Author's Accepted Manuscript

www.elsevier.com/locate/image

Adjustable compression method for still JPEG images

Jerónimo Mora Pascual, Higinio Mora Mora, Andrés Fuster Guilló, Jorge Azorín López

PII: S0923-5965(15)00005-3
DOI: http://dx.doi.org/10.1016/j.image.2015.01.004
Reference: IMAGE14921

To appear in: *Signal Processing: Image Communication*

Received date: 9 January 2014
Revised date: 8 January 2015
Accepted date: 8 January 2015

Cite this article as: Jerónimo Mora Pascual, Higinio Mora Mora, Andrés Fuster Guilló, Jorge Azorín López, Adjustable compression method for still JPEG images, *Signal Processing: Image Communication*, http://dx.doi.org/10.1016/j.image.2015.01.004

Adjustable Compression Method for Still JPEG Images

Jerónimo Mora Pascual, Higinio Mora Mora, Andrés Fuster Guilló, Jorge Azorín López

Specialized Processors Architecture – Laboratory
Department of Computer Science Technology and Computation
University of Alicante
Alicante, Spain

Abstract:

There are a large number of image processing applications that work with different performance requirements and available resources. Recent advances in image compression focus on reducing image size and processing time, but offer no real-time solutions for providing time/quality flexibility of the resulting image, such as using them to transmit the image contents of web pages. In this paper we propose a method for encoding still images based on the JPEG standard that allows the compression/decompression time cost and image quality to be adjusted to the needs of each application and to the bandwidth conditions of the network. The real-time control is based on a collection of adjustable parameters relating both to aspects of implementation and to the hardware with which the algorithm is processed. The proposed encoding system is evaluated in terms of compression ratio, processing delay and quality of the compressed image when compared with the standard method.

Index Terms:

Image representation, Compression, Real-time, JPEG, image quality assessment

1. Introduction

Image compression methods play an important role in many applications with limited resources for viewing, storage and processing. A framework in which these limitations are obvious is the development of applications for the internet and mobile devices such as phones or PDAs. In this context, developing interfaces aimed at the end user is strongly conditioned by the time it takes to download and view multimedia data. These aspects led to research into ways to improve compression rates and the quality of multimedia elements such as pictures, video or audio clips in order to speed up user interaction and take advantage of the heterogeneity of the available infrastructure [1, 2, 3].

As a result, it is appropriate for several applications to have control over the data processing time to adapt available resources to the needs of each moment. The priority in these applications is to be able to adjust to timing constraints that indicate the precise moment in which data should be available either for the user's viewing or as input to another process. A good example for this issue is navigation systems, where images from the digital map must be updated constantly and where it is required to obtain the map image in time (*timeout*). In these applications, providing the map in the required time takes priority over obtaining a detailed comprehensive record of it. It should be of sufficient quality to enable the user to distinguish objects of interest on the map image so as to be able to use the service [4].

One scenario that is becoming increasingly relevant for real-time still image compression is displaying and viewing websites. Some of the most popular sites (Flickr, Google Streetview, Panoramio, Yellow Pages, etc.) base their business on providing graphic content coded as still images (photos, maps, banners, etc.). This fact, combined with the diversity of connected devices and of available bandwidths (PCs, workstations, PDAs, cellphones, etc.), makes it advisable to tailor a service to each user profile that can adjust the quality, size and viewing speed of images. In Real-Time Systems (RTS), controlling the runtime is a central aspect of their design. Ensuring that real-time applications operate correctly depends not only on the accurate logic of the algorithms and of the results they give, but also on when results are available [5, 6, 7]. In recent years, computer architectures have been presented that tackle the issue of real-time from the hardware layer [8, 9].

One common technique for approaching RTS design is the *imprecise computation model*, which was presented formally by J. Lin in 1987 [10, 11]. The strategy of this model is based on solving transitional overload by reducing the execution of the optional part when there are insufficient resources. It is true that this affects the quality of the result, but it will always remain acceptable if the mandatory part has been completed.

The known methods of compression that take the temporary aspect into consideration for constructing images, allow the time needed to be adjusted for a complete viewing of an image through progressive image transmission (PIT) [12], such as the Progressive JPEG (PJPEG) [13, 14] and fractal compression [16, 17].

All these methods are mainly aimed at minimizing the storage space of still images, and offer a sufficient quality for the viewer. It is in this last aspect where they exchange time and storage space for quality of image.

However, these methods do not tackle the problem of real-time itself, where automatic control is needed over the processing time and the quality of the result, depending on the needs of the application. The non-predictability of the response time limits the usefulness of the work carried out to provide the result with the least possible delay in complying with the usability of the application itself, either due to hardware requirements or to users themselves [18, 19].

For video stream, the proposals adapt the bitrate of the requested stream to changes in network conditions, by switching between different video representations [20, 21]. However, up until recently, there were not many applications in which real-time JPG solutions looked attractive. With the popularity of web browsers running over slow modem links, and with the ever-increasing performance of personal computers and mobile devices, an adjustable encoder/decoder for still images has become an interesting option for web applications [15].

To respond to these applications, this paper proposes a method of coding still images based on a JPEG algorithm that includes flexibility in the processing time when compressing/decompressing images. The adjustment mechanisms are based on the predictability of the execution times of the algorithm's modules, and on applying the design principles of imprecise computation that allow for partial execution.

2. Real-Time Image Compression Model

The JPEG algorithm (Joint Photographic Experts Group), which is formally known as ISO / IEC IS 10918-1 | ITU-T T.81 [22], is the current standard for compression and decompression of still images and continuous tone. In this paper, we have chosen the JPEG format as the baseline image encoder/decoder, rather than other compression algorithms such as JPG2000, as it is the most popular lossy compression method that uses the human viewing scheme to discard a lot of the image's original information while maintaining the perceived quality [23]. Another reason is the controversy to do with patents for some parts of the JPEG2000 algorithm [24]. As a result, JPG2000 is not widely supported in web browsers, and is not therefore generally used on the World Wide Web, while the standard JPEG compression algorithm has been widely adopted as an efficient electronic way of storing and transmitting images.

Traditional image compression algorithms reduce the spatial cost of images by reducing the amount of data that represent them. Usually this compression is accompanied by some loss of quality. Some of these compression methods allow users to act on the compression ratio of the image, which relates quality to the size of the compressed image.

These compression algorithms only consider the possibility of specifying how much we want to reduce the binary size of the image with the compression ratio parameter, without ever taking into account the processing time needed to do so.

The approach suggested in this paper considers environmental conditions in regulating the processing time of the algorithm. The principle of imprecise computation will be applied to fragment the compression process of the image in several mandatory and optional subtasks. The optional tasks can be partially completed during processing and will only be run when there is enough time. In this way, although the processing exceeds the full time available, it is possible to obtain results within the timing limit imposed.

The behavior of the method proposed will be defined by a collection of control parameters of the algorithm ($\vec{d} = <d_1, d_2, \ldots, d_t>$), which have an influence on processing time and quality. These parameters adjust the compression time and image quality depending on the restrictions imposed by the application. The proposed model uses a method like the one shown below:

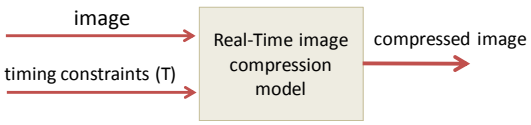


Fig. 1. Proposed RT image compressor outline

However, to determine the right timing constraints for each application and each scenario, it is necessary to consider the characteristics of the environment, as they affect the image coding and decoding. These aspects, named as environment conditions (ECs), will set the quality and processing time needed to provide the right

degree of satisfaction for the user. This requires a background study into these conditions and analysis of all cases to assess the right time and quality for each one. For example, for the scenario described in the introduction of viewing web pages with several still images on connected mobile devices, the operating conditions could be as follows: number of images on the page required, images size, screen size, connection bandwidth, performance of the device, etc. Studying these conditions will determine certain time restrictions that will affect the compression, transmission, and decompression processes.

Each application will have to code its own environment conditions to produce the time limit (T) by which each image should be available. This process carries out a function that relates a time-bound T to the environment conditions (ECs). Thus, to meet these timing constraints, the model must incorporate a time control unit that receives the maximum processing time as a parameter, and produces the value of the control parameters as a result. Figure 2 shows the general outline.

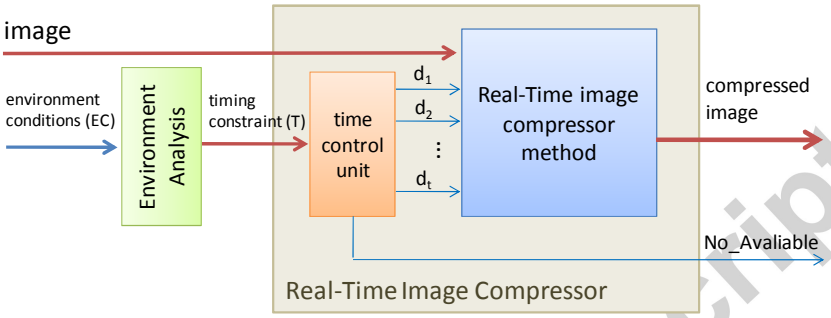


Fig. 2. General outline of the RT image compression model

This time control unit determines the real-time behavior of the compression method and makes the algorithm finish on time, and should continue to determine response times for the whole system. Its purpose is to determine and select the real-time behavior for each stage of the algorithm, establishing the maximum number of permissible iterations for the algorithm and set up the number of mandatory and optional steps.

As mentioned previously, in some applications it may be convenient adjust the size and quality of the images to the operating context. As example of operation of the proposed system depicted in previous figure and of adjustable compression applications, a "real-time" navigation system can be considered. In this application it may be convenient adjust the size and quality of the images to the operating context in order to maintain an acceptable service to the user. Nearly all cell phones on the market now include some sort of GPS capability therefore it is common using navigation systems apps over them. However, since there are multiple system and multiple GPS manufacturers, the operating conditions will most likely not be the same for every one of them. The performance perceived by user is depending on both cell-phone system capabilities (eg. processor, memory, battery) and environment conditions (eg. satellite coverage, user speed). So that, real-time features can be included into the system to provide an homogeneous quality of service. The "*Environment Analysis Module*" checks the system and environment conditions and deduces from them the 'T' constraint for the "*Time control unit*". In this way, several scenarios could occur: for high-speed user movement that requires frequent map-image refresh, a variable T time can been set to allow the system to compute the maps in accordance with system, user speed and necessary refresh rate; for low system capabilities or weak satellite coverage, the image process can be made with fewer data in order to maintaining response times; or finally for example, low battery modes of the system can make advisable setting appropriate T time in order to show the images with the least possible computation.

We can use different strategies to implement the *environment analysis module* and *time control unit* that safeguard predictability in the response times of the whole system. We can construct the controller by means of combinational logic. The circuit inputs will be the coded values of the environment conditions or the T value and will provide the T value or the value of the real-time parameters as outputs. When the length and number of inputs is low, we can use a look-up table with pre-calculated data that can select the value of the outputs directly from the inputs. The actual value of the inputs into the table will be used to address it. This second option has the advantage of being able to reconfigure the modules to adapt to new situations by simply updating the table contents.

If the overall constraint cannot be met, even with minimum delay in all the parameters, the *Not_Available* signal is activated. When this occurs, the application must decide what action to take: either to reject the compression or to obtain the shortest time compressed image achievable.

The environment analysis module checks the environmental conditions that determine the constraints of execution. Those conditions refer to the system and the operating context. For example, table 1 describe some of most common environmental conditions for general-purpose applications. Those conditions will be tabulated

and integrated in the environment analysis module in order to determine the more appropriate T value, to satisfy the conditions.

Table 1
Examples of most common environmental conditions

Environmental condition	Brief description
System	*Related to hardware.*
Hardware processing capabilities.	Performance of the hardware.
Memory available.	Memory available to store the image.
Battery mode.	Acceptable battery consumption for processing.
Operating context	*Related to the operating context and applications*
Coverage.	Mobile coverage when applicable.
Critical level.	Critical level of the application related to meet time constraints.
Terms of use	Specific conditions of use of the system.

2.1. Real-time parameters

In the following paragraphs, the image compression/decompression algorithms and hardware are analyzed to identify those features which affect the quality and processing time. For the application level, we will study the effect caused by calculating the coefficients for the DCT and the spatial subdivision of the images. On the hardware level, we will analyze the impact of accuracy on the representation of the operands on the performance of and benefits for the compression/decompression process. The objective in the latter case is to design specific systems that adjust the computer architecture to the algorithms that they should run on. These systems can be embedded in other more general systems and work as a dedicated processor or as a coprocessor to run their specific tasks. Subsequently, in the empirical evaluation section, the relationship between the value of each parameter and algorithm behavior will be determined in order to identify the variation range.

2.1.1. DCT coefficients

DCT coefficients involve the spectral components of the image. The coefficients with a higher frequency represent the parts of the image with most detail. Removing the DCT coefficients in the encoding of the image affects the degree of detail with which it can be seen. However, for a given resolution, the detail of the images is generally higher than the human eye can perceive, so if they are removed after carrying out the IDCT, the loss of detail will not be perceived by a human observer and it may be assumed by certain applications [23].

The number of DCT coefficients that are calculated is in proportion to the time delay of this stage of the algorithm. So, the number of coefficients is a candidate for adjusting the time of the processing method to the existing restrictions.

Depending on whether the timing constraints are in the compression or the decompression stage, the following situations can arise:

- Unlimited compression and decompression time: the complete DCT and IDCT are calculated for the two processes. This is the standard method. For example the compression process made by a camera when takes a JPEG photo and the decompression process when it is displayed later in a PC by user.
- Unlimited compression time and limited decompression time: the complete DCT is estimated for the compression, but the decompression process rejects the coefficients with the highest frequency. For example the compression made by satellite's high resolution camera and the decompression made by navigator App on a standard phone where user is moving.
- Limited compression time and unlimited decompression time: only the maximum number of DCT coefficients that allow the compression to be carried out in the time available is calculated, and in the decompression the reverse process is performed with the existing coefficients. For example the compression process made by an ultra-high speed burst shooting of a camera (our proposal should allow to increase the rate of still images per second) and the decompression process when it is displayed later in a PC by photographer.
- Limited compression and decompression time: only calculates the maximum number of DCT coefficients that allow the compression to take place within the time available. In the decompression, coefficients can continue to be deleted, if necessary, or all of them can be used if time permits. For example the compression made by a street view capturing system on a car (like Google street view) when it is moving along the city the decompression made by City Map App on a standard phone where user is moving for a stroll in the city.

The parameters that indicate the amount of coefficients that are calculated according to the available time are $d_1 \equiv n_1$ for the DCT in the compression process and $d_1 \equiv n_2$ for IDCT during decompression.

2.1.2. Space subdivision

The basic idea of this technique is to build a set of D subimages by uniform sampling of the original image in order to adapt the processing time. In this way, each subimage represents a stripped-down version of the original image, whose size is D times smaller than the size of the original image. A very shorter processing time is needed to compute each individual subimage and, consequently, allows to adjust the whole processing time. Therefore, this subdivision helps to achieve the timing constraints, because successive processing introduces flexibility into the response times and into the quality of the result. Thus, the compression of these sub-images can be performed separately, and while it is being decompressed, the information from each sub-image is being added to the resulting image until it obtains the greatest quality allowed by the time available. In this approach, if only a short time is available, it is possible to have an image of lower quality but close to the original appearance.

The value D is a design parameter of the system that will determine the capacity to adjust to the time constraints. The larger the value of D, the greater the flexibility for the system. When considering the whole image, the value of D is equal to 1. To include space subdivision in the basic algorithm, a value D of sub-images needs to be established. Only a subgroup of these will be used for the compression or decompression of the complete image. The parameter $d_2 \equiv N_1$ indicates the number of sub-images that will be used in the compression task.

The number of D subimages will determine the adjustment capacity of the method applied in this technique. The larger D is, the more flexibility that the time control unit will have to meet the more demanding timing requirements. However, the partial use of the set of subimages must be accompanied by the method of filling the empty gaps, using the nearest pixels. Therefore, the selection order of the images provided must complete the information of the complete image, using evenly spaced pixels. The following table indicates the selection order of these subimages for different values of D. The subimages will be coded and transmitted in this order shown which will assist in the partial processing stage.

Table 2
Selection order of subimages in partial processing

Number of subimages (D)	Generation order
4	1, 2, 3, 4
8	1, 5, 3, 7, 2, 6, 4, 8.
16	1, 9, 5, 13, 3, 11, 7, 15, 2, 10, 6, 14, 4, 12, 8, 16
32	1, 17, 9, 25, 5, 21, 13, 29, 3, 19, 11, 27, 7, 23, 15, 31, 2, 18, 10, 26, 6, 22, 14, 30, 4, 20, 12, 28, 8, 24, 16, 32

In the real-time decompression task, the number of subimages needed to reconstruct the original image depends on the available time. The highest quality image will be obtained using all the subimages coded in the compression process. But, if necessary, an approximated image will be obtained by using a smaller number of coded subimages. The parameter $d_2 \equiv N_2$ will be used to indicate the number of subimages used in the decompression.

2.1.3. Operand precision

The temporal complexity of an algorithm can be reduced by simplifying the mathematical operations included. This technique basically consists of applying the concepts of the imprecise computation at the hardware level [8], [10]. A native flexible processing of these functions in terms of time and precision is achieved by acting at the hardware level. Imprecise computation is therefore a very useful tool for adjusting the temporary execution cost of the algorithm to the temporary boundaries.

The precision of the arithmetical operations used in running the instructions has a direct influence on the time that the processor needs to compute this calculation [8]. One strategy to incorporate flexibility into the processing is to adapt the operand length to the time available for the compression task. The precision is reduced while a time constraint is established. So, the smaller precision and length of the operand, the less instruction delay. This reduction in time depends on the cost of the instruction, and will be different for each operator.

In the JPEG compression method, the most significant arithmetic operations are addition and multiplication. The parameter that determines the precision of the operator in each operation is a vector indicating the number of significant fractional digits for the operands in the arithmetic instructions of addition (p_a) and multiplication (p_m): $d_3 \equiv \vec{p} = <p_a, p_m>$.

To incorporate this characteristic into the compression model, the execution of traditional arithmetic operators must replace the original hardware in order to obtain the profit for time flexibility. This low-level flexible processing can be used in all the modules of the method. As in the previous paragraphs, different precision limits can be established for the tasks of compression ($\vec{p}_1$) and decompression ($\vec{p}_2$).

2.2. Real-time compression/decompression algorithm

The compression/decompression model proposed here is summarized in the diagram shown in figure 3. The output control parameters $\vec{d} =< d_1, d_2, d_3 >$ of the time control unit are related to each real-time characteristic identified in the previous section. Thus, $<\vec{d}>$ consists of three parameters: the number of DCT coefficients (n) to compute, the number of subimages (N) that the image breaks down into, and the operand precision needed to compute the mathematical operations ($\vec{p}$).

$$\vec{d} = <n, N, \vec{p}> \quad (1)$$

On the basic JPEG sequential-compression model described, the modifications specified in the previous section will be added to satisfy the real-time constraints established by the time control unit.

Figure 3 shows the complete diagram for the compression phase and the data flow between each stage. The control parameters for this phase are defined by the vector $\vec{d}_1 = <n_1, N_1, \vec{p}_1>$. As can be seen, the transformation operations computed by the JPEG real-time algorithm are as follows: image pre-processing, space subdivision, colour space conversion, colour subsampling and DCT calculation.

As a result, the algorithm gains a collection of the N_1 compressed subimages $\{fc_1,\ldots,fc_{N1}\}$ placed adjacently in the order set out in table 1.

For the decompression task, this method includes the task of computing, in reverse order, the opposite operations to those involved in the compression task. The time control unit will also establish the optimum value of the parameters included in the algorithm satisfying the timing constraints in the decompression phase. These parameters, defined by the vector $\vec{d}_2 = <n_2, N_2, \vec{p}_2>$, will influence the normal running of the operations in this phase to adjust the time and the decompressed image quality to fit the requirements of the application.

The adapted real-time operations of the algorithm for the compression and decompression tasks are described below.

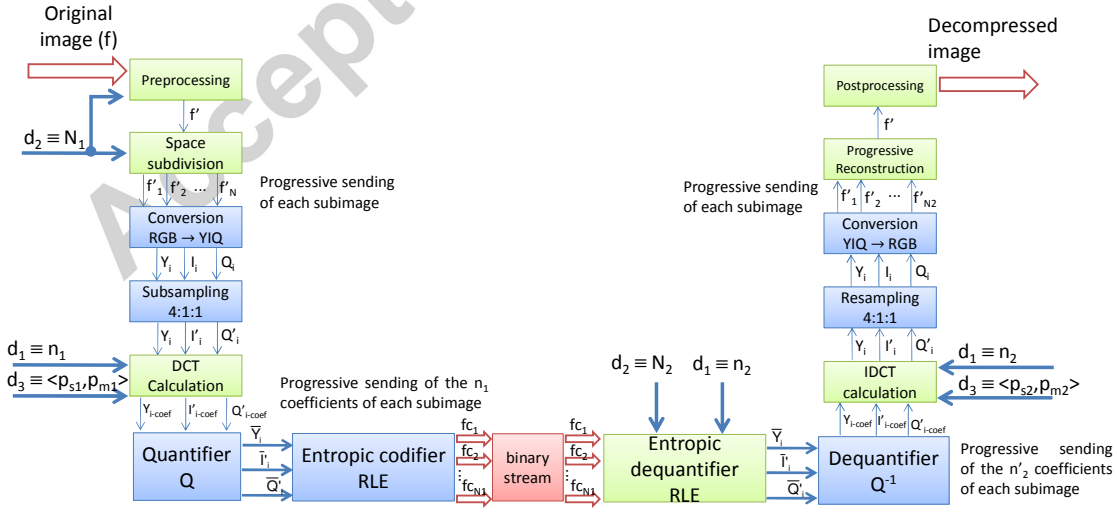


Fig. 3. Real-time image compression/decompression

2.2.1. Image preprocessing

The aim of this stage is to prepare the image for its subsequent division into subimages. This transformation consists of adapting the size of the initial image to the processes that will be run later in terms of compression

and real-time management. By resizing, the optimum size will be sought for the space subdivision of the image, so as to create the blocks for the DCT and the chrominance reduction of the colour model used. To do this, once the *jumps* corresponding to the value of *N* have been calculated, the size immediately above the size of the original image that allows these three processes to occur will be computed.

Resizing the image involves adding additional rows and columns to help the processing. Thus, the resized image must have a number of rows divisible by *x-jump*, 8 and 2, and a number of columns divisible by *y-jump*, 8 and 2.

This stage must be carried out completely, regardless of the timing constraints and the time control parameters. According to the model of imprecise computation, it is a mandatory stage.

2.2.2. Spatial subdivision

Once the jumps in each size and the coordinates of the first pixel for each subimage are known, the resized image is divided into the D subimages.

The control parameter *N*, with N ≤ D, will indicate the number of these subimages that must be built to satisfy the constraints of the application. As a result of this stage, a sequence of D-sized subimages will be obtained. The size of each subimage is N times inferior to the size of the resized original image.

Each subimage must be processed independently of the others. There are different design alternatives for processing the collection of subimages, depending on the degree of parallelization and segmentation of the rest of the algorithm stages. In this paper, a sequential processing of the subimages will be used that favours progressive processing. Thus, there is no need to wait for all the subimages to pass for a block to start running the next block. As a result, it is not necessary for all the processed subimages to have concluded; the first subimage just needs to have finished providing some useful information while more detail is added by the subimages arriving afterwards.

2.2.3. Discrete Cosine Transformation (DCT)

The DCT stage is independently applied to the Y, I' and Q' colour components of each subimage successively. Each component is divided up into 8×8 blocks where the DCT 2D will be applied. First the DCT of each row is obtained, and the DCT is then calculated for each column. The calculation for this transformation uses the well-known Vetterli and Ligtenberg algorithm [25].

In this transformation, an adjustable delay is incorporated by controlling the DCT coefficients required by each 8×8 matrix, whereby introducing parameter *n* as an input of the time-control unit. The value of this parameter will indicate how many coefficients must be calculated to satisfy the application's timing constraints. This value can range from 1 to 64, where 64 is the normal running of the basic method. Thus, for a value n_1, only the first n_1 coefficients of the 8×8 matrix will be computed according to the known zig-zag route. To do this, the 1D DCT is computed for all rows and only the DCT of the necessary columns is processed to obtain the right coefficients.

As a result of this stage, a linear n_1-sized vector with the mandatory coefficients is obtained (Fig. 4).

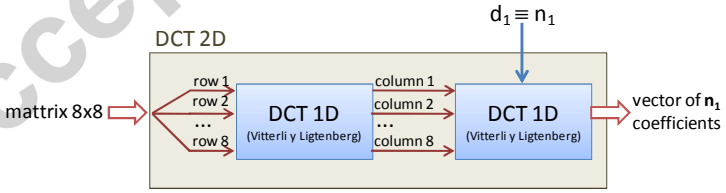


Fig. 4. Discrete Cosine Transformation

Once the coefficient vector of each block is obtained, they are quantified and coded to minimize the space cost, as in the basic method.

2.2.4. Decoding

This is the first task of the image decompression step. In this stage, the time control parameters n_2 and N_2 are actuated. This stage begins with the set of N_1 compressed subimages as per the JPEG compression diagram and generates the decoded vectors coding the coefficients corresponding to each subimage.

The parameter N_2 indicates how many subimages must be decompressed to meet the requirements of the problem, whereas n_2 indicates the number of coefficients that must be completed for each vector.

The quality of the image generated will be limited by the information provided by the compression step. Therefore, applying these restrictions will depend on the information used at that stage. If the time parameter

values of this phase are equal or higher to the ones used in the compression, an additional loss of information will not occur. However, lower values will cause any subimages and components of the coefficients vector for which there is not enough time to be calculated, to be discarded. The subimages that are not added are the last ones in the data sequence of the compressed image, whereas the discarded coefficients are the ones with the largest frequencies to maintain the maximum subjective quality allowed by the time available. The positions of the discarded coefficients will have a value of zero.

2.2.5. Inverse Discrete Cosine Transformation (IDCT)

The IDCT calculation is computed once the dequantification step has been performed. For this, groups of *n* elements will be taken and an 8×8 matrix is formed with each group, considering the zigzag order followed by the compressor. The IDCT is computed for each matrix by means of the same pattern used for the DCT, i.e. the IDCT 1D is first calculated for the rows, and then for the columns. The control parameter n_2 indicates the number of coefficients of each vector being processed in this stage.

The IDCT block output has all the 8×8 matrices that make up the Y, I' and Q' components of each subimage.

2.2.6. Progressive rebuild and post-processing

Depending on the time available, the rest of the subimages will be decompressed until a maximum of N_2 subimages. Finally, when the process of decompression has finished, the original image is composed using the subimages that have been decompressed.

To rebuild the resulting image, the subimages are incorporated progressively, with more information obtained with each subimage added. Image quality is thus improved gradually. The process ends when the final image is made up of all the subimages that have decompressed in the time available.

The image is reconstructed with the criterion of obtaining the best possible subjective quality. Thus, the gaps of blank pixels that correspond to non-generated subimages will be covered by adjacent pixels according to the criterion of the nearest pixels. As the number of provided subimages is known, the reconstruction process builds the original image by placing the pixels of the subimages or their duplicates. As mentioned in the previous section, average neighbouring pixel values can be placed to alleviate the aliasing effect that is caused.

Finally, the resulting image is resized to its original dimensions, as it had been resized during pre-processing to ready it for the subsequent processing.

3. Empirical Evaluation

This section studies the behaviour of the real-time compression method with regard to the value of the control parameters and the following features: compressed-image quality, processing delay time and compression rate (*cr*).

The empirical experiments are aimed at evaluating the compression phase of the method since the suggested real-time decompression step uses the same principles of cost reduction and image quality to adjust the delay of this phase to the timing constraints. To evaluate the compressed image quality, the image is compressed and then decompressed completely, that is, without applying any temporary adjustment used in the decompression.

There are several quality assessment methods for digital images [26-33]. In this work, the quality will be measured both by objective and by subjective evaluation. The objective quality is evaluated in five ways using new recent measures: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Feature Similarity Index (FSIM), Fourier Transform-Based Image Quality Measure (Q) and Singular Value Decomposition based (SVD) method. In all quality methods, the RGB color image is converted to gray scale with "*rgb2gray*" Matlab function.

The PSNR value is a mathematical measure of image quality based on the pixel difference between two images. The PSNR used is defined in (2) and (3).

$$MSE(f, f') = \left[\frac{1}{MN} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left(f(x,y) - f'(x,y) \right)^2 \right]^{1/2} \quad (2)$$

$$PSNR(f, f') = 10 \log \left[\frac{MAX^2}{MSE} \right] \quad (3)$$

where MSE is the *Mean Square Error* of a colour component, M and N are the image's dimensions, $f(x, y)$ is the value of the pixel (x, y) in the original image, $f'(x, y)$ is the value the pixel (x, y) in the compressed image and MAX is the maximum possible pixel value.

The Structural Similarity is a metric for measuring the similarity between two images [26-29] that can be viewed as a quality measure of one of the images being compared with the original one [26]. The mathematical expression for calculating SSIM is:

$$SSIM(f,f') = \left[\frac{(2\mu_f \mu_{f'} + c_1)(2\sigma_{ff'} + c_2)}{(\mu_f^2 + \mu_{f'}^2 + c_1)(\sigma_f^2 + \sigma_{f'}^2 + c_2)} \right] \quad (4)$$

where μ_f is the average of f; σ^2_f is the variance of f; $\sigma_{ff'}$ is the covariance of f and f'; and c_1 and c_2 are constant used for to stabilize the division with weak denominator (eg. c1 = 0.001 x 255^2 and c2 = 0.003 x 255^2).

The Feature Similarity (FSIM) compares the low-level feature sets between the reference image and the distorted image. The expanded formulation of this index can be found in [33]. The Fourier Transform-Based Image Quality (Q) utilizes the phase and magnitude of the Fourier transform of the image [30]; this metric uses the phase and magnitude together to compute visual quality and they prove the scalability of the proposed method in a reduced-space representation. Finally, the Singular Value Decomposition-based (SVD) calculates the quality of images based on its distortion level. The full mathematical formulation can be found in [32].

3.1. Experimental setup

The test images are coded in the RGB colour model with a colour depth of 24 bits per pixel. The images are selected so that they collect as varied content characteristics as possible, with homogeneous and bordering regions. The test set has five images. One of them is the example image depicted in this paper that has a resolution of 240x192 pixels (96 ppp) and 24-bit colour depth.

The algorithms have been coded in Matlab® 2014a due to the tools and libraries incorporated for image processing. The experiments have been performed using a 2 GHz Intel Core i5 and 8 GB of RAM memory.

The simulation process aims to evaluate the performance of the adjustable compression method in contexts conditioned by different operating restrictions. To that end, a set of tests has been scheduled to measure the compressed image quality, compression delay and compression rate for each control parameter (number of coefficients, number of subimages and operand precision. The tests have been performed over all images in the test set and the results shown in the following sections are the average of the obtained results. The experiments are designed to analyse how the value of the control parameters influences the cited characteristics. Firstly, the relationship of each parameter is studied independently to the others so as to determine their isolated effect; next the effect of combining them is then analysed.

This phase is the basic use of the control unit described in this method, as it allows for an analysis of the effect that using the model has on images and determines the appropriate quality/cost for each application. The information obtained in these experiments gives us the knowledge to calibrate the exact value of the parameters that relates them to the timing constraints of each scenario. These relationships will be taken into account for the construction of the time-control unit of the system.

To evaluate the scope of the results obtained, the reference value of the JPG standard algorithm and the graphic data for each studied characteristic are shown together.

3.1. Number of coefficients DCT (n)

In this experiment, the behaviour of the compression model is analyzed for different values of the first control parameter $d_1 \equiv n$. This parameter indicates the number of DCT coefficients calculated, and can take a value of between 1 and 64. The values for the other parameters providing their greatest quality will be maintained, and will correspond with the values specified in the JPEG standard method: $d_2 \equiv N=1$, with D=1 and $d_3 \equiv \vec{p}$ = simple precision.

3.1.1. Compressed image quality

Compressed image quality is a basic aspect for some applications to run correctly. To evaluate the convenience of the real-time compression method, the loss of quality caused by the influence of the control parameters needs to be tested. To do this, we used objective measurements of the PSNR, SSIM, Q and SVD between the original image without compression and the successive images compressed for increasing values of n.

The measure obtained according Fourier Transform-based method (Q) consists of two pieces: phase (Q_{phase}) and magnitude ($Q_{mag.}$) of the transform [30]. In our experiments, the combined value of both measures is not calculated because of the need for a training process to calculate the weights and constants of their combination by solving optimization problems [30] that goes beyond the scope of this work.

The following figure shows the evolution of the PSNR and SSIM calculation for compressed images with different numbers of DCT coefficients, and table 3 shows full comparison with all objective methods mentioned.

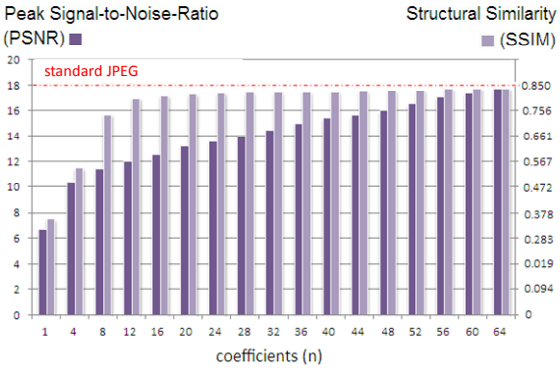


Fig. 5. Objective quality of the compressed image with regard to the number of coefficients

Table 3
Objective quality metrics of the compressed image with regard to the number of coefficients of the DCT.

	Coefficients (n)																
	1	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
PSNR	6.70	10.41	11.46	12.04	12.55	13.22	13.61	13.97	14.47	15.05	15.44	15.68	16.02	16.53	17.07	17.40	17.78
SSIM	0.34	0.47	0.75	0.81	0.82	0.825	0.83	0.832	0.832	0.833	0.835	0.836	0.838	0.841	0.847	0.849	0.850
FSIM	0.631	0.875	0.905	0.927	0.931	0.944	0.948	0.951	0.952	0.955	0.956	0.957	0.957	0.957	0.958	0.958	0.958
Qphase	0.934	0.939	0.945	0.947	0.949	0.950	0.950	0.950	0.950	0.950	0.950	0.951	0.951	0.951	0.951	0.951	0.951
Qmag.	0.953	0.963	0.970	0.974	0.976	0.977	0.978	0.978	0.978	0.979	0.979	0.979	0.979	0.979	0.980	0.980	0.980
SVD	66.04	33.54	25.94	20.27	20.19	20.36	19.62	19.75	20.68	17.07	15.05	14.36	14.67	12.10	9.43	7.50	6.92

It clearly shows that the more DCT coefficients are discarded, the greater quality loss. However, the human visual examination does not detect this loss until too many coefficients are discarded, as also shown by the SSIM, FSIM and Q objective measurements. The quality results in Table 3 show that the Fourier Transform-based method in [30], Qphase and Qmag values, works in DCT similar to Fourier domain and it is scalable in a reduced space representation of the image. This is a good advantage for quality computation and for prediction performance of the compressed image in real time processing. Figure 6 shows examples of images obtained for certain coefficient values. These examples show a progressive degradation of the image when *n* decreases, but small values of *n* are accepted with a minimum loss of quality.

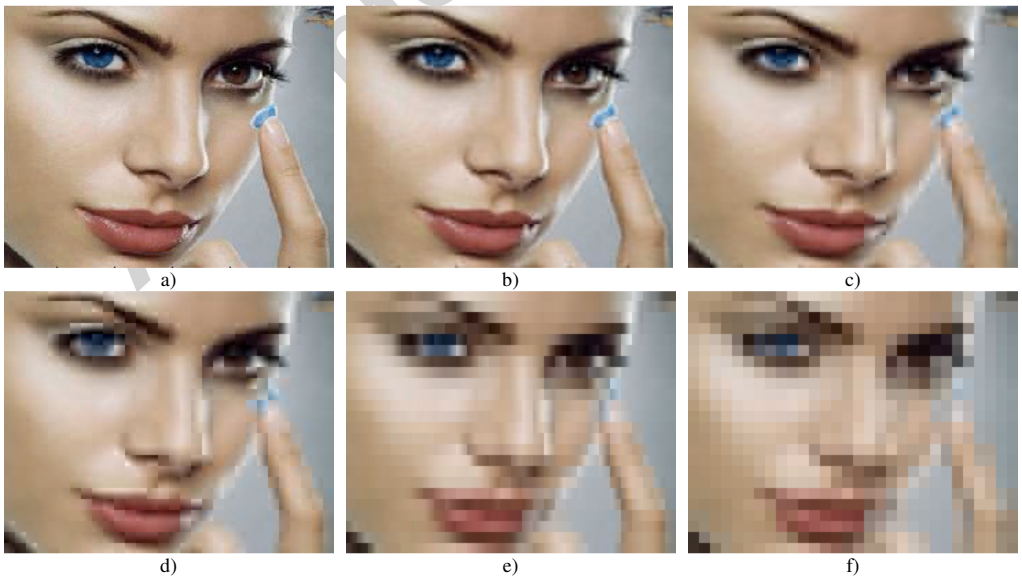


Fig. 6. Subjective quality of the compressed image considering the number of coefficients:
a) Original image, n=64 b) n = 16; c) n = 4; d) n = 3; e) n = 2; f) n = 1

3.1.2. Delay estimations

The relationship between processing time and the number of coefficients will determine the adjustment capacity of the method when this parameter is manipulated. Figure 7 shows the evolution of the time cost in relation to the n parameter in the compression process. The data shown relate to the cost of the method working at the highest performance level, that is, with maximum quality, n=64. The time cost drops with the number of calculated coefficients, in a slightly linear trend. The delay drops to less than 20% of the total cost. However, the relevant aspect is the flexibility introduced by this parameter in the delay of the algorithm.

In its relationship with the standard method, a slight worsening in the cost is observed when all the coefficients are considered (n=64), due to the stage introduced in the pre-processing for the space subdivision and to the cost of temporary management.

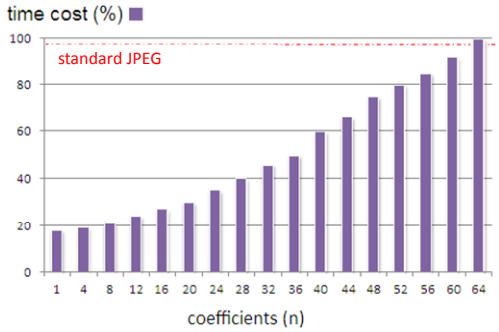


Fig. 7. Time cost of the compression method with regard to the number of coefficients

3.1.3. Compression rate

The third aspect to study is the influence of the number of calculated coefficients on the compression rate. In developing this experiment, we considered a constant compression factor similar to the one used in the standard JPEG algorithm.

The data obtained express an inverse relationship between the compression rate obtained and the number of coefficients. As can be seen in figure 8, for small values of n the compression rate increases considerably, whereas it remains stable once it reaches a certain threshold, close to the normal value of the standard method.

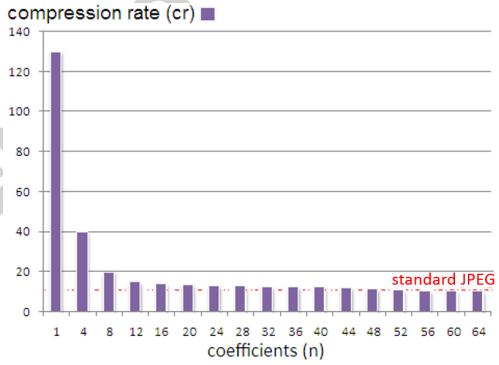


Fig. 8. Relation between compression rate and the amount of coefficients used.

3.2. Number of subimages (N)

The space subdivision suggested in this paper also introduces flexibility into the study characteristics. To analyze the scope and the effects caused by the number of subimages in the compression method, the combined effect of parameters D and N were analyzed. Parameter D is the number of subimages into which the original image is divided, and N is a subset of them that is used for the compression. The remaining control parameters are maintained at their standard value, with the necessary precision and the whole set of DCT coefficients, specifically $d_1 \equiv n=64$ and $d_3 \equiv \vec{p}$ = simple precision.

3.2.1. Compressed image quality

The effect caused by dividing the image into D subimages for the compression task was analyzed. In this test, all the subimages were compressed and then used to reconstruct the original image, that is, N=D. The figure 9 depicts the objective image distortion measured with PNSR and SSIM measures.

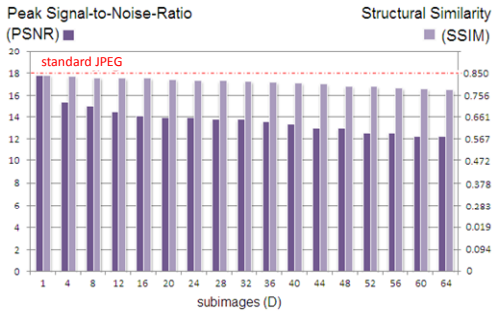


Fig. 9. Objective quality of the compressed image with regard to the number of subimages (D)

Even though all the subimages were used for the reconstruction, greater distortion occurs as the number of subimages dividing the original image increases. The more subimages are generated, the more correlated are the pixels, and the greater the distortion. Although the DCT considerably reduces the block effect for adjacent pixels, the technique of building the subimages from the evenly distributed pixels in the original image makes this distortion visible, but does not overly affect the image structure. The Table 4 show quality results for all measures.

Table 4
Objective quality of the compressed image with regard to the number of subimages (D)

	Subimages (D)																
	1	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
PSNR	17.78	15.44	15.05	14.47	14.14	13.97	14.00	13.80	13.80	13.61	13.42	13.01	13.01	12.55	12.58	12.30	12.30
SSIM	0.85	0.849	0.843	0.84	0.841	0.835	0.830	0.829	0.825	0.824	0.820	0.815	0.806	0.805	0.798	0.794	0.792
FSIM	0.958	0.955	0.954	0.953	0.953	0.952	0.953	0.952	0.952	0.952	0.952	0.952	0.952	0.952	0.951	0.952	0.951
Q_{phase}	0.951	0.950	0.950	0.950	0.949	0.950	0.950	0.948	0.950	0.949	0.948	0.950	0.950	0.950	0.950	0.949	0.949
$Q_{mag.}$	0.980	0.978	0.987	0.979	0.978	0.978	0.979	0.978	0.979	0.979	0.978	0.979	0.978	0.979	0.979	0.978	0.978
SVD	6.92	8.42	10.71	11.22	11.45	9.11	10.68	9.028	10.28	10.32	11.08	10.20	10.48	11.40	10.42	12.35	12.42

Once the value of D was set, we analyzed the effect that an incomplete reconstruction due to the time constraints has on the compressed image. For this, the image was decompressed again by taking all the subimages used in its compression. Figure 10 shows the objective error made with the partial reconstruction using PNSR and SSIM metrics and it illustrates an upward trend in the increase in quality as the number of subimages goes up. These data complement the results of figure 9, showing that a better image is obtained with less subdivision. However, the quality perceived by a human observer remains acceptable for a few subimages.

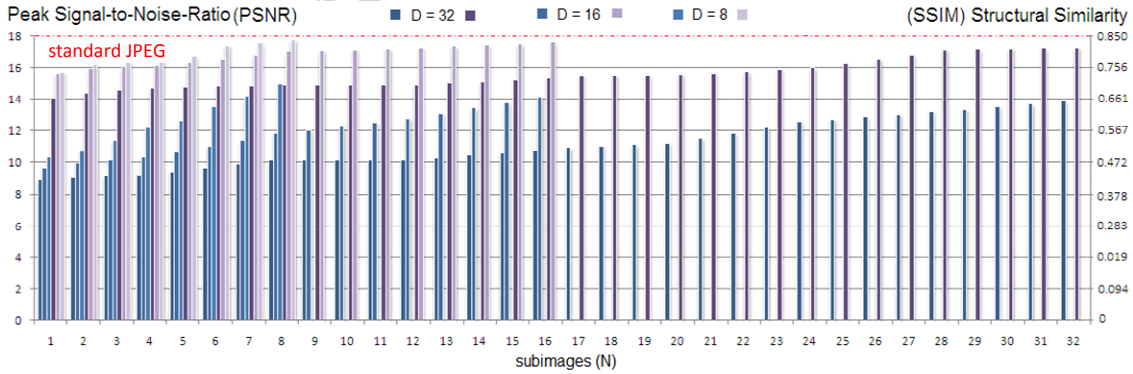


Fig 10. Objective quality of the compressed image with regard to the number of subimages (N)

3.2.2. Delay estimations

The space subdivision technique suggested in this paper provides great flexibility in processing the compressed image, although it introduces an after cost of the pre-processing and post-processing stages to perform the initial subdivision of the subimages and the final reconstruction from them (figure 11). This cost carries little weight in the total delay of the method.

As in the previous step, the D quantity of subimages must be established. Only the ones indicated by the time control parameter N are used in the reconstruction. The larger the D value, the greater the flexibility of the

control unit to adjust the time to the problem requirements, while obtaining worse quality in the decompression. Therefore, a trade-off occurs between the two aspects that help to form the DCT 8x8 blocks. In our experiments, a value of D=32 was used.

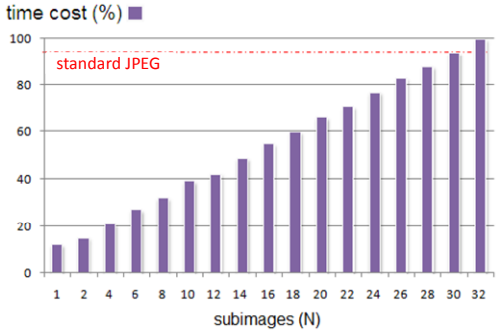


Fig. 11. Time cost with regard to the number of subimages (N)

The results obtained show that this technique introduces flexibility into the delay with the number of subimages that help the time control unit in its decision-making process. The previous figure shows that the cost of the method when it takes all the subimages is worse than the cost of the classic method. This is due to the delay introduced in the space decomposition stage during pre-processing.

3.2.3. Compression rate

Regarding the compression rate when the space subdivision is applied, it can be concluded that the fewer subimages are used, the less storage space is needed, and the greater the rate of compression as a result. The following test shows the relationship between the compression rate and the number of subimages used in the image compression.

Figure 12 shows a drop in the compression rate as the N value increases. The method produces increasingly less compression as more subimages are taken, because more information must be stored each time. Another effect on this characteristic is caused by the D number of subimages. It has been proved that the compression rate and D are inversely related by the same value as N. That is, the larger D is, the smaller the compression rate. This is because the DCT compression model is based on pixel redundancy and code reduction (compaction of zeros through RLE). Consequently, the smaller the image size to be compressed is, the fewer compression margins we found.

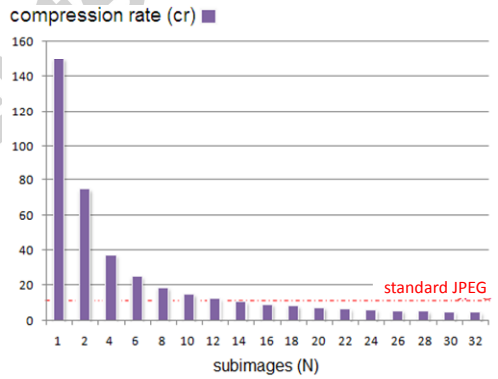


Fig 12. Compression rate with regard to the number of subimages

3.3. Operand precision

The third control parameter to adjust the cost to the requirements of the problem is the accuracy of the operands involved in the image compression/decompression process. For this, specific hardware is needed to run the proposed method that is embedded in the general system. This device can adjust the precision or length of the operands at a low level according to the parameters of the time control unit to run the operations that occur with the required delay.

This section only studies the effect caused by precision on the quality of the compressed image. To estimate the influence of precision on the temporary cost, a hardware implementation needs to be developed for the method. In these experiments, the cutback of digits was simulated by assigning to zero all positions not

considered according to the value of the control parameters. However, these operands always keep their binary size.

The compression rate is not altered by the loss of precision in the operands, since this only affects the fractional part of the numbers and the operations of calculating the DCT and IDCT. The entire division of the standard method then eliminates the fractional part to quantify and code the results.

To perform this experiment, the test images were processed using different precisions for the addition and multiplication operations. Figure 13 shows the results obtained for each operator and for different operand lengths. The other control parameters remain fixed at $d_1 \equiv n=64$ and $d_2 \equiv N=1$, with D=1.

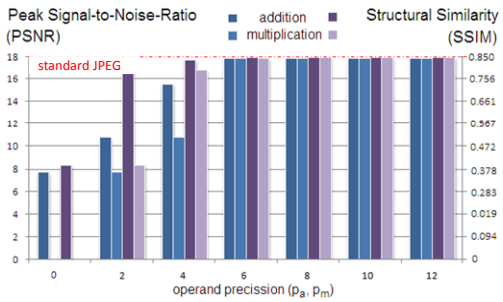


Fig. 13. Objective quality of the compressed image with regard to operand precision

Operands with a maximum precision of 12 fractional digits were used, equivalent to numbers coded in simple precision. The fractional digits were reduced continuously by two until only the whole part of the calculations was obtained.

The previous chart shows that quality is maintained despite that using a low number of decimal digits in the image coding and only when these digits are significantly reduced, the quality drastically drop. In the case of multiplication, the image is completely distorted when a minimum number of decimal digits are not considered. The next table show results for all quality measures.

Table 5
Objective quality of the compressed image with regard to operand precision

	Operand Precission (p_a, p_m)						
	0	2	4	6	8	10	12
PSNR	--	10.79	15.56	17.78	17.78	17.78	17.78
SSIM	--	0.56	0.78	0.85	0.85	0.85	0.85
FSIM	--	0.955	0.958	0.958	0.958	0.958	0.958
Q_{phase}	--	0.925	0.951	0.951	0.951	0.951	0.951
$Q_{mag.}$	--	0.945	0.980	0.980	0.980	0.980	0.980
SVD	--	64.51	30.54	6.92	6.92	6.92	6.92

These data show that no need for high-precision arithmetic when implementing the compression method to obtain acceptable subjective quality. Using this adjustment technique brings a major benefit in terms of flexibility in the processing time when it is run on specific hardware that can handle low-level flexibility. In a previous author's work [8], we describe an arithmetic unit with these novel characteristics.

3.4. Combined tests

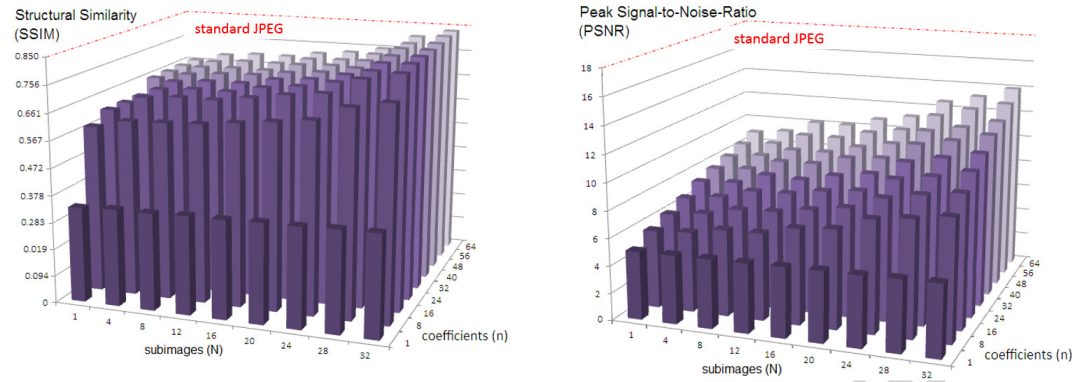
This section studies the joint application of real-time parameters in the image compression/decompression process. The combined effect provides greater capacity for adjusting quality and the cost of obtaining the image. Tests carried out on the delay and the compression rate achieved did not consider precision in the addition and multiplication operations, as a processor would be needed that had been adapted to the operative drop in fractional digits.

3.4.1. Compressed image quality

As could be expected, all parameters influence the quality of the resulting image, distorting it according to their particular kind of deformation: however, it was observed that when some of the parameters reach values close to their limit, the weight of the parameter in the deformation is much greater, and the image is modified according to the particular nature of that parameter.

In the case of the parameter relating to operand precision, it was observed that when precision dips under a certain threshold, the image distorts in the same way as shown in the previous section, but has no effect in other

cases. For this reason, the pictures in figure 14 only show the joint evolution of the number of coefficients and the number of subimages. With the combination of these two parameters, there is a gradual drop in quality the fewer coefficients and subimages are considered. In the case of the PSNR, the slope of the index is roughly constant with N and n, though for measuring the SSIM, the image maintains an acceptable quality except in extreme cases.



a) Structural Similarity b) Peak Signal-to-Noise-Ratio

Fig 14. Objective quality of the compressed image with regard to combined parameters

With respect to other objective measures, the table 6 summarizes the quality results which coincide with the quality relations showed in the previous figure, where it is observed the same scalable properties of the method in [30].

Table 6
Objective quality of the compressed image with regard to combined parameters

D = 32		Coefficients DCT (n)								
		1	8	16	24	32	40	48	56	64
N = 1	PNSR	5.00	5.73	6.21	6.74	7.35	7.75	8.1	8.53	8.96
	SSIM	0.328	0.576	0.609	0.609	0.609	0.65	0.655	0.657	0.657
	FSIM	0.625	0.865	0.901	0.905	0.905	0.906	0.906	0.907	0.907
	Q_{phase}	0.929	0.934	0.939	0.941	0.941	0.941	0.941	0.941	0.941
	$Q_{mag.}$	0.951	0.955	0.959	0.960	0.961	0.962	0.963	0.963	0.963
	SDV	75.82	45.51	39.01	35.48	33.02	30.25	28.05	25.89	24.50
N = 8	PNSR	5.04	6.34	7.07	7.76	8.33	8.75	9.06	9.68	10.16
	SSIM	0.335	0.609	0.670	0.674	0.679	0.684	0.684	0.688	0.695
	FSIM	0.626	0.875	0.911	0.917	0.920	0.919	0.923	0.925	0.926
	Q_{phase}	0.929	0.939	0.943	0.943	0.944	0.945	0.945	0.945	0.945
	$Q_{mag.}$	0.952	0.961	0.965	0.965	0.966	0.968	0.969	0.970	0.971
	SDV	73.25	43.21	36.23	34.12	31.25	28.25	26.02	24.31	22.98
N = 16	PNSR	5.1	7	7.6	8.27	8.76	9.48	9.73	10.31	10.82
	SSIM	0.3396	0.628	0.686	0.697	0.702	0.704	0.712	0.712	0.713
	FSIM	0.627	0.882	0.915	0.925	0.929	0.928	0.931	0.933	0.935
	Q_{phase}	0.929	0.941	0.947	0.947	0.947	0.947	0.947	0.947	0.948
	$Q_{mag.}$	0.952	0.963	0.971	0.973	0.975	0.975	0.976	0.977	0.977
	SDV	70.99	40.91	32.51	31.15	28.64	25.24	23.11	21.85	18.91
N = 24	PNSR	5.15	8.24	8.95	9.63	10.25	10.96	11.4	12.06	12.63
	SSIM	0.344	0.656	0.717	0.731	0.734	0.735	0.74	0.744	0.745
	FSIM	0.628	0.889	0.923	0.938	0.940	0.943	0.945	0.946	0.947
	Q_{phase}	0.930	0.941	0.949	0.949	0.949	0.949	0.949	0.949	0.949
	$Q_{mag.}$	0.953	0.965	0.973	0.975	0.976	0.977	0.977	0.978	0.978
	SDV	69.86	35.15	28.93	27.91	24.83	20.15	18.21	17.44	15.14
N = 32	PNSR	5.23	8.9	9.81	10.64	11.31	12.07	12.52	13.34	13.8
	SSIM	0.350	0.730	0.798	0.808	0.811	0.813	0.817	0.823	0.825
	FSIM	0.629	0.901	0.929	0.945	0.946	0.947	0.948	0.950	0.952
	Q_{phase}	0.934	0.945	0.949	0.949	0.949	0.949	0.949	0.950	0.950
	$Q_{mag.}$	0.953	0.970	0.976	0.978	0.978	0.979	0.979	0.979	0.979
	SDV	68.04	28.67	22.55	21.04	18.47	15.87	13.01	11.51	10.28

3.4.2. Delay estimations

Applying all the parameters together produces a wide range of possibilities for adjusting the process to the needs of each application. Figure 15 shows how the time cost evolves in relation to the number of coefficients and subimages considered.

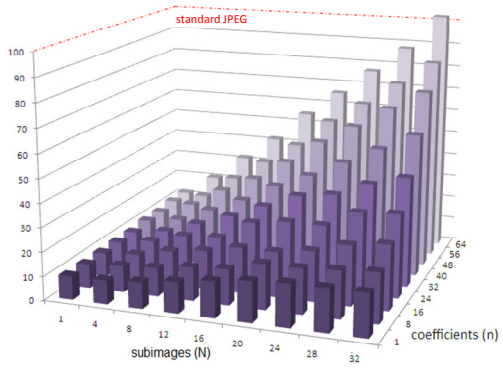


Fig 15. Time cost with regard to the number of subimages (N) and coefficients (n)

As could be expected, a drop in the time cost is brought about by the decrease of both parameters. Furthermore, using fewer subimages in the process has a greater effect on cost when more coefficients are used. For small n values, there is no significant saving in time. This means that it is possible to achieve a greater rate of compression with a lower associated computational cost while keeping the subjective quality of the images at reasonable levels.

3.4.3. Compression rate

By combining the number of coefficients and the number of subimages, considerably high rates of compression can be achieved. The joint graph for this case is not shown here, due to the large values achieved. For low values of both parameters, rates of several hundred were achieved, much higher than the standard JPEG compression rates. These results suggest considerable flexibility for adjusting the transmission of still images to multiple application scenarios.

4. Conclusions

In specific applications for digitally processing images, timing constraints exist in which the data conditioning the validity of the obtained results must be provided. In these systems, the quality of the images obtained is of lesser importance than when the results are obtained.

This paper's main contribution consists of proposing a real-time compression model for still images to adapt the processing time required for image compression and decompression to the time requirements of each application, such as for transmission over the internet. The model proposed is an advance in the field of real-time image processing, as existing proposals generally aim to act on the image quality and are not focused on the processing delay in terms of time constraints.

The model consists of a time control unit and an image compressor/decompressor module, which has adjustable time behaviour according to a set of real-time characteristics, adjusted according to the environmental context, and influencing the time taken to compress and decompress images. These characteristics are: the number of DCT coefficients used, the number of subimages created from the original image to be compressed, and the length of the fractional part of the operands. Each of these characteristics is associated with a time control parameter configured by the control unit.

Our real-time compressor system operates as follows: for a particular application, environmental analysis determines the time restrictions in which the image should be available; these timing constraints are provided by a timing parameter that indicates the permitted delay. This parameter marks the moment that the process ends; the time control unit will configure the control parameter of the compression method from this temporal level; finally, the control parameters establish the value of the real-time characteristics that allow this method to compress or decompress the image in the pre-established time.

The experiments performed show that configuring the real-time characteristics of the method allows for great adjustment of the time cost of the algorithm. It has also been proved that the processing time has consequences on the compression rate and on the quality of the result obtained. However, in applications with a human interface, the loss of objective quality that occurs when reducing the processing time cannot be perceived

subjectively by the human eye. This effect provides a number of configurations so that processing can be adapted to the amount of time available.

References:

[1] R. Fattal, D. Lischinski and M. Werman, *Gradient domain high dynamic range compression*, ACM Transactions on Graphics (2002) 249–256.

[2] M. Okuda and N. Adami, *Two-layer coding algorithm for high dynamic range images based on luminance compensation,* Journal of Visual Communication and Image Representation, Vol. 18, nº 5 (2007) 377-386.

[3] J. Lévy Véhel, F. Mendivil and E. Lutton, *Overcompressing JPEG Images with Evolution Algorithms,* Lecture Notes in Computer Science, vol. 4448 (2007) 383-390.

[4] J. W. Greer, D. M. Rheingold and N. Toutenhoofd, *Mobile Webcasting Method of Multimedia and Geographic Position for Real Time Web Log*. United States Patent 20060282215, (2007).

[5] J. A. Stankovic, R. Rajkumar, *Real-Time Operating Systems*. Real-Time Systems n. 28, (2004) 237-253.

[6] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, Springer, 2004.

[7] J. A. Stankovic, *Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems*. Computer, Vol. 21, nº 10 (1988) 10-19.

[8] H. Mora-Mora, J. Mora-Pascual, J. M. Garcia-Chamizo and A. Jimeno-Morenilla *Real-Time Arithmetic Unit*. Real-Time Systems Vol. 34, nº 1 (2006) 53-79.

[9] D. Chaikalis, N. Sgouros, D. Maroulis and P. Papageorgas, *Hardware Implementation of a Disparity Estimation Scheme for Real-Time Compression in 3D Imaging Applications*. Journal of Visual Communication and Image Representation nº 19: (2008), 1-11.

[10] K. J. Lin, S. Natarajan and J. W. S. Liu, *Imprecise Results: Utilizing Partial Computations in Real-Time Systems*. National Aeronautics and Space Administration, 1987.

[11] J.M. García-Chamizo, J.M. Mora-Pascual, H. Mora-Mora and M.T. Signes-Pont, *Calculation Methodology for Flexible Arithmetic Processing*. VLSI-SOC: (2003), 350–355.

[12] Hernandez-Cabronero, M.; Blanes, I. Serra-Sagrista, J; Marcellin, M.W. *A Review of DNA Microarray Image Compression.* International Conference on Data Compression, Communications and Processing (CCP), (2011).

[13] Kaaniche, M. ; Pesquet, J.-C. ; Benazza-Benyahia, A. ; Pesquet-Popescu, B., *Two-dimensional non separable adaptive lifting scheme for still and stereo image coding. IEEE International Conference on Acoustics Speech and Signal Processing, (2010) 1298 – 1301.*

[14] J. H. In, S. Shirani and F. Kossentini, *On RD Optimized Progressive Image Coding Using JPEG.* IEEE Transactions on Image Processing nº8 (2000) 1630-1638.

[15] M. Minovic, M. Milovanovic, and D. Starcevic, *Delivering Educational Games to Mobile Devices*, International Journal of Knowledge Society Research, vol. 2, no. 2. (2011).

[16] Jampour, M., Yaghoobi, M. and Ashourzadeh M, *Fractal Images Compressing by Estimating the Closest Neighborhood with Using of Schema Theory. Journal of Computer Science Vol. 6, No. 5, (2010).*

[17] Zhang, A, Pei Yang, P., *An improved algorithm for fractal image encoding based on relative error*. International Congress on Image and Signal Processing (CISP), 2012.

[18] K-Y. Min and J-W. Chong, *A Design of Real-Time JPEG Encoder for 1.4 Mega Pixel CMOS Image Sensor SoC*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E88-A, nº 6, (2005) 1443-1447.

[19] S.N. Pattanaik, J.E. Tumblin, H. Yee and D.P. Greenberg, *Time-dependent visual adaptation for realistic real-time image display*. Proceedings of SIGGRAPH, (2000) 47–54.

[20] F. De Simone, F. Dufaux, F. Comparison of DASH adaptation strategies based on bitrate and quality signalling, IEEE 15th International Workshop on Multimedia Signal Processing (2013), 87-92.

[21] C. Liu, I. Bouazizi, M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in Proc. of the second annual ACM conference on multimedia systems, (2011).

[22] ISO/IEC. *Information Technology-Digital Compression and Coding of Continuous-Tone Still Images-Requirements and Guidelines*, International Standard Organization. 81: (1994), 09-92.

[23] S. G. Wolf, R. Ginosar, Y. Y. Zeevi, *Spatio-Chromatic Image Enhancement Based on a Model of Human Visual Information Processing*. Journal of Visual Communication and Image Representation Vol. 9, nº 1 (1998) 25-37.

[24] Y. Kato, *JPEG2000 encoder*, United States Patent 7327892, (2005).

[25] M. Vetterli and A. Ligtenberg, *A Discrete Fourier-Cosine Transform Chip*. IEEE Journal on Selected Areas in Communications Vol. 4, nº 1 (1986) 49-61.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing, vol. 13, no. 4, (2004) 600-612.

[27] Liu, A., Lin, W. and Narwaria, M., *Image Quality Assessment Based on Gradient Similarity*, IEEE Transactions On Image Processing, Vol. 21, nº 4, (2012).

[28] M. Decombas, F. Dufaux, E. Renan. B. Pesquet-Popescu, F. Capman, *A new object based quality metric based on SIFT and SSIM*, IEEE International Conference on Image Processing (ICIP), (2012) 1493-1496.

[29] Z. Wang and Q. Li, *Information Content Weighting for Perceptual Image Quality Assessment*, IEEE Transactions on Image Processing, vol. 20, no. 5, (2011) 1185–1198.

[30] M. Narwaria, W. Lin, I. V. McLoughlin, S. Emmanuel, and L-T. Chia, *Fourier Transform Based Scalable Image Quality Measure*, IEEE Transactions on Image Processing, vol. 21, no. 8, (2012), 3364-3377.
[31] A. Liu, W. Lin, M. Narwaria, *Image quality assessment based on gradient similarity*, IEEE Trans Image Process. vol. 21, no. 4. (2012), 1500-1512.
[32] A. Shnayderman, A. Gusev, A.M. Eskicioglu, *An SVD-Based Grayscale Image Quality Measure for Local and Global Assessment*, IEEE Transactions On Image Processing, vol. 15, no. 2, (2006), 422 - 429.
[33] Lin Zhang, Lei Zhang, X. Mou, D. Zhang, *FSIM: A Feature Similarity Index for Image Quality Assessment*, IEEE Transactions on Image Processing, Vol. 20, no. 8, (2011), 2378 - 2386.

Highlights

- Our method performs real-time encoding/decoding still images based on JPEG standard.
- We introduce parameters that concern to the performance of the image processing.
- Adjusting the parameters affords to time/quality meet the application constraints.
- Experiments and quality test were made to probe the method consistency.