Finally, the ***RGB average is calculated*** from the 'Y' matrix, obtaining the variable *RGB_results* that will contain the average RGB levels of the whole video object of analysis, being able then to observe which the predominant color is.

Figure 2 presents the operation diagram of the automatic RGB values calculation algorithm which is essential to perform the video classification.

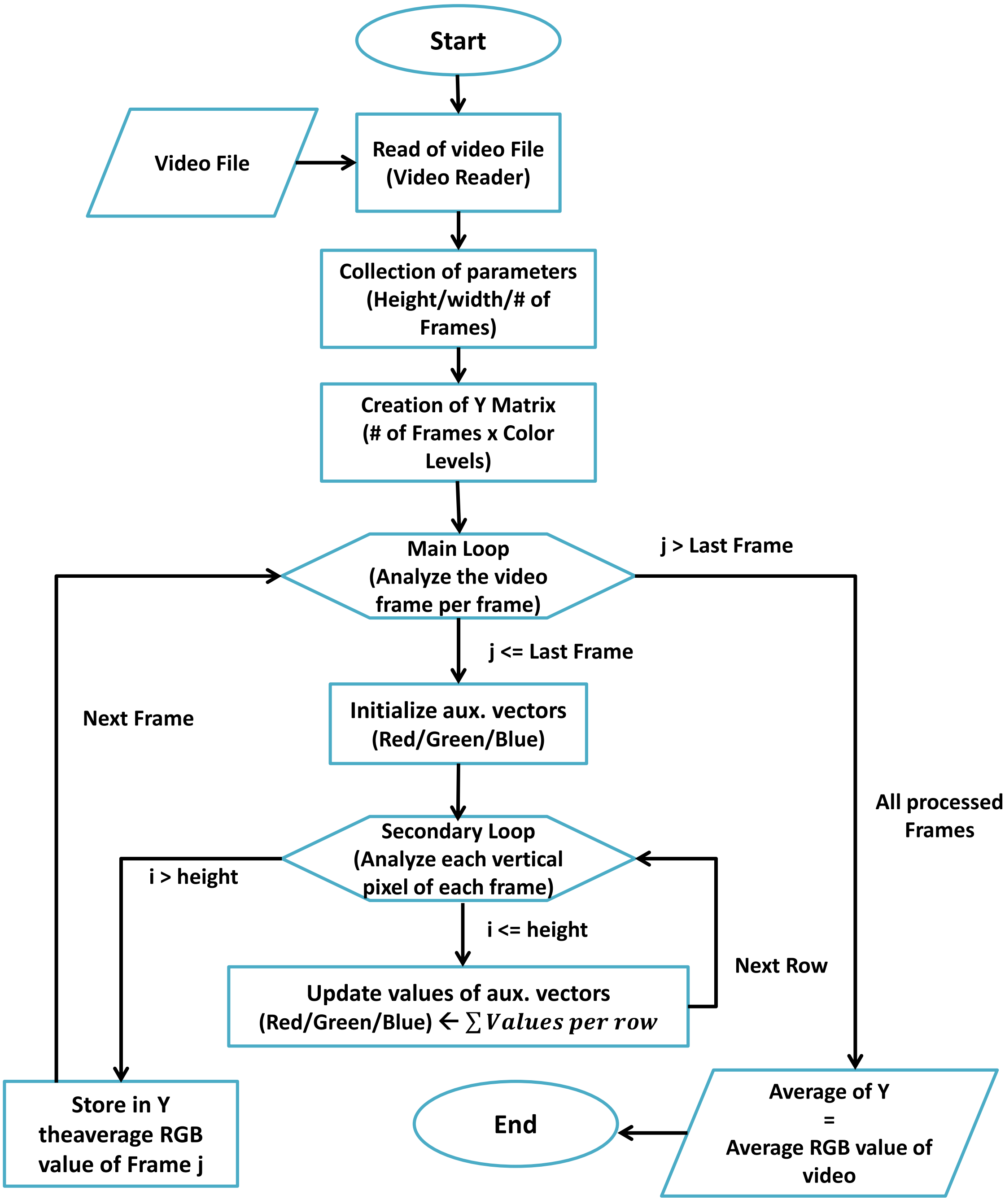

Figure 2. Algorithm for determining the predominant color

In order to test the correct operation of our algorithm, different videos which present different predominant colors have been analyzed. Furthermore, it is important to note that the RGB level of pure colors present the following features (see Table 1):

Table 1. Features of pure colors

Chroma of Pure	RGB Levels		
	Red (R)	***Green (G)***	***Blue (B)***
Chroma of Pure red	255	0	0
Chroma of Pure green	0	255	0
Chroma of Pure blue	0	0	255
Chroma of Pure white	255	255	255
Chroma of Pure black	0	0	0

The results of this study are presented in Table 2:

Table 2. Video processing result by RGB algorithm implemented in MATLAB®

Original Video File	RGB Levels		
	Red (R)	***Green (G)***	***Blue (B)***
blue.mp4	37	117	**154**
whales.mp4	24	70	**149**
red.mp4	**127**	79	51
eleph.mp4	**131**	91	58
green.mp4	67	**93**	64
dogs.mp4	89	**97**	55
black.mp4	**44**	**44**	**44**
white.mp4	**186**	**195**	**210**

As Table 2, for each original video files the predominant chroma has been highlighted, with the exception of the videos whose chromatic spectrum turns out to be white or black, since these two chromas are characterized by:

- White chroma: its three RGB levels have a value very close to 255 and a very small difference between them.
- Black chroma: its three RGB levels have a value very close to 0 and a very small difference between them.

3.2 Considerations related to codec compatibility

We must know that not all video formats (or containers) are compatible with all codecs. A small study based on tests has been carried out with FFmpeg and VLC to check which codec used in this work is compatible with the containers. Table 3 shows the results of compatibilities of video formats and codecs:

Table 3. Relationship between containers and compatible codecs.

Original Video Format	Compatibility of Codecs					
	APCN[9]	*FMP4 [10]*	*H264 [11]*	*WMV1 [12]*	*FLV1 [13]*	*XVID [13]*
.mp4	×	✓	✓	×	×	×
.avi	✓	✓	✓	✓	✓	✓

3.3 Network operation algorithm

In order to serve the requested video, it is important to have available the videos. When a client requests to connect to a surveillance camera, the request is initially received by the server which redirects the request to the cloud that store the video captured by surveillance cameras. Additionally, the server should ask to the client about the limitations. After connecting with the cloud that stores the videos, the server receives the video and analyzes its features. Finally, the video is sent to the client. Figure 3 shows the message exchange between a client and the video server.

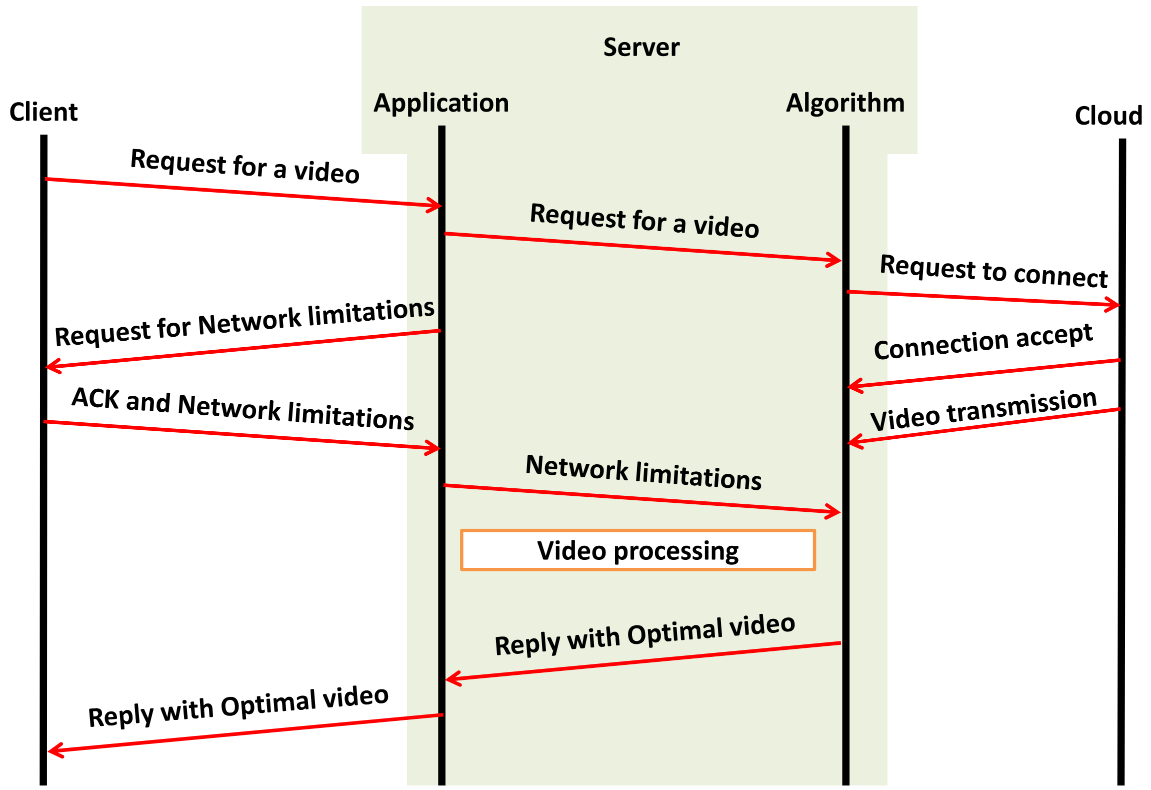

Figure 3. Message exchange between a client and the video server.

Additionally, we need to stablish a connection protocol between the cloud and the server and the server and the client. To carry out these tasks, we have implemented a concurrent TCP socket to establish the communication between both the cloud and the server (see Figure 4a) and a UDP socket to establish the communication between both the server and the clients (see Figure 4b). As Figure 4 shows the server is able to assist several requests from several clients. In the same way, as the server receives a request for a video, the server transmits this request to the cloud in order to download the video. The advantage of using concurrent TCP socket-based protocol is that the cloud or the server can assist several requests, creating for each new request a thread that opens a new process to start with a possible transcoding of a requested video. When the server knows the network requirements, the designed algorithm transcodes the original video according to these requirements and the client does not need to install any additional software. The use of TCP connections between both the cloud and the server ensures a reliable connection with congestion control and flow control mechanisms that guaranties the correct delivery of all data and messages. However, it is recommended to use UDP connections between the server and clients because one of the most important issues is to deliver the requested video as quick as possible.

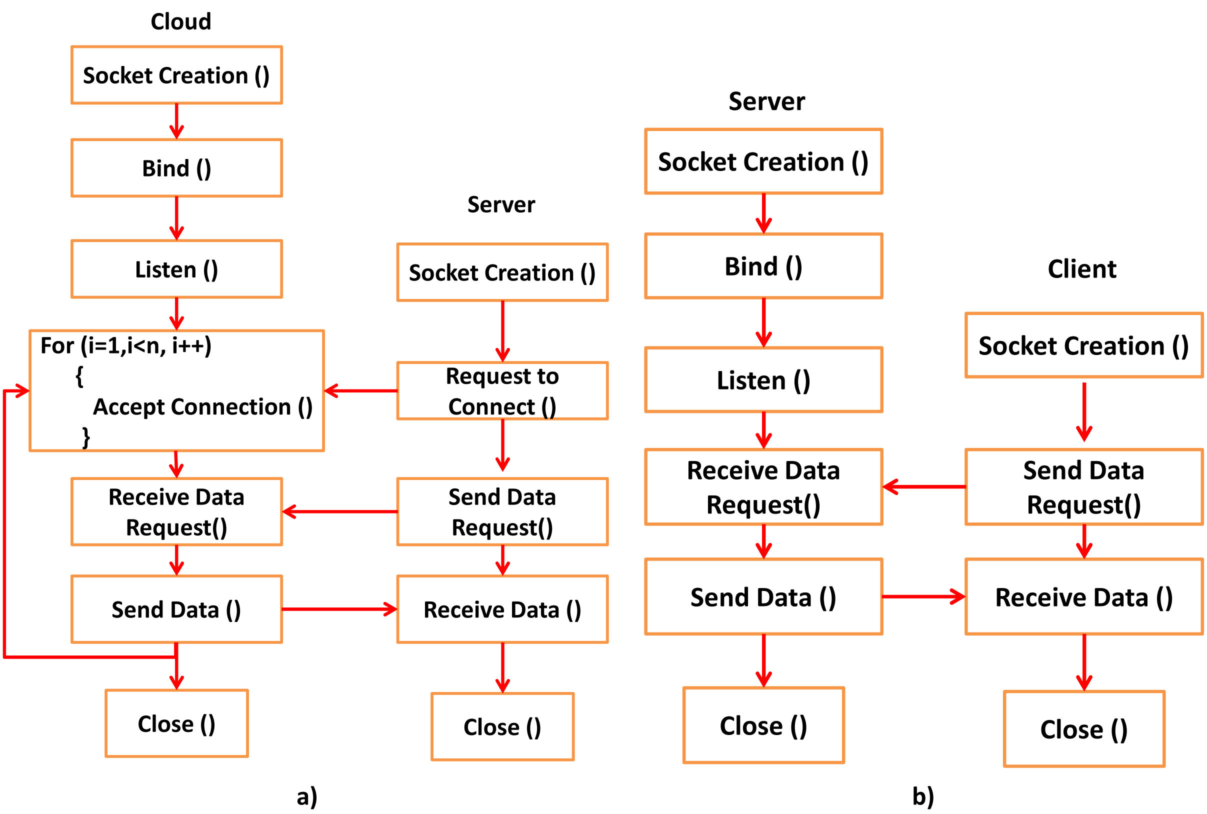

Figure 4. Flow diagram of connection between a) the cloud and the server and b) the server and clients.

3.4 Decision algorithm

In order to automatize the process of video encoding, it is needed to develop a decision algorithm that should take onto account the limitations of the network. As design rules, we take into consideration a previous study [4] As a summary of this article and design rules for our decision algorithm, we elaborated Table 4.

Table 4. Design rules for our decision algorithm

Predominant Color	Selected Codec as a function of the parameter						
	Compression size (%)	*Transcoding time*	*Subjective Quality*	*Consumed Bandwidth*	*Maximum Delay*	*% of Packet Loss*	*Maximum Jitter*
RED GREEN BLUE	H264	H264/XVID	APCN/H264	H264	H264	H264/XVID	H264
WHITE BLACK	H264	XVID	APCN/H264	XVID	XVID	H264/XVID	XVID

The algorithm (See Figure 5) works as follow:

1 Given a video file captured by the camera, firstly, the system checks if there are some QoS restrictions in the network.

2 If not, the video with the original features and codec will be transmitted. If there are such restrictions, the system will check if there are also QoE requirements.

3 If there is not QoE restrictions, the system analyze the video content to check if black or white are the predominant colors. In such case, H264 will be selected to encoding the video. In any other case, XVID will be used.

4 If there are QoE requirements and the predominant color of video is black or white, the video will be encoded with XVID. Finally, if red, green or blue are detected as predominant color, the video will be encoded with H264.

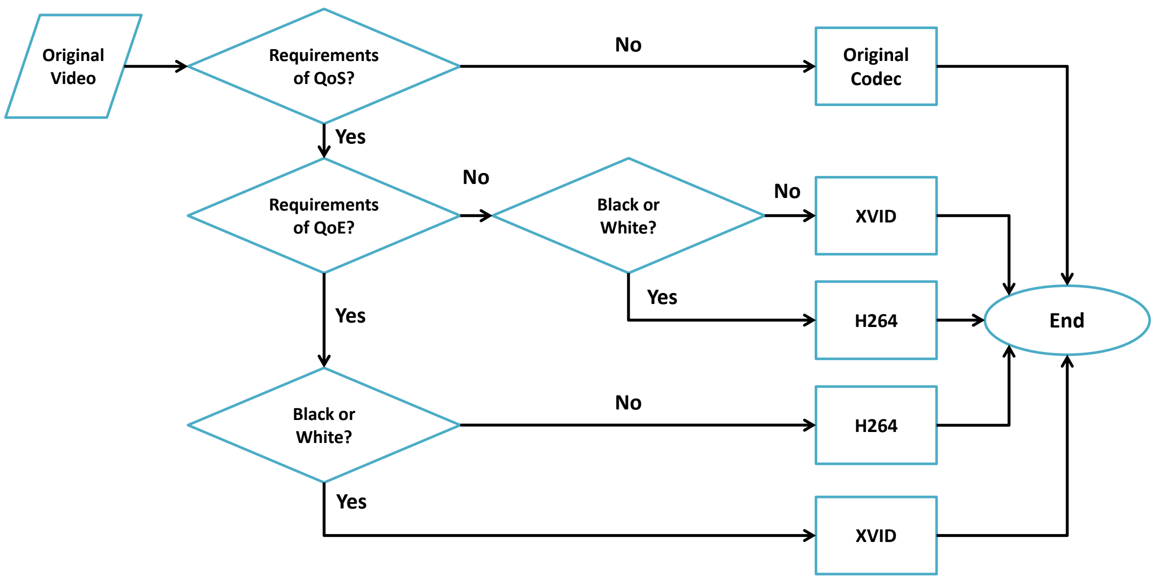

Figure 5. Decision algorithm designed for automatically selecting the most suitable compression codec

4. Test bench and Results

To check the correct operation of our system, we have implemented a topology composed by a client laptop wirelessly connected to a router. On the other side, a server that access to a cloud where the original videos are stored is running our algorithm (See Figure 6). The different videos have been requested to the server and the different network parameters have been measured. This section shows the results of network performance can offer when transcoding the video in different codecs over some network conditions. The computation time for transcoding the videos, QoS and QoE results are presented.

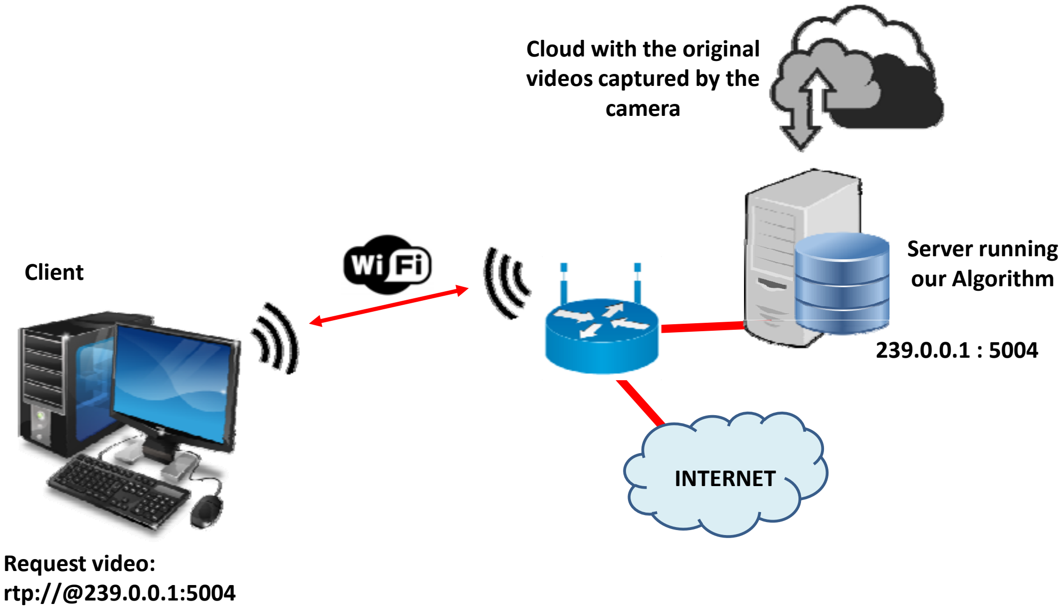

Figure 6. Proposed scenario for transmitting the video

4.1 Results of FFmpeg

One of the main issues in real-time streaming of videos is the computation time to adapt the video to the network constraints.

The first test try to compare the computing time our system needs to transcoding our original videos. Additionally, the compression ratio is also an interesting factor to be considered.

As a first step prior to the transcoding our files, we have performed a change of video container to the .avi container, since it is the container format that offers the most compatibility when it comes to transcoding.

Firstly, Figure 7 shows the compression ratio for each codec. To check this parameter, we have compared the size data of the original video in .avi and the size data of the final file. In the results, the APCN results have been omitted because in most cases the size file increases in a 1000%.

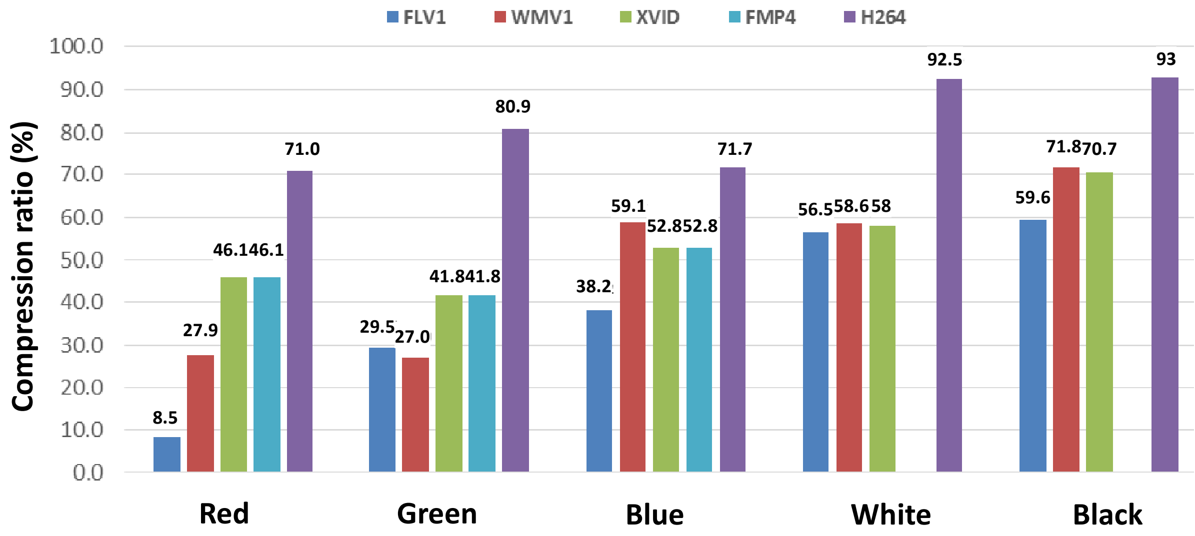

Figure 7. Compression ratio obtained in each case.

As Figure 7, in general, the compression codec H264 offers a percentage reduction in size greater than the rest of codecs independently of the chromatic spectrum present in the video.

By means of the real time parameter that is used in recoding a video, a deduction can be made of the video codec that offers a more adequate solution for the result of the time used to perform its compression.

Table 5 shows the results obtained by the FFmpeg application regarding to the elapsed time when each video in different codecs are transcoding. The videos called 'red', 'blue', and 'green' have not been included since their duration were around 1s and the results of processing are not relevant.

The temporary parameter that is going to be analyzed is the one called 'Real' and the justification offered is that it is the temporary parameter that explicitly measures the clock time it takes to execute the command from the moment it is executed. This value does not take into account the internal processes of the CPU or processes that other users of the system are running, as the case of 'User time' and 'System time'. These values are provided by FFmpeg.

Table 5. Results of the consumed time to transcoding the video

Video Description	Results of the elapsed time			
	Output Codec	**Real time (s)**	**User time (s)**	**System time (s)**
Whales (Blue as Predominant color)	APCN	1.770	6.817	0.005
	H264	0.331	1.005	0.013
	FLV1	0.351	0.773	0.009
	WMV1	0.355	0.793	0.007
	XVID	0.334	0.966	0.013
	FMP4	2.427	9.018	0.047
Dogs (Green as Predominant color)	APCN	1.203	4.638	0.035
	H264	0.242	0.738	0.010
	FLV1	0.398	0.582	0.008
	WMV1	0.293	0.585	0.006
	XVID	0.241	0.744	0.010
	FMP4	1.700	6.273	0.035
eleph (Red as Predominant color)	APCN	1.166	4.506	0.031
	H264	0.241	0.765	0.081
	FLV1	0.280	0.587	0.007
	WMV1	0.300	0.594	0.006
	XVID	0.252	0.755	0.012
	FMP4	1.820	6.737	0.036
Black (Black as Predominant color)	APCN	0.259	0.927	0.180
	H264	0.147	0.434	0.023
	FLV1	0.055	0.067	0.005
	WMV1	0.062	0.074	0.005
	XVID	0.049	0.106	0.008
	FMP4	0.244	0.400	0.017
White (White as Predominant color)	APCN	0.535	1.927	0.036
	H264	0.323	0.905	0.056
	FLV1	0.135	0.170	0.010
	WMV1	0.136	0.171	0.009
	XVID	0.114	0.257	0.016
	FMP4	1.054	0.157	0.010

Finally, Figure 8 presents the compression time in seconds that out system has used to transcoding a video depending on the predominant color of video. These values are provided by the FFmpeg program with their internal functions.

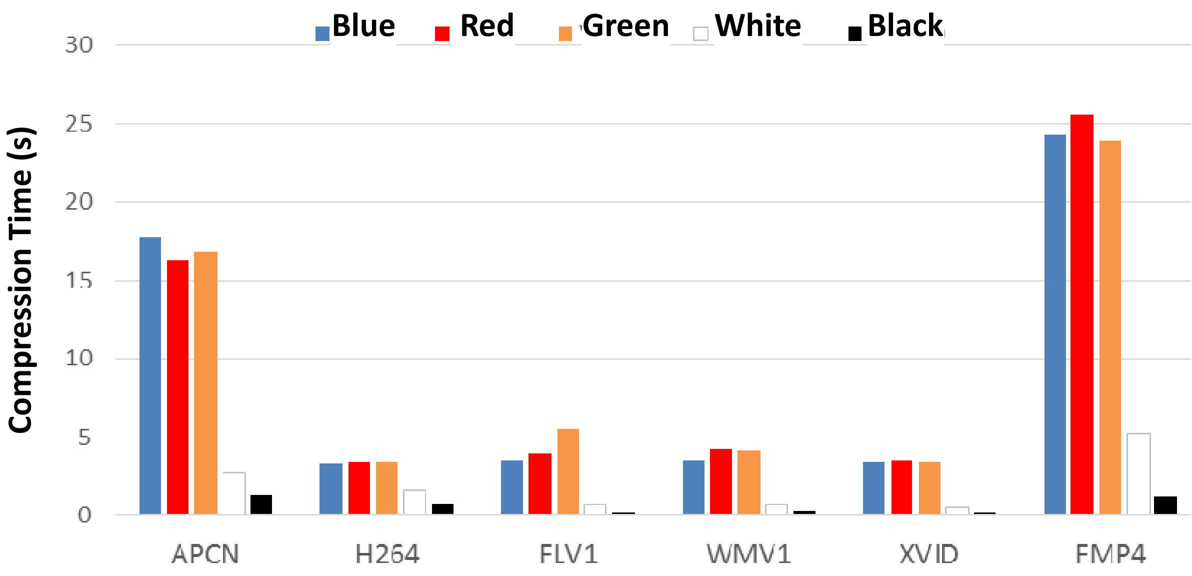

Figure 8. Compression time

The results shows that H264, FLV1, WMV1 and XVID are the codecs that presents de lowest compression time while FMP4 needs at least 5 times more in performing the same task. Additionally, we have observed that videos where black or white are the predominant color the process of transcoding is the fastest one.

4.2 Network performance results

After implementing the proposed algorithm, we need to check the network performance and visual appearance of the received video. The goal is to obtain the QoS parameters in order to determine which codec is more suitable according to the results offered in its transmission. These tests have been carried out with the VLC and Wireshark software which are tools for reproducing and analyzing multimedia files and for analyzing and capturing network traffic. This subsection shows the results of this study that has collected results of the consumed bandwidth, the packet loss, the latency and the jitter register during the transmission.

4.2.1 Consumed bandwidth

The average values of bandwidth consumed when the different videos are transmitted through the network are shown in Figure 8.

As Figure 9 shows the consumed bandwidth varies considerably as a function of the predominant color regardless of the type of recoding performed. It is possible to highlight that, in the case of having a video in which the predominant colors are white or black, the H264 codec offers a remarkable result in terms of the consumed bandwidth. When the predominant colors of the video are red, green or blue, the XVID codec offers the best result by providing a very low consumed bandwidth.

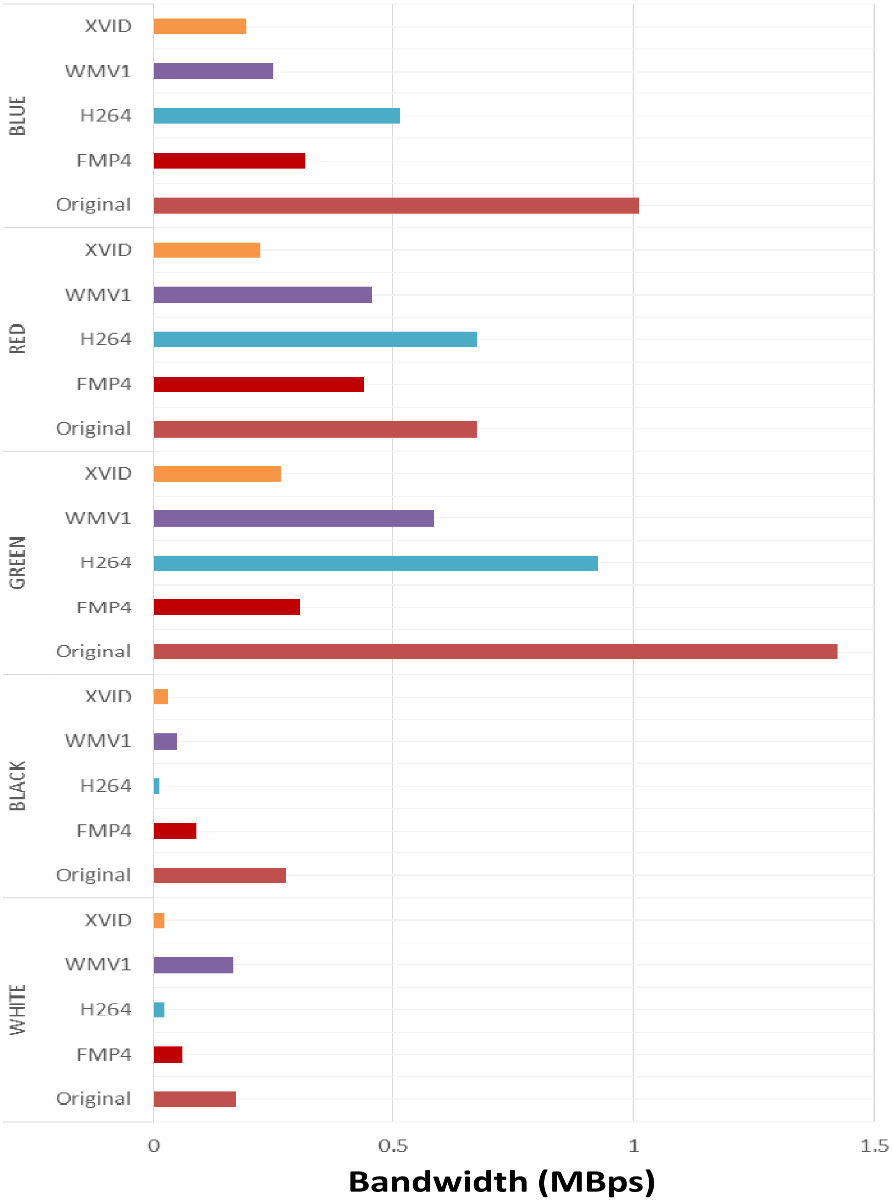

Figure 9. Average consumed bandwidth

4.2.2 Results of delay

Figure 10 shows the average value of delay registered when each video with different codecs is transmitted and Figure 11 shows the maximum values of delay registered when each video with different codecs is transmitted.

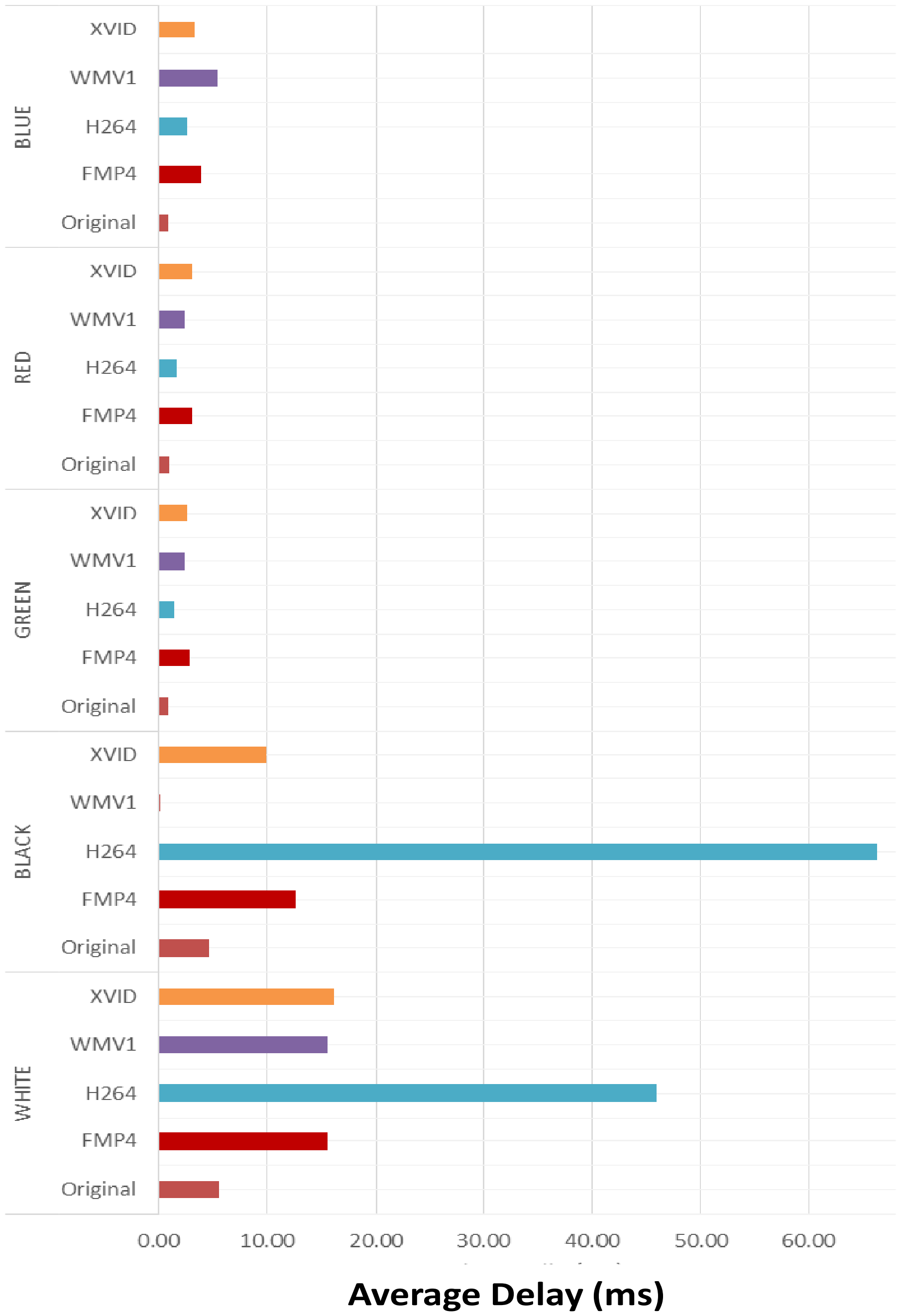

Figure 10. Average delay

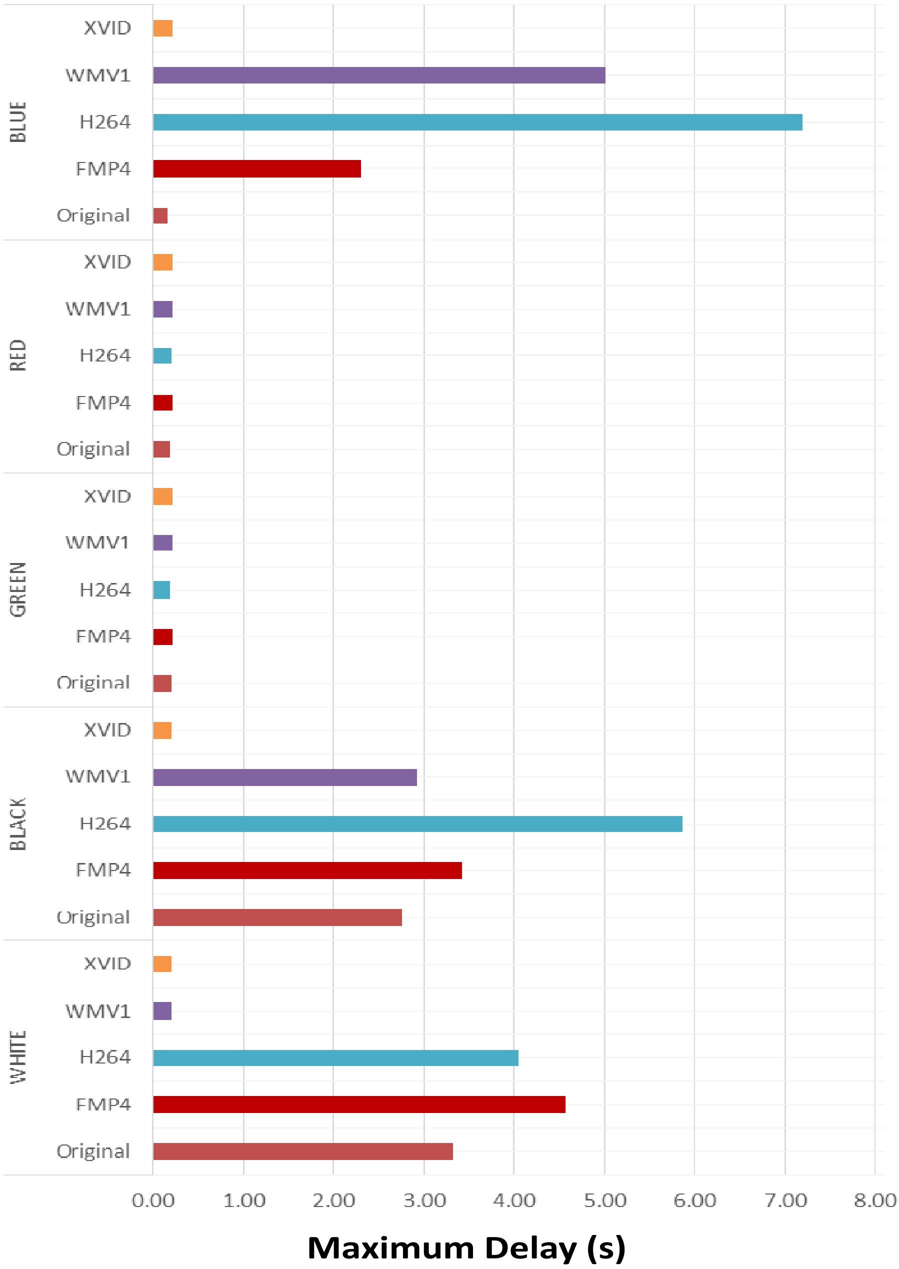

Figure 11. Maximum delay

As Figure 10 shows when files whose predominant colors are red, green or blue, the codec that offers the best results is H264, observing good results also when the XVID codec is used. If we consider the maximum delay registered for each case (results of Figure 11), the XVID codec would be the best option.

For videos where the predominant color is white or black, for both cases, the average delay and the maximum delay are the lowest one when XVID codec.

4.2.3 Results of jitter

Figure 12 shows the average value of average jitter registered when each video with different codecs is transmitted and Figure 13 shows the maximum values of jitter registered when each video with different codecs is transmitted.

According to the results when the color that predominates in the multimedia stream is blue, red or green, the codec that offers the best result in terms of average and maximum jitter is the H264 codec. When white or black are the predominant colors, the codec that seems to be the most optimal is WMV1. Because WMV1 is a proprietary codec, the XVID codec could be taken as a second option:

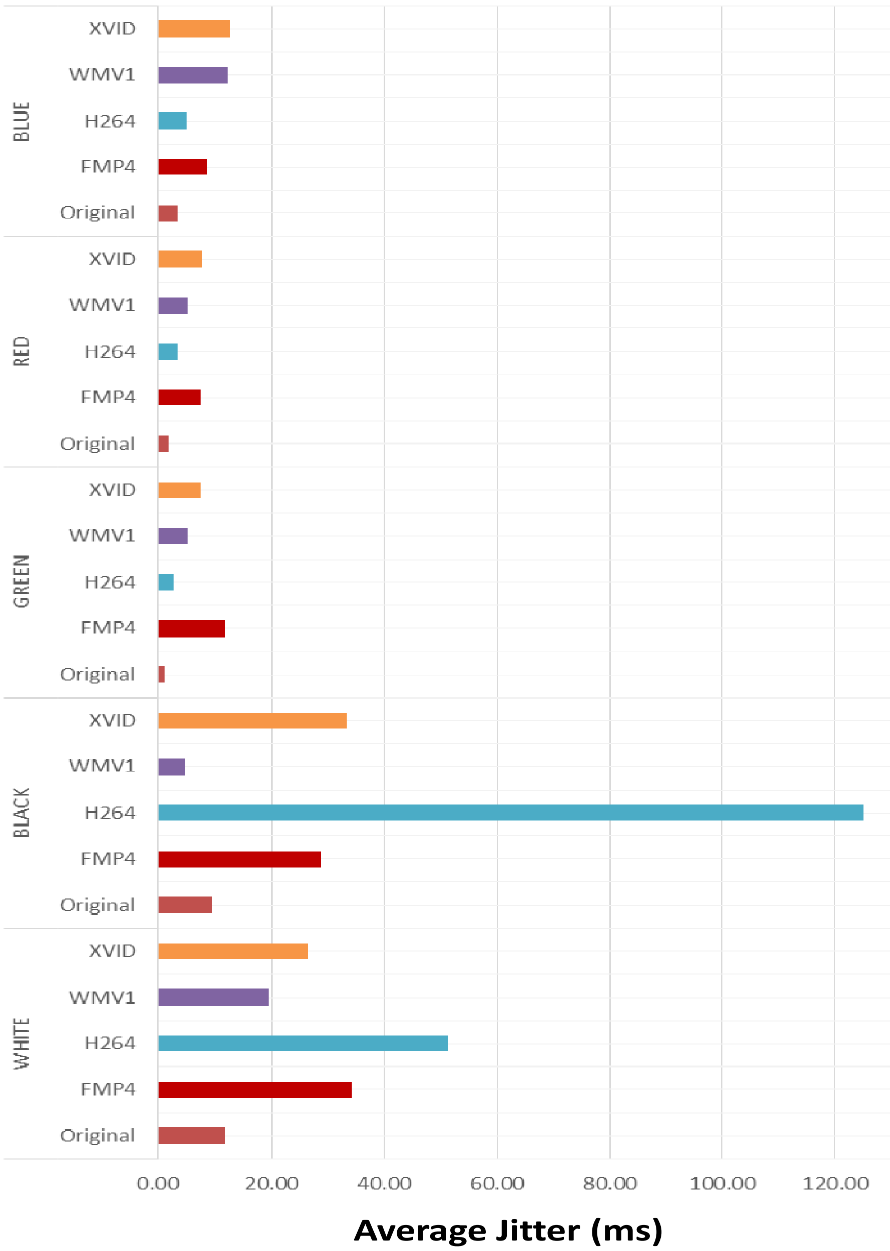

Figure 12. Average jitter

Network Protocols and Algorithms
ISSN 1943-3581
2017, Vol. 9, No. 1-2

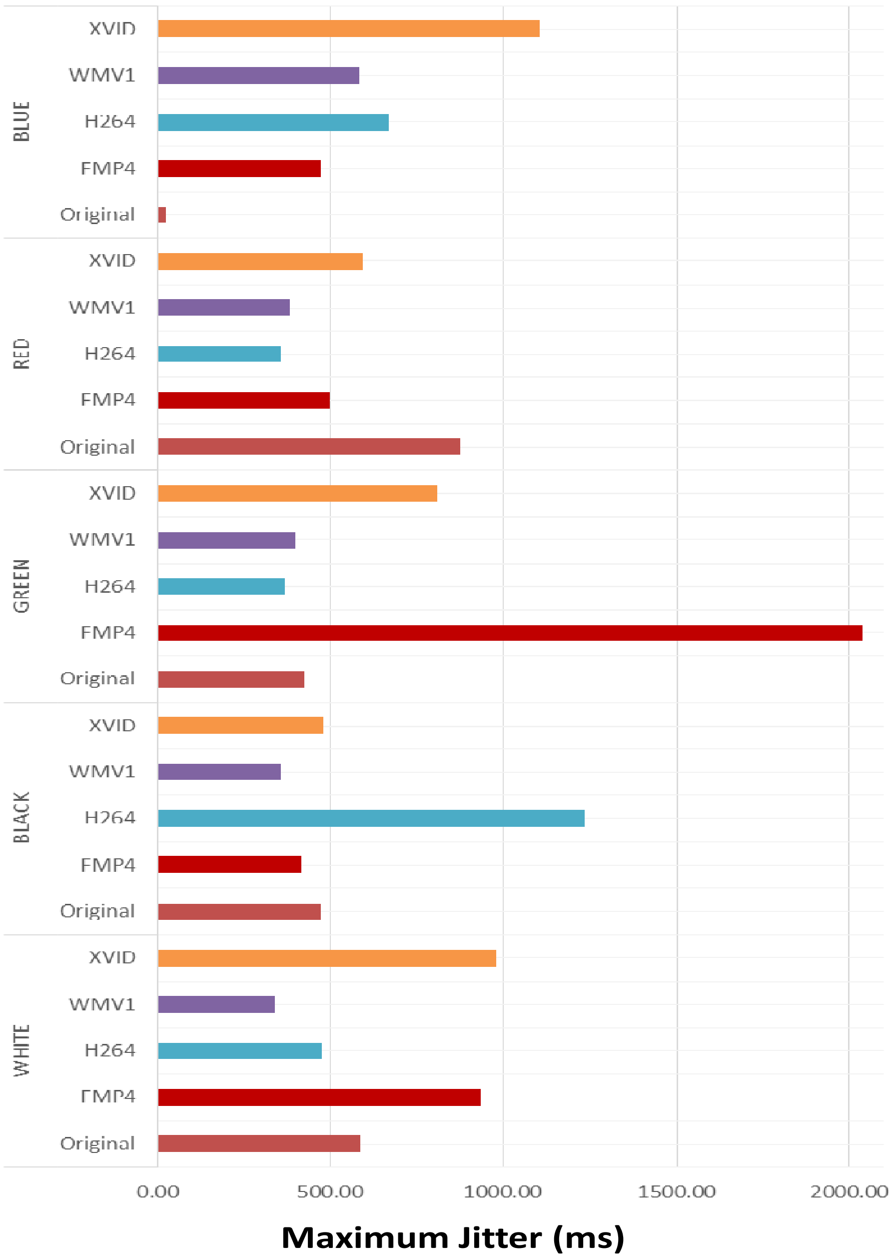

Figure 13. Maximum jitter

4.2.4 Results of packet loss

The last parameter we need to analyze the QoS of network is the percentage of packet loss when each video is transmitted (See Figure 14).

According to the previous results, when the predominant colors are red, green or blue, the codec that presents the lowest packet loss percentage is the XVID codec while the codec H264 offers the better results when white or black colors are the predominant ones in the video.

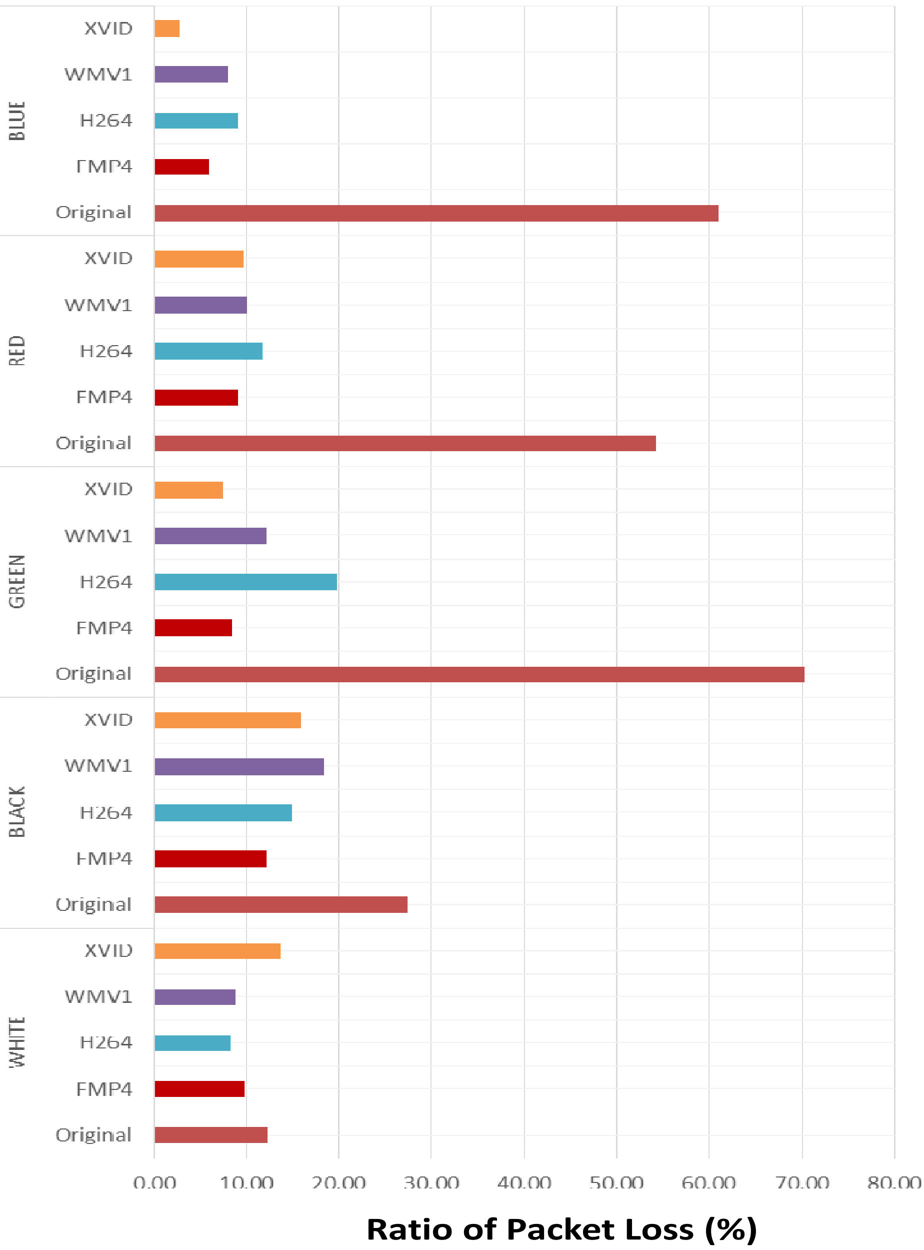

Figure 14. Percentage of packet loss

4.2.5 Subjective perception of videos after streaming

To conclude the test bench of our system, we want to check the perceived quality by the users. To determine it, we have shown the original video and the transcoding video to a set of persons to express the relationship between both videos.

Table 6 shows the results of the subjective perception of videos after streaming through the network. the videos have been tagged as Very Good, Good, Medium, Low and Very Low as a function of the image quality observed, being for a category 'Very Good' the quality closest to the original and 'Very low' which has been appreciated more distorted. Some of the main effects of the visual perception given by the variation of the jitter and delay are ghosting, blurring, error chrominance, etc. Due to the packet loss in the network, we can observe (among others effects) black pixels and finally, if a video has been incorrectly transcoded, we can observe problems of blurriness or hard blockness, among other visual effects.

Table 6. Subjective perception of videos after streaming through the network.

Output Codec	**Subjetive Quality**				
	eleph.avi (Red)	***dogs.avi (Green)***	***whales.avi (Blue)***	***white.avi (White)***	***black.avi (Black)***
APCN	Very Good	Very Good	Very Good	Very Good	Very Good
FLV1	Medium	Medium	Medium	Medium	Medium
WMV1	Low	Medium	Medium	Medium	Medium
XVID	Medium	Medium	Medium	Medium	Medium
FMP4	Low	Medium	Medium	Medium	Medium
H264	Good	Good	Good	Good	Good

As Table 6 shows APCN codec offers a quality almost equal to that of the original video although it presents the worst results in terms of percentage compression ratio and compression time. On the other hand, H264 codec offers a good quality in relation to the original video file which means that this codec stands out over the others. Finally, FMP4, in addition of presenting very poor performance, presents very low quality in the perception of resultant video.

5. Conclusion

The use of video surveillance systems in the tasks of environmental monitoring is nowadays widely used. When video flows are transmitter through a network, we should try to obtain the best quality in the received video. For this reason, this paper has presented the implementation and test of an autonomous video compression system that adapts the video format to the network constraints. The system is based on an algorithm determines which is the best compression codec for transcoding the video as a function of the measured network parameters and the predominant color of the requested video. The results have demonstrated that in terms of QoS and QoE, the H264 codec is a good option when the video predominant color of videos are black or white while XVID offer interesting results when red, green or blue are the predominant colors in the video.

As future works, we would like to investigate the influence of secondary colors in a sequence with a certain predominant color or group of colors. We also want to investigate the frequency with which, the execution of the algorithm must be executed to achieve the best balance between the improvements in the videos and the overload of the system. Because the final goal is to have an adaptive video streaming system while maintaining the QoE of the final user [14], as well as other authors have done for VoIP [15], we would like to improve the code implemented in Python and add more functionality in terms of codec decision taken, video conversion formats and the selection of capture device among others. Finally, it could be interesting to add, other parameters related to image properties such as luminance or exposure could be included.

Acknowledgement

This work has been supported by the "Programa para la Formación de Personal Investigador – (FPI-2015-S2-884)" by the "Universitat Politecnica de Valencia". The research leading to these results has received funding from "la Caixa" Foundation and Triptolemos Foundation

References

[1] Urry J. "Climate change and society". Why the social sciences matter. pp. 45-59. 2015. Palgrave Macmillan, London. DOI: https://doi.org/10.1057/9781137269928_4
[2] Lloret J., Mauri P. V., Jimenez J. M., Diaz J. R. "802.11 g WLANs design for rural environments video-surveillance". In proceedings of the ICDT'06 International Conference on Digital Telecommunications, 2006. Cap Esterel (France). August 29-31, 2006 (pp. 23-23). DOI: 10.1109/ICDT.2006.1
[3] Garcia M., Lloret J., Edo M., Lacuesta R., "IPTV distribution network access system using WiMAX and WLAN technologies". In Proceedings of the 4th edition of the UPGRADE-CN workshop on Use of P2P, GRID and agents for the development of content networks, Munich (Germany), June 11 - 13, 2009. (pp. 35-44). Doi:10.1145/1552486.1552513
[4] Jiménez J. M., Díaz J. R., Sendra S., Lloret J., "Choosing the Best Video Compression Codec Depending on the Recorded Environment". In proceedings of the 2014 IEEE Global Communications Conference (GLOBECOM 2014), Austin, TX (USA), December 8-12, 2014. (pp. 1347-1352). DOI: 10.1109/GLOCOM.2014.7036995
[5] Ferman A.M., Tekalp A.M., Mehrotra R. "Robust color histogram descriptors for video segment retrieval and identification". IEEE Transactions on Image Processing, Vol.11, No.5, pp.497-508. May 2002. DOI: 10.1109/TIP.2002.1006397
[6] Lee J. "Automatic prefilter control by video encoder statistics". Electronic Letters, Vol.38, No.11, pp.503-505. 2002. DOI: 10.1049/el:20020361
[7] Zinner T., Hohlfeld O., Abboud O., Hossfeld T. "Impact of frame rate and resolution on objective QoE metrics". In proceedings of the Second International Workshop on Quality of Multimedia Experience (QoMEX 2010), Trondheim (Norway) June 21-23, 2010. DOI: 10.1109/QOMEX.2010.5518277
[8] Neogi A. and Chiueh T., "Compression techniques for active video content". In Proceedings of the Data Compression Conference (DCC 2002), April 2-4, 2002. Snowbird, UT, (USA). (p. 466-472) DOI: 10.1109/DCC.2002.1000009
[9] APCN features. In Apple web site. Available at: https://www.apple.com/support/assets/docs/products/finalcutpro/Apple_ProRes_June_2014_loc_es_GLOBAL.pdf [Last access: December 10, 2017]

[10] FMP4 features. In FourCC web site. Available at: https://www.fourcc.org/fmp4/ [Last access: December 10, 2017]

[11] Bahri N., Grandpierre T., Ayed M. A. B., Masmoudi N., Akil M., "Embedded real-time H264/AVC high definition video encoder on TI's KeyStone multicore DSP", Journal of Signal Processing Systems, vol. 86, Iss. 1, pp. 67-84, 2017. https://doi.org/10.1007/s11265-015-1098-x

[12] WMV1 features. In Microsoft web site. Available at: https://msdn.microsoft.com/es-es/library/windows/desktop/ff819510(v=vs.85).aspx [[Last access: December 10, 2017]

[13] https://offers.adobe.com/en/na/leap/landings/fmle3.html FLV [Last access: December 10, 2017]

[14] XVID features. In XVID web site. Available at: https://www.xvid.com [Last access: December 10, 2017]

[15] Álvarez A., Poxueco L., Cabrero S., García Pañeda X., García R., Melendi D., Díaz Orzuela, G. "Adaptive Streaming: A subjective catalog to assess the performance of objective QoE metrics", Network Protocols and Algorithms. Vol 6, Iss. 2, pp. 123-136, 2014. DOI: http://dx.doi.org/10.5296/npa.v6i2.5461

[16] Estepa A.J., Estepa R., Vozmediano J., Carrillo P. "Dynamic VoIP codec selection on smartphones", Network Protocols and Algorithms. Vol 6, Iss. 2, Pp. 22-37, 2014. DOI: http://dx.doi.org/10.5296/npa.v6i2.5370

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).