We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Adaptive implementation of nonlinear fuzzy image enhancement algorithms in the compressed JPEG images

Camelia Florea, Aurel Vlaicu, Mihaela Gordan and Bogdan Orza
*Technical University of Cluj-Napoca
Romania*

1. Introduction

With the increasing sizes of high resolution images, their storage and processing directly in the compressed domain has significantly gained importance. Algorithms for compressed domain image processing (Smith, 1993) provide a powerful computational alternative to classical (pixel level) based implementations. While linear algorithms can be applied straightforward to the DCT (Discrete Cosine Transform) encoded JPEG images, this is not the case for non-linear image processing, as for example contrast enhancement algorithms. The implementation of operations involving nonlinear operators is challenging, but not impossible, and when the right combination is found, the processing is much faster, due to the reduced number of coefficients, because the majority of the coefficients in the DCT domain are zero after the quantization.

The field of image processing in the compressed domain is just emerging and the algorithms reported in the literature are mostly based on linear arithmetic operations between pixels. Implementations of image and video sequence processing in the compressed JPEG/MPEG domain are already presented in the literature. Tang (Tang, 2004) defined an algorithm based on the contrast, measured as the ratio of high-frequency content and low-frequency content in the bands of the DCT matrix. In (An et al., 2004) is used the Tang algorithm but each block is enhanced according to its block classification: smooth block or high activity block. A MPEG based image enhancement algorithm for people with low-vision was developed (Kim & Peli, 2003); the contrast enhancement was performed by modifying the quantization matrices for inter and intra frames. Lee (Lee, 2006) uses a basic concept of the Retinex theory for the enhancement of images, directly in the compressed domain.

In recent years, many researchers have applied the fuzzy set theory (Pedrycz & Gomide, 1998; Tizhoosh, 2000; Gonzales, 2008) to develop new techniques for contrast improvement. Fuzzy rule-based contrast enhancement and fuzzy intensification operator, are well-known rather simple approaches with good visual results, but as any fuzzy algorithm, these are by default nonlinear, thus not straightforward applicable on the JPEG bitstream (zig-zag ordered quantized DCT coefficients). Few solutions are reported for this issue, having moderate performances.

This chapter intends to present our approach to a fast and accurate alternative implementation in the compressed domain, for fuzzy image processing algorithms with one or few thresholds as main non-linearity, the other non-linearities being implementable by classical operations in the compressed domain (as the square in the case of the intensification operator) or not needing an explicit non-linear formulation (as is the case of the fuzzy inference in the Takagi-Sugeno fuzzy systems). The solution proposed here can also be applied to other Takagi-Sugeno fuzzy systems for image processing, or for other types of linguistic modifiers (not only the intensification operator) that one can find useful for image processing. Particularly the strategy is applied on two well-known contrast enhancement methods: the one based on the fuzzy intensification operator, and the fuzzy rule-based contrast enhancement method with a Takagi-Sugeno system. The fuzzy sets parameters are adaptively chosen by analyzing the histogram of the DC coefficients of the compressed blocks as an approximation of the grey level statistics of the image, in order to optimally enhance the image contrast.

The core idea behind the presented strategy refers to the implementation of the brightness threshold comparisons, for which an adaptive solution which takes into account the frequency content of each block in the compressed domain JPEG image is suggested. More specifically, this adaptive approach consists in conditionally deciding the need for decompression and pixel level thresholding in a DCT transformed pixel block, only when this is strictly required to preserve accuracy (by a large amount of AC energy of the block). In all the other cases it is enough to compare the DC coefficient (average grey level) value of the block to the equivalent grey level threshold, which is a good estimate of the location of all the pixels brightness versus the threshold. Such an algorithm has been proven (by experiments) to significantly increase the computational efficiency in image processing algorithms applied to a JPEG image (preserving the visual quality of the processing result). The computational efficiency is high since by the strategy, one can avoid the decompression and recompression prior and after the processing. This guarantees the optimal quality at minimum computational cost. The performance of the suggested approach, applied on the 2 fuzzy contrast enhancement methods mentioned above is illustrated in this chapter on several grey scale and color images, and some implementation details are also given. The algorithm is applied only on the luminance component. However, it can be used to enhance color images as well, with no change of the chrominance components (which is a rather common approach in color image enhancement).

2. Fuzzy image enhancement algorithms

Image enhancement involves processing an image in order to make it visually more pleasant to the observers. It is one of the fundamental tasks in image processing. Images have sometimes poor contrast or are blurred. Image enhancement includes a series of different point and spatial operations to improve the contrast, as: piecewise linear grey scale stretching transformation, grey scale clipping, histogram modification/equalization or even highly nonlinear grey scale mappings (Gonzales, 2008).

A particular class of useful techniques in image contrast enhancement is provided by the application of fuzzy sets theory and fuzzy inference systems to this task. The fuzzy sets theory's foundation was set by Prof. Zadeh in 1965 (Zadeh, 1965), followed later on by the fuzzy logic basis, established in 1973; since then, the applications of fuzzy sets theory and

fuzzy logic to scientific computing are extremely vast, and still continue to evolve, along with other modern algorithms in the area of soft computing. In short, to underline the essence of “fuzzy thinking”, which makes it such an appealing tool for the reformulation and implementation of various data processing algorithms, one can see fuzzy variables, fuzzy logic and fuzzy reasoning as the extension of the crisp (binary) reasoning to the infinite valued logic case, which allows for a mathematical representation of the imprecision and uncertainty in the definition of terms – typical to the human-like thinking – and, dually, for a certain “granularity” in defining otherwise precise terms in a more “approximate”, flexible manner. Thus, the fuzzy sets theory and fuzzy logic framework has the main benefit of allowing to still work with a small set of terms, as, e.g.: yes/no answers, true/false evaluations, black/white categorization, but behind each such term, one understands by default (as in the natural language) a certain amount of imprecision, approximation: black is rather “nearly black”, not only the black body’s black; white is “nearly white”, not only pure snow-like white; and, a certain grey can be to some extent, let’s say, dark enough to be similar to black, or white enough to be similar to white, or maybe, almost at all black nor white. Of course, for the natural way in which humans express their reasoning and for their interpretation of the observations, there is nothing special in this kind of processing; however the fuzzy mathematical framework is what provides the required foundation to implement this close to human-like thinking in computer programming.

The core concepts behind fuzzy logic and fuzzy reasoning are: the fuzzy set; the operations on fuzzy sets; the fuzzy relations (which are often expressed in the form of fuzzy rules); the fuzzy approximate reasoning. The fuzzy sets, also called linguistic terms, are generalizations of the crisp sets from the classical sets theory, in the sense that they are categorizations of data with associated soft membership degrees having any values between 0 (no membership at all) and 1 (complete membership), showing in fact how much similar is the data to the prototype of the category. One can also see any fuzzy set as a linguistic value for a human-like expressed concept – e.g., in the example above, black; in the human language, black is more than pure perfect black (the dark body’s black), as we are all aware of. But even so, we can say that a dark grey satisfies the concept “black” rather than the concept “average grey” or “white”, thus, we would assign it some membership degree of, e.g., 0.8 to the fuzzy set “black” and only 0.2 to a fuzzy set representing the concept “average grey”. The specification of the memberships of all data range over which the fuzzy set has been defined (according to the type of data we attempt to describe by our sets) gives the most important attribute of a fuzzy set, by which the fuzzy set can be completely described, i.e., *the membership function of the fuzzy set*. Let us denote the set of all data of some given type that we want to describe/analyze by X ; this set is usually referred as the *universe of discourse* in fuzzy sets theory. Let us consider that a fuzzy set A should group the data satisfying an approximate concept c_A . Then the definition of this fuzzy set is complete if its membership function, expressed as: $A: X \rightarrow [0;1]$, is given.

Various operations can be applied on and between fuzzy sets – most of them being extensions of the crisp sets operations. Some of them, as the complement of a fuzzy set, the intensification, the fuzzification, the power of a fuzzy set, are unary operators (with a single operand). Others, as the union and the intersection of fuzzy sets, are binary or n-ary operators (with two or many operands).

Fuzzy logic, on the other hand, is based on both the concept of fuzzy sets and fuzzy rules. Just as fuzzy sets are generalization of crisp sets, *fuzzy rules* are generalization of the crisp

rules employed in classical expert systems. Fuzzy logic theory is employed in the *fuzzy logic systems*, which are the algorithmic representation of fuzzy reasoning based on rules. A fuzzy logic system, also called fuzzy rule-based system or fuzzy inference system, is just a data processing system having on its input – the crisp value of the input (observed) data, and on its output – the value of some output variable, generated as a result of some fuzzy processing, computed through a *fuzzy inference mechanism* (which implements the *approximate reasoning theory*), based on the representation of the input data classes and output data classes as fuzzy sets and on expressing some associations between the input and the output data classes (input and output fuzzy sets) – associations called *fuzzy rules* and expressed as *fuzzy relations*. Several variants of fuzzy logic systems are currently available; among them, the most widely used are the *Mamdani fuzzy inference systems* (characterized by the presence of fuzzy sets over the input and the output data spaces) and the *Takagi-Sugeno fuzzy inference systems* (in which the input data space is described by fuzzy sets, but the output data space is covered by singleton sets – that is, crisp sets containing each a single data value). The Takagi-Sugeno fuzzy systems are especially appealing for their simple forms and the simplicity of the computations required.

Fuzzy sets and fuzzy rule based systems proved to be powerful tools also for several signal and image processing applications, as often reported in the literature since '80s (Tizhoosh, 2000; Bezdek, 1999). Basically, the theoretical fuzzy sets/fuzzy logic framework found on the basis of grey scale processing is the following: the grey level distribution (grey level range) of a digital image can be modeled by a single fuzzy set or by a fuzzy partition, describing linguistically the concept of “bright” or of “degrees of brightness”. This is compliant to the human perception of the brightness, and therefore allows an easy formulation (in linguistic terms) of several desired grey scale modification algorithms. Then, by applying some operators on the membership functions associated to these fuzzy sets, or by formulating fuzzy logic rules specifying how the output brightness in some spatial location should depend on the input brightness in the original image (rules that are further incorporated in the knowledge base of an image processing fuzzy inference system), one can achieve different processing of the grey levels in the image (of the grey scale dynamics in the image).

A particularly useful class of such fuzzy image processing methods is that of fuzzy contrast enhancement methods – based on both fuzzy sets theory and fuzzy rules based systems (Pal & King, 1981; Pedrycz, 1998; Tizhoosh, 2000). However, few of these frequently used contrast enhancement methods have presented equivalent implementation in the compressed domain, probably because of their nonlinear nature.

2.1. The fuzzy sets based contrast INT (intensification) algorithm as a grey scale transformation

This method is implemented, in principle, by a very simple operation in the fuzzy sets theory: the intensification operation applied on a fuzzy membership function. In principle, one can represent/model the grey scale of a digital image by a single fuzzy set, describing the linguistic concept of “bright levels”: depending on the particular grey level content of each image, more or less of its pixels will have high membership degrees to this fuzzy set. Roughly speaking, if a grey level has a membership less than 0.5 to the “bright levels” set, it is more likely to be dark than bright, whereas in the opposite case it is more likely to be bright than dark. Then, an intensification operator applied on this fuzzy set (Pedrycz &

Gomide, 1998) will increase the membership degrees above 0.5 and decrease the membership degrees below 0.5. If the “intensified” fuzzy set membership function is used in conjunction with the initially defined grey scale to membership function mapping, one can map the newly obtained membership values back to gray levels, and of course for the grey levels that had membership above 0.5 (memberships that are now increased) we will obtain more brighter corresponding grey levels, and the opposite for the memberships below 0.5. This would naturally lead to contrast improvement; recall that this basic point processing operation aims mainly to maximize (increase) the dynamic range of the image. A higher contrast in an image can be achieved by darkening the gray levels in the lower luminance range (typically under 0.5 on a [0; 1] scale) and brightening the ones in the upper luminance range (Gonzales & Woods, 2008).

In the following, the mathematical formulation of the fuzzy set based contrast intensification operation for image contrast enhancement is briefly summarized. Let us consider a very simple (trapezoidal shaped) fuzzy set membership function for representing the concept of “Bright” grey levels, with the linguistic meaning defined above; this is a typical form in fuzzy image processing algorithms – shown below in Fig.1.a. Let us denote the membership function of this fuzzy set by $B: L \rightarrow [0; 1]$. Let $INT(\cdot), INT: [0; 1] \rightarrow [0; 1]$ be the intensification operator applied to a fuzzy membership function.

Thinking in fuzzy terms, according to the meaning of the “Bright” fuzzy set, the lower the dynamic range of the actually observed image will be, the more the membership degrees different (further apart) from 0 and 1 we will have. Thus, the lower the contrast of the image, the larger will be the fuzziness of the fuzzy set. Therefore, in terms of fuzzy membership degrees, enhancing the contrast would roughly mean to reduce the fuzziness of the set (Pal & King, 1981; Pedrycz, 1998). An easy way to achieve this is to apply the fuzzy intensification operator on the membership function of the fuzzy set introduced above. Then the resulting fuzzy set of reduced fuzziness will have the membership function $B': L \rightarrow [0; 1]$, given by:

$$B'(I) = INT(B(I)) = \begin{cases} 2 \cdot B^2(I), & \text{if } 0 \leq B(I) \leq 0.5 \\ 1 - 2 \cdot (1 - B(I))^2, & \text{if } 0.5 \leq B(I) \leq 1 \end{cases} \quad (1)$$

The expression in equation (1) represents a well known operator in the fuzzy sets theory, namely the intensification (INT) operator; when applied on digital images, it has the effect of contrast enhancement.

Fig. 1. a. The linear slope; b. The fuzzy intensification operator.

The choice of the fuzzy set membership function B is not unique, and its definition should be done depending on the particular application. However, for general purpose contrast enhancement tasks, a suitable (due to its simplicity) selection would be a piecewise linear shape, in the form presented in Fig.1.a. This is also the case considered in this paper, in order to allow for an efficient implementation in the compressed domain.

For this shape of B , the result of the fuzzy intensification operator would be the membership function B' shown in Fig.1.b., with less fuzziness than B (or closer to a crisp set than B). The selection of the parameters l_m, l_M, l_{thd} or alternatively, of the slope of B in the range $[l_m; l_M]$ and of the parameter l_{thd} which gives $B(l_{thd}) = 0.5$, should be either set by default to some predefined values (e.g. typically $l_m = 0; l_M = 255; l_{thd} = 128$) or, for better performance, should be selected from the image histogram. In the latter case, the selection is based on the following rules:

- it is reasonable to assume that the value equally plausible to be considered dark and bright (i.e., of 0.5 membership to B) is the median grey level from the histogram;
- the grey level dynamics of the particular image, defined as $[l_{min}; l_{Max}]$, where l_{min} = minimum grey level in the histogram and l_{Max} = the maximum grey level in the histogram, should obey the constraints:

$$\begin{aligned} B(l_{min}) &> 0, \text{ and} \\ B(l_{Max}) &< 1 \end{aligned} \quad (2)$$

imposed by the fact that the maximum range of fuzzy membership degrees is $[0; 1]$ and that, to guarantee some contrast enhancement especially for narrow histograms, l_{min} and l_{Max} should not take the extreme membership values 0 and 1, which cannot be modified by the algorithm. In grey level notation terms, the constraint (2) can be rewritten as:

$$\begin{aligned} l_m &\leq l_{min}, \text{ and} \\ l_M &\geq l_{Max} \end{aligned} \quad (3)$$

We choose in this implementation to keep one constraint to the limit, i.e. either $(l_m = l_{min} \text{ and } l_M \geq l_{Max})$ or $(l_m \leq l_{min} \text{ and } l_M = l_{Max})$, whatever of the two can be satisfied for the currently processed image.

Expressing analytically the membership function B from Fig.1.a. as:

$$B(l) = \begin{cases} 0, & \text{if } l \in [0; l_m] \\ a \cdot l + b, & \text{if } l \in [l_m; l_M], \\ 1, & \text{if } l \in [l_M; 255] \end{cases} \quad (4)$$

and taking into account the above rules and constraints, we can express the slope a as:

$$a = \begin{cases} \frac{B(l_{thd}) - B(l_m)}{l_{thd} - l_{\min}}, & \text{if } (l_m = l_{\min} \text{ and } l_M \geq l_{Max}) \\ \frac{B(l_M) - B(l_{thd})}{l_{Max} - l_{thd}}, & \text{if } (l_m \leq l_{\min} \text{ and } l_M = l_{Max}) \end{cases}, \quad (5)$$

which is equivalent to:

$$a = \begin{cases} \frac{0.5}{l_{thd} - l_{\min}}, & \text{if } (l_m = l_{\min} \text{ and } l_M \geq l_{Max}) \\ \frac{0.5}{l_{Max} - l_{thd}}, & \text{if } (l_m \leq l_{\min} \text{ and } l_M = l_{Max}) \end{cases} \Rightarrow a = \frac{0.5}{\max\{l_{thd} - l_{\min}, (l_{Max} - l_{thd})\}}. \quad (6)$$

The value of the second parameter, b , is obtain from $B(l_{thd})$:

$$B(l_{thd}) = 0.5 \Rightarrow a \cdot l_{thd} + b = 0.5 \Rightarrow b = 0.5 - a \cdot l_{thd}. \quad (7)$$

Notice that for the default case, $l_m = 0, l_M = 255, l_{thd} = 128$, the values of a, b will be:

$$a = \frac{1}{255}, b = 0.$$

2.2. The fuzzy rule-based contrast enhancement algorithm

Another possible way to express the image contrast enhancement in terms of fuzzy logic is by the means of a Takagi-Sugeno fuzzy rule based system (Gonzalez & Woods, 2008; Tizhoosh, 2000). A common formulation assumes the description of the grey scale of the input image by 3 linguistic terms, denoted by *Dark*, *Gray* and *Bright*. Typically, the terms *Dark* and *Bright* are represented by trapezoidal-shaped fuzzy membership functions, whereas the term *Gray* is described by a triangular-shaped fuzzy membership function, as illustrated in Fig.2.a. Accordingly, on the universe of discourse of the output variable (i.e., the grey scale of the enhanced image), the other 3 linguistic terms are defined, referred here as: *Darker*, *Midgray* and *Brighter*. Since we consider a Takagi-Sugeno fuzzy system, the output fuzzy sets will be fuzzy singleton (or, numerical constants denoted by l_v^d for *Darker*, l_v^g for *Midgray* and l_v^b for *Brighter*), as shown in Fig.2.b. If one denotes the input variable (describing grey levels in the input range) by l , $l \in \{0, 1, \dots, 255\}$, and the output variable (describing the gray level in the output image) by l_o , $l_o \in \{0, 1, \dots, 255\}$, the fuzzy rule base of the Takagi-Sugeno contrast enhancement fuzzy system comprises the following 3 rules:

- R1: IF l is *Dark* THEN l_o is *Darker*
- R2: IF l is *Gray* THEN l_o is *Midgray*
- R3: IF l is *Bright* THEN l_o is *Brighter*,

or, equivalently,

R1: IF l is *Dark* THEN $l_o = l_o^d$ is *Darker*

R2: IF l is *Gray* THEN $l_o = l_o^g$ is *Midgray*

R3: IF l is *Bright* THEN $l_o = l_o^b$ is *Brighter*

Fig. 2. a. Input, and b. Output membership functions for fuzzy, rule-based contrast enhancement.

Then for any value l^* at the input of our Takagi-Sagano contrast enhancement fuzzy system, in the output image, the corresponding brightness l_o^* is obtained by applying the Takagi-Sugeno fuzzy inference, as:

$$l_o^* = \frac{\mu_{Dark}(l^*) \cdot l_o^d + \mu_{Gray}(l^*) \cdot l_o^g + \mu_{Bright}(l^*) \cdot l_o^b}{\mu_{Dark}(l^*) + \mu_{Gray}(l^*) + \mu_{Bright}(l^*)}, \quad (8)$$

where: $\mu_{Dark}(l^*)$, $\mu_{Gray}(l^*)$, $\mu_{Bright}(l^*)$ denote the membership degrees of the currently processed brightness l^* to the input fuzzy sets *Dark*, *Gray* and *Bright*.

In the implementation presented here, we assume that the input fuzzy sets form a fuzzy partition of the universe of discourse of l :

$$\mu_{Dark}(l) + \mu_{Gray}(l) + \mu_{Bright}(l) = 1 \quad (9)$$

The numerical constants defining output singletons were selected here at: $l_o^d = 1$ (black), $l_o^g = 127$ (gray), $l_o^b = 255$ (white).

For a general computational framework of any of the 3 membership degrees required by equation (8), we propose to represent each of the input fuzzy sets membership functions by a trapezoidal function $f_r : L \rightarrow [0; 1]$, where L is the dynamic grey level range of the image, $L = \{0, 1, \dots, 255\}$, represented generically in Fig.3, in analytical form:

$$f_{tr}(a,b,c,d,l) = \begin{cases} 0, & \text{if } l \in [0, a) \\ \frac{l-a}{b-a}, & \text{if } l \in [a, b) \\ 1, & \text{if } l \in [b, c] \\ \frac{-l+d}{d-c}, & \text{if } l \in (c, d] \\ 0, & \text{if } l \in (d, 255] \end{cases} \quad (10)$$

With this generic function f_{tr} , any of the 3 membership functions, $\mu_s : L \rightarrow [0; 1]$, where $s \in \{Dark, Gray, Bright\}$, in Fig.2.a can be expressed, for particular choices of the parameters a, b, c, d :

$$\begin{aligned} \mu_{Dark}(l) &= f_{tr}(0, 0, T_1, T_2, l) \\ \mu_{Gray}(l) &= f_{tr}(T_1, T_2, T_2, T_3, l) \\ \mu_{Bright}(l) &= f_{tr}(T_2, T_3, 255, 255, l). \end{aligned} \quad (11)$$

The thresholds: T_1 , T_2 and T_3 are chosen from the image histogram like the minimum the mean and the maximum gray level values (as suggested in (Gonzales & Woods, 2008) and exemplified in Fig.13.a).

Fig. 3. Function for computing the membership degree on dark, gray and bright.

3. Basis of image processing in the compressed domain

The bitmap format for image storage is seldom used in day-to-day applications; even medical images are stored in JPEG format. This is because of the advantages offered by the JPEG format: little storage capacity needed and better performances in information transmission (via internet). For images stored in JPEG format, it is recommended to process them directly in the compressed domain, in order to reduce the time needed to process data. First, this time economy is related to the fact that it is no longer necessary to decompress the image, to process it at pixel level and to recompress it back; and, second, the image processing in the compressed domain means that there are fewer data to process.

Linear processing can be easily implemented in the compressed domain (Smith & Rowe, 1993; Smith, 1995) because there are no problems to process an image involving linear operations, such as adding a constant, multiplying with a constant, adding of two images, progressive cross fade of two images. The implementation of operations involving nonlinear

operators is challenging, but not impossible, and when the right combination is found, the processing is much faster, due to the reduced number of DCT coefficients. Further, the basic steps used to compress/decompress the JPEG images will be briefly presented (Smith & Rowe, 1993). The image is first divided into 8x8 blocks, and each 8x8 block is individually processed. A DCT is applied on each block providing the DCT coefficients, which are quantized. Many small coefficients, usually high frequency ones, are quantized to zero. The next step is zig-zag scanning of the DCT matrix, followed by RLE (Run Length Encoding) and entropy coding (Huffman coding). In the decoder, the inverse steps must be computed.

There are two ways to enhance the images which are compressed using JPEG (Fig.4):

- (1) the compressed domain processing - no decompression/ recompression, but the enhancement algorithm must be reformulated in the DCT image representation space;
- (2) the pixel level processing - enhancement of the image after full decompression, direct manipulation of the pixels is adopted, than recompress the enhanced image;

Fig. 4. The two ways to process images compressed JPEG.

For each 8x8 pixels blocks from image, instead of 64 data implied in the pixel level processing, in the compressed domain only a smaller amount of data is processed, because the majority of the coefficients in the DCT domain are zero after the quantization. In our algorithm an adaptive minimal decompression is used: full decompression is no longer needed, but decompression is used for the block having many details, for an improved accuracy of processing.

The following notations will be introduced: considering the original $H \times W$ grey level image, divided into 8×8 pixels blocks (as in the typical process of JPEG encoding), we denote any such block of pixels by the matrix $U[8 \times 8]$, containing the grey level values of the pixels in the range $[0; L]$, where L is the dynamic grey level range of the image, $L = \{0, 1, \dots, 255\}$: $U = \{u(i, j) \in [0; L], \text{ where } i = 0, 1, \dots, 7; j = 0, 1, \dots, 7\}$. Consider $U_{dct}[8 \times 8]$ to be the matrix of the zig-zag ordered quantized DCT coefficients of any 8×8 pixels block from the original image U (directly available in the JPEG image coding, as shown in Fig.4), where:

$$U_{dct} = \begin{cases} u_{dct}(0,0) - DC \text{ coefficient,} \\ u_{dct}(i, j) - AC \text{ coefficients, (where } i, j = 0, 1, \dots, 7; \text{ with } i \neq j \neq 0) \end{cases} \quad (12)$$

In the JPEG compression steps, prior to applying the DCT on each block, all the luminance values are scaled symmetrically towards 0, from the $[0; 255]$ range to the $[-128; 127]$ range. Since we have in the compressed domain all the gray values scaled symmetrically towards 0, we should express all the terms from classical equations (pixel level algorithms) in terms of these translated grey levels. Note that the DC coefficient will also be scaled towards 0. To

differentiate between a grey level l in the original range $[0; 255]$ and its translated version, we will denote the latter by adding the subscript “ t ”, by l_t : $l_t = l - 128 \Rightarrow l = l_t + 128$.

Unlike the basic grey scale transformation formulation of the contrast enhancement, if one aims to obtain the same effect with the general approach presented in Fig.4, the algorithms given in the previous section must be reformulated as a block level processing. Whereas this is not a problem for the linear operations, the nonlinear operations must be carefully addressed.

To compute the membership degrees, we need to perform the operation of adding a constant and multiplying with a constant each pixel brightness value in a DCT block. These imply linear operations. The multiplication with a constant implies multiplying the constant with each coefficient from the DCT matrix, whereas the addition of a constant can be seen as a translation of the average brightness of the block, therefore it will affect only the DC coefficient of the 8×8 pixels block. These two scalar operations needed for the JPEG compressed image processing (Smith 1995), can be performed directly on the U_{dct} matrix by simply changing the values - there is no need to reconstruct the quantized array or even the zig-zag vector. The other operations (linear or nonlinear), needed for the reformulations of the fuzzy algorithms are explained in algorithms description sections.

As mentioned in section above, a reasonable choice for the thresholds values would be from the image histogram. But, in the JPEG compressed domain image representation, the pixel grey levels are not directly available (without decompression). Therefore, one cannot directly compute the grey level histogram for the image and, as such, neither the median grey level. However, what we do have available are the average values of the grey levels in each 8×8 pixels “patch” in the image, given by the DC coefficients of the blocks composing the image. Roughly speaking, if they would be the only ones used to reconstruct the pixel level representation (without any AC information), they would give an approximation of the image, with some block boundary effects/ distortions and some loss of details, but however still preserving the significant visual information. Therefore, the histogram built only from the DC coefficients will have also approximately the same shape as the grey level histogram, built from pixel-level data. This is illustrated in Fig.5.

Fig. 5. Histogram: a. of DC coefficients in the compressed domain, b. at pixel level.

4. An adaptive approach to the brightness thresholding in the compressed domain

The core idea is to develop a fast and accurate alternative implementation in the compressed domain, to the classical fuzzy image enhancement algorithms, based on global brightness thresholding, which is usually performed at pixel level. One of the most important difficulties in deriving a compressed domain implementation for these algorithms is the fact

that it involves one or two threshold comparison of each grey level to a mean value, which cannot be directly transposed to a processing on the DCT coefficients domain. To this, we developed an adaptive approach, which conditionally decides the need for decompression and pixel level threshold in a DCT transformed pixel block, only when this is strictly required to preserve accuracy (by a large amount of AC energy of the block). In all the other cases it is enough to compare the DC coefficient (average grey level) value of the block to the equivalent grey level threshold, which is a good estimate of the location of all the pixels brightness versus the threshold. Such an algorithm can significantly increase the computational efficiency in image processing algorithms applied to a JPEG image, avoiding compression and decompression prior and after the processing.

The lowest the number of the zero AC coefficients, the more textured is the 8x8 block considered (being proportional to the spatial frequencies in the area). The largest the energy of the AC coefficients, the more significant the luminance changes in the 8x8 pixels area. Large amount of energy usually corresponds to sharp edge and important surface/area discontinuities. Thus in this case, taking into account the average luminance of the area is very likely to produce incorrect thresholding results; such a case should be therefore treated differently than the uniform or almost uniform areas, requiring the block decompression, pixel level processing and compression.

The adaptive algorithm is based on formulating some logical rules to differentiate between the cases presented above and to define the best processing steps according to each situation to maximize the thresholding accuracy while minimizing the computational complexity by processing as few image elements as possible. Based on the DCT coefficients, one can see that the frequency content of each 8x8 block from the image can be estimated, classifying the blocks into:

- 1) Approximately uniform blocks (of almost uniform luminance) - the energy contained in the AC coefficients is very small.
- 2) Non-uniform blocks (of significant variable luminance) which could contain:
 - a) Only a few details, but significant for the object (for example, horizontal, vertical and oblique edges, like in Fig.6.a)
 - b) A large number of significant details for the object (like in the case of a "chess table" image of 8x8 presented in Fig.6.b).

Fig. 6. Non-uniform blocks: a. Horizontal, vertical and oblique edges; b. "chess table"

Fuzzy image processing in the compressed domain is not a trivial task, since as it can be seen, if one aims to strictly apply classical algorithms in that form, the comparison would necessarily need the decompression; otherwise one cannot have the grey level value in each of the 64 possible spatial locations, but then we would return to what we aimed to avoid, i.e. full decompression before processing. Fortunately, there are typically many 8x8 pixels blocks in general purpose digital images where the local variation of the brightness is small around the average (i.e., around the DC coefficient of the block). Thus for every such block, it is reasonable to assume that if its average grey level falls on one side of the threshold DC_{thd} , on the same side will be all the individual brightnesses of the pixels in the block.

In the case of 8×8 blocks of pixels exhibiting a large variation of the grey levels around the average brightness value (this is equivalent in the DCT coefficients space to large values of the AC coefficients – which will mean that enough of the 64 pixels of the block have significantly different brightness values than the value of the DC coefficient of the block), performing the comparison with the threshold only on the DC coefficient and using the resulting selected expression of the contrast intensification function for all the 64 pixels in the block can lead to an erroneous processing of the grey levels of the pixels whose grey levels are significantly different from the average brightness value. An example illustrating the processing errors resulting in such a case is presented in Fig.7: notice that for the selected region in Fig.7.b), some of the blocks close to the boundaries of the apple and the tail of the banana, which have a large variance of the grey level, exhibit a much too significant increase of the brightness as compared to the surrounding area of the background. This is due to the incorrect selection of the processing function, determined by the comparison of only the DC coefficient of the block with the threshold DC_{thd} . To avoid the processing errors introduced by the incorrect selection of the grey level enhancement function in the case of the blocks with a highly non-uniform luminance distribution, (large AC energy content), one must perform the decompression of all the blocks containing higher energy factor, and process them at the pixel level (see Fig.7.c.). Even if this reduces to some extent the computational efficiency, as compared to the case of a complete processing in the compressed domain, the loss is not significant, since statistically the number of blocks for which the decompression must be applied is small for most images (i.e., only the blocks exhibiting sharp edges must be decompressed for processing, all the others can be processed completely in the compressed domain), as it can be seen in the experimental part (table 1 and 2). For the 8×8 pixels blocks that are fully decompressed before processing, no rewriting of the contrast intensification algorithm is needed; it is processed at pixel level, using the classical algorithm.

Fig. 7. a. Original image “Banana.jpg”; b. Image processed without consider AC energy content; c. Image processed with adaptive algorithm in the compressed domain.

In the adaptive implementation, the decision whether an 8×8 pixels block needs to be decompressed for applying the enhancement algorithm or not, is done based on estimating its amount of grey level variation versus the average brightness as being proportional to the average AC energy content within the block, amount denoted by E_{AC} and described by the following formula:

$$E_{AC} = \frac{\sum_{i=0}^7 \sum_{j=0}^7 |U_{dct}(i, j)|^2 - U_{dct}^2(0,0)}{63} \quad (13)$$

Then, if the computed estimate of the brightness variation E_{AC} in the currently processed block of 8×8 pixels exceeds a certain threshold, denoted here as e_{thd} (is too large to allow the assumption that the selection of the enhancement function can be done based on the average luminance value $u_{dct}(0,0)$ only), the decision is made to decompress the pixels block and process the grey levels individually, using pixel level equations. Otherwise, the brightness variation within the block is small enough to allow the selection of the processing function based only on the position of $u_{dct}(0,0)$ towards the threshold l_{thd} and do the processing in the compressed domain directly on the U_{dct} matrix, as is describe below (in the next sections).

The e_{thd} value is chosen from the experiments, taking into account the image statistics with respect to the amount and magnitude of image edges (the amount and sharpness of the details within the image). Thus, for images with similar statistics with respect to the frequency content, the same AC energy threshold e_{thd} may be reliably used in the selection of the processing type (with/without decompression). For a given image class, the value e_{thd} providing the best compromise between computational complexity (minimal number of blocks decompressed) and quality of the enhanced image (which, without compromises, should appear visually identical to the image enhanced at pixel level) can be found by examining:

- the plot of the mean square error (MSE) between the enhanced image using our algorithm and the enhanced image using the original algorithm (applied directly at pixel level) for the entire image versus various values of e_{thd} , and
- the plot of the number of blocks decompressed in applying our algorithm, denoted EffBlocks (the estimated efficiency of the adaptive processing method for the compressed domain, evaluated by examining the numbers of blocks processed at pixel level as reported to the total number of 8×8 pixels blocks in the image), versus various values of e_{thd} .

Fig. 8. e_{thd} influence for an image (e.g. Eye.jpg)

Such a pair of plots for the image in Fig.9 is given in Fig.8. Their examination allows us to state that the compromise value of e_{thd} appears at the intersection of the MSE vs. e_{thd} curve and EffBlocks vs. e_{thd} curve. Thus, the optimal threshold should be chosen in the neighborhood of this intersection point, for images with similar grey level distribution statistics.

A very small e_{thd} value will always lead to images of a very good quality, but the number of processed blocks in the compressed domain will be quite small, so the complexity of the computing algorithm is not significantly smaller, compared to the direct processing on the pixel. An appropriate e_{thd} value will lead to the increase of the number of blocks processed in the compressed domain and, in this way, a fast fuzzy algorithm for image enhancement could be obtained.

The adaptive approach to the brightness thresholding in the compressed domain is defined for each 8×8 pixels block of the JPEG image, as follows:

- 1) Compute the average of AC coefficients energy from DCT block, denoted E_{AC} , using equation (13).
- 2) If $E_{AC} < e_{thd}$ (where e_{thd} represents the optimal selection threshold between the uniform blocks and the blocks with a significant number of details) $\Rightarrow$ process the block in the compressed domain.
- 3) Otherwise, if $E_{AC} \geq e_{thd}$ (the block has a significant content of details) $\Rightarrow$ decompress the block and process every pixel from the block separately, using classical equation.

5. A reformulation of the fuzzy contrast enhancement algorithms in the compressed domain

5.1. Contrast enhancement algorithm based on fuzzy INT operator in the compressed domain

Before discussing the implementation of each operation involved by the contrast enhancement with fuzzy INT operator directly in the compressed domain (Florea et al., 2009), we would first like to rewrite the grey level transformation implemented by the algorithm in analytical form as: $f_{int} : L \rightarrow L$, embedding in its form the conversion of the grey levels to fuzzy membership degrees (fuzzification given by B) the INT operator and the conversion of the resulting membership degrees back to gray levels (defuzzification, given by B^{-1}).

Taking into account that, for a given image with the minimum grey level in the histogram l_{min} and the maximum grey level in the histogram l_{Max} , the previously imposed constraints guarantee that $[l_{min}, l_{Max}] \subseteq [l_m, l_M]$, the fuzzification is completely described by:

$B(l) = a \cdot l + b$, for any l in the image.

Therefore, $B'(l)$ becomes:

$$B'(l) = INT(B(l)) = \begin{cases} 2 \cdot (a \cdot l + b)^2, & \text{if } 0 \leq (a \cdot l + b) \leq 0.5 \\ 1 - 2 \cdot (-a \cdot l + 1 - b)^2, & \text{if } 0.5 < (a \cdot l + b) \leq 1 \end{cases} \quad (14)$$

This can be written in a form more suitable for the identification of the needed operations, to be performed in the compressed domain as:

$$B'(l) = \begin{cases} 2 \cdot a^2 \cdot l^2 + 4 \cdot a \cdot b \cdot l + 2 \cdot b^2, & \text{if } 0 \leq (a \cdot l + b) \leq 0.5 \\ -2 \cdot a^2 \cdot l^2 + 4 \cdot a \cdot (1-b) \cdot l + 1 - 2 \cdot (1-b)^2, & \text{if } 0.5 < (a \cdot l + b) \leq 1 \end{cases} \quad (15)$$

The resulting grey level, $l' = f_{\text{int}}(l) = B^{-1}(B'(l))$, will be:

$$f_{\text{int}}(l) = \frac{B'(l) - b}{a} \Rightarrow$$

$$f_{\text{int}}(l) = \begin{cases} 2 \cdot a \cdot l^2 + 4 \cdot b \cdot l + \frac{2 \cdot b^2 - b}{a}, & \text{if } 0 \leq (a \cdot l + b) \leq 0.5 \\ -2 \cdot a \cdot l^2 + 4 \cdot (1-b) \cdot l + \frac{1 - 2 \cdot (1-b)^2 - b}{a}, & \text{if } 0.5 < (a \cdot l + b) \leq 1 \end{cases} \quad (16)$$

In the following, we describe the implementation of each term from equation (16) in the compressed domain.

Taking into account that, in the compressed domain, prior to applying the DCT based compression of the JPEG algorithm, all the luminance values are scaled to the $[-128, 127]$ range, including the threshold grey level l_{thd} . Consequently, the threshold for the fuzzy algorithm in the compressed domain should change to its equivalent in the range $[-128, 127]$, denoted here by DC_{thd} : $DC_{thd} = l_{thd} - 128$. Note that the resulting DC coefficient (which gives the average brightness in the block and used in our implementation as an estimate for selecting the processing rule) is also scaled towards 0.

Since the value of l_{thd} can be computed from the membership function B as:

$$a \cdot l_{thd} + b = 0.5 \Rightarrow l_{thd} = \frac{0.5 - b}{a} \Rightarrow l_{thd} - 128 = \frac{0.5 - b}{a} - 128, \text{ the expression of } DC_{thd} \text{ becomes:}$$

$$DC_{thd} = \frac{0.5 - b}{a} - 128. \quad (17)$$

In the compressed domain all the grey values are scaled symmetrically towards 0, which means, that we should express all the terms in equation (16) in terms of these translated grey levels:

$$l_t = (l - 128) \Rightarrow l = l_t + 128,$$

$$l_t^2 = (l - 128)^2 \Rightarrow l^2 = l_t^2 + 2 \cdot 128 \cdot l_t - 128^2. \quad (18)$$

Replacing the expressions of l and l^2 from equation (18) in equation (16), we will have the following resulting form of the intensification function:

$$f_{\text{int}}(l_t) = \begin{cases} 2 \cdot a \cdot l_t^2 + (4 \cdot a \cdot 128 + 4 \cdot b) \cdot l_t + (2 \cdot a \cdot 128^2 + 4 \cdot b \cdot 128 + \frac{2 \cdot b^2 - b}{a} - 128), \\ \text{if } 0 \leq (a \cdot l + b) \leq 0.5 \\ -2 \cdot a \cdot l_t^2 + (-4 \cdot a \cdot 128 + 4 \cdot (1 - b)) \cdot l_t + (-2 \cdot a \cdot 128^2 + 4 \cdot (1 - b) \cdot 128 + \\ + \frac{1 - 2 \cdot (1 - b)^2 - b}{a} - 128), \text{ if } 0.5 < (a \cdot l + b) \leq 1 \end{cases} \quad (19)$$

In order to implement the fuzzy algorithm in the compressed domain, linear operations and nonlinear operations are necessary. The linear operation needed are adding and multiplying a constant to each pixel brightness in an image block. But also, the square of an image is needed (equivalent to l_t^2), which is a nonlinear operation. It can be obtained using the multiplication algorithm of two images, in the compressed domain, developed by B. Smith (Smith & Rowe, 1993). In order to apply the contrast enhancement algorithm described above, we need to compute the square of each pixel luminance in the 8×8 block, in the compressed domain. That is, we should obtain the DCT of the 8×8 block of squared luminance. We denote this DCT matrix by $U_{dct,sq}[8 \times 8]$. According to (Smith & Rowe, 1993), $U_{dct,sq}$ can be obtaining from U_{dct} as follows:

$$\begin{aligned} u_{dct,sq}(x_1, x_2) &= \frac{1}{4 \cdot Q(x_1, x_2)} \sum_i \sum_j C(i, x_1) \cdot C(j, x_2) \cdot [u(i, j)]^2 = \\ &= \sum_{y_1, y_2, w_1, w_2} u_{dct}(y_1, y_2) \cdot u_{dct}(w_1, w_2) \cdot W_Q(y_1, y_2, w_1, w_2, x_1, x_2) \end{aligned} \quad (20)$$

$$\text{where: } W_Q(y_1, y_2, w_1, w_2, x_1, x_2) = \frac{Q(y_1, y_2) \cdot Q(w_1, w_2)}{256 \cdot 64 \cdot Q(x_1, x_2)} \cdot W(x_1, y_1, w_1) \cdot W(x_2, y_2, w_2)$$

$$\text{with: } W(x, y, w) = \sum_i C(i, x) \cdot C(i, y) \cdot C(i, w);$$

$$C(i, x) = A(x) \cdot \cos \frac{(2 \cdot i + 1) \cdot x \cdot \pi}{16}; \quad A(x) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } x = 0 \\ 1, & \text{for } x \neq 0 \end{cases}.$$

We can efficiently compute this sum by noticing several facts: firstly, that for compressed images, most of the coefficients in the matrix U_{dct} are zero and, secondly, that in the function $W_Q(y_1, y_2, w_1, w_2, x_1, x_2)$, only about 4 percents of the terms are non-zero.

The reformulation in the compressed domain for the classical fuzzy image enhancement algorithm using INT operator, expressed in equation (1), can be described by the following equations:

$$\begin{aligned} U_{dct,INT} &= k_1 \cdot U_{dct,sq} + k_2 \cdot U_{dct} \\ u_{dct,INT}(0,0) &= u_{dct,INT}(0,0) + k_3 \end{aligned} \quad (21)$$

where k_1 , k_2 and k_3 are real-valued constants defined by:

$$\begin{aligned} \text{if } u_{det}(0,0) < DC_{thd} \Rightarrow & \begin{cases} k_1 = 2 \cdot a; \\ k_2 = 4 \cdot a \cdot 128 + 4 \cdot b; \\ k_3 = 2 \cdot a \cdot 128^2 + 4 \cdot b \cdot 128 + \frac{2 \cdot b^2 - b}{a} - 128 \end{cases} \\ \text{if } u_{det}(0,0) \geq DC_{thd} \Rightarrow & \begin{cases} k_1 = -2 \cdot a; \\ k_2 = -4 \cdot a \cdot 128 + 4 \cdot (1 - b); \\ k_3 = -2 \cdot a \cdot 128^2 + 4 \cdot (1 - b) \cdot 128 + \frac{1 - 2 \cdot (1 - b)^2 - b}{a} - 128 \end{cases} \end{aligned} \quad (22)$$

Notice that: $u_{det}(0,0)$ is the corresponding DC coefficient of the 8x8 pixels block of the matrix U, and $u_{det,INT}(0,0)$ is the DC coefficient of the enhanced matrix, obtained using our algorithm.

There are typically many 8x8 pixels blocks in general purpose digital images where the local variation of the brightness is small around the average (i.e., around the DC coefficient of the block). Thus for every such block, it is reasonable to assume that if its average grey level falls on one side of the threshold DC_{thd} , on the same side will be all the individual brightnesses of the pixels in the block. This assumption will not hold in only one case, but the namely, when although the individual pixels brightnesses are close to the average (DC) value, then DC value is itself very near to the threshold DC_{thd} . However, in this case, as presented in Fig.1.b, the change of the membership degree will only very slightly affect the brightness, and as such, the error that can appear by selecting, for some of the pixels, the wrong equation for processing will be visually not noticeable. The adaptive implementation was applied: for any such blocks (described above) the processing is performed directly in the compressed domain using the equation (21), but, the blocks with high energy contents will be decompressed and pixel level processed using the equation (1).

The software implementation was done using the C++ programming language. A set of different images, with different dimensions, content, details content, contrast and medium luminance has been chosen for implementation. The performance of the algorithm has been examined, with respect to: MSE, and EffBlocks.

When we have a dark/bright image and we want to enhance it, if we define the fuzzy set membership function B with the default value of the parameter l_{thd} (i.e., 128), the image will be more dark/bright, but if we define the membership function using for l_{thd} the median value selected from the histogram of the DC coefficients, the image will be correctly enhanced.

Results of enhanced images using the fuzzy contrast enhancement algorithm, using INT operator directly in the compressed domain, are presented in the next figures:

Fig. 9. Original image Eye.jpg and enhanced image using the proposed algorithm.

Fig. 10. a. Original image Cars.jpg; b. Enhanced image using the default value for the parameter l_{thd} of the fuzzy set B, $l_{thd}=128$; c. Enhanced image using the proposed algorithm, with the value l_{thd} adaptively selected from the histogram of the DC coefficients;

Fig. 11. Original image oldPicture.jpg and enhanced image using the proposed algorithm;

Fig. 12. Original image Keyboard.jpg and enhanced image using the proposed algorithm;

<i>Image name</i>	<i>e_{thd}</i>	<i>l_{thd}</i>	<i>Number of blocks processed at pixel level</i>	<i>Total number of blocks 8x8</i>	<i>EffBlocks [%]</i>	<i>MSE [%]</i>
banana.jpg	10	157	147	1344	10.938	0.223
eye.jpg	10	117	104	912	11.404	1.714
medical1.jpg	20	97	49	1768	2.771	0.641
medical2.jpg	20	90	67	1976	3.391	0.7259
medical3.jpg	20	52	235	1440	16.319	8.272
biopsi1	10	98	0	1200	0.000	0.0082
biopsi2	20	85	34	1200	2.833	0.074
lisa.jpg	10	88	70	1000	7.000	0.262
cars.jpg	15	161	25	1200	2.083	1.059
keyboard.jpg	15	34	2	4800	0.042	0.172
oldPicture1.jpg	10	207	174	4800	3.625	0.11
oldPictGirl.jpg	15	117	63	3136	2.009	1.125
photoInside.jpg	15	99	39	4800	0.813	0.831
seaLion.jpg	15	55	46	4800	0.958	0.177
barber.jpg	15	114	147	1520	9.671	2.918
nature.jpg	10	117	31	1728	1.794	2.89
lena.jpg	10	118	407	4096	9.937	0.6
mouse.jpg	20	51	47	1000	4.700	0.06
wom1.jpg	10	107	4	1024	0.391	0.66
sap1.jpg	20	83	31	784	3.954	2.334
fingerPrint.jpg	10	117	118	2016	5.853	0.824
jacob_strand.jpg	15	156	16	1200	1.333	0.541

Table 1. Results for different values *e_{thd}*

5.2. The reformulation of the fuzzy rule-based algorithm in the compressed domain

To compute the membership degrees for the fuzzy rule-based contrast enhancement algorithm in the compressed domain (Popa et al., 2008), we need to perform the operation of adding a constant and multiplying with a constant each pixel brightness value in a DCT block. These are linear operation.

Let denote with $K_s[1 \times 2]$ ($s \in \{Dark, Gray, Bright\}$) the line vector of two constants (necessary for adding and multiplying the zig-zag ordered quatezed DCT coefficients) needed to compute the membership degrees in the compressed domain using the trapezoidal function:

$$K_s = [k_1^s, k_2^s] = f_{tr}^{DCT}(a, b, c, d, u_{dct}(0,0)) = \begin{cases} [0 \ 0], & \text{if } u_{dct}(0,0) \in [-128, a-128) \\ \left[\frac{1}{b-a}, \frac{128-a}{b-a} \right], & \text{if } u_{dct}(0,0) \in [a-128, b-128) \\ [1 \ 1], & \text{if } u_{dct}(0,0) \in [b-128, c-128) \\ \left[\frac{-1}{d-c}, \frac{-128+d}{d-c} \right], & \text{if } u_{dct}(0,0) \in [c-128, d-128) \\ [0 \ 0], & \text{if } u_{dct}(0,0) \in [d-128, 128) \end{cases} \quad (23)$$

Therefore, we will have:

$$\begin{aligned} K_{Dark} &= [k_1^{Dark}, k_2^{Dark}] = f_{tr}^{DCT}(0,0, T_1, T_2, u_{dct}(0,0)) \\ K_{Gray} &= [k_1^{Gray}, k_2^{Gray}] = f_{tr}^{DCT}(T_1, T_2, T_2, T_3, u_{dct}(0,0)) \\ K_{Bright} &= [k_1^{Bright}, k_2^{Bright}] = f_{tr}^{DCT}(T_2, T_3, 255, 255, u_{dct}(0,0)) \end{aligned} \quad (24)$$

Notice that: $u_{dct}(0,0)$ is the corresponding DC coefficient of the 8×8 pixels block of the matrix U .

Differently from the contrast enhancement using the INT operator, the fuzzy algorithm Takagi-Sugeno implies the comparison with more than one threshold, which is much more difficult. For each of the 8×8 pixels blocks the comparison with the thresholds in the compressed domain is done on the DC coefficient only, since it is reasonable to assume that a block level classification is likely to correctly place all the pixels in the block on the correct subrange of the membership function (where it is piece wise linear). This is valid for the blocks with moderate frequency content.

Once we have established in this fashion, for a certain pixels block, which of the 5 cases given by equation (23) seems more suitable to apply for all grey levels in the block, we directly compute the 3 vectors K_s ($s \in \{Dark, Gray, Bright\}$), which will be used to compute the 3 membership degrees of each pixel in the block to the fuzzy sets *Dark*, *Gray* and *Bright* in a single step for the entire block, using a matrix-vector formulation.

If one denotes by $D^s[8 \times 8]$, $\forall s \in \{Dark, Gray, Bright\}$, the matrices of membership degrees of the 64 grey levels in the 8×8 pixels block to the 3 input fuzzy sets, then any element in D^s is given by: $d^s(i, j) = \mu_s(u(i, j))$, $\forall i, j = 0, 1, \dots, 7$.

Furthermore, for all the elements in the block, the function μ_s is assumed to have the same linear form, generally speaking, $\mu_s(u(i, j)) = c_m \cdot u(i, j) + c_a$, with: c_m, c_a scalar constants for the multiplication and addition. So,

$$d^s(i, j) = c_m \cdot u(i, j) + c_a, \forall i, j = 0, 1, \dots, 7. \quad (25)$$

where: $C_a[8 \times 8]$, $c_a(i, j) = c_a$, $\forall i, j = 0, 1, \dots, 7$.

Applying a DCT, quantization, zig-zag scanning on both sides of the equation (25), and denoting the resulting vector obtained from D^s by $D_{dct}^s[1 \times 2N]$, one gets its form as:

$$D_{dct}^s = c_m \cdot U_{dct} + C_a^{dct}, \quad (26)$$

where using the notations in equation (23):

- $C_a^{dct}[1 \times 2N]$, $C_a^{dct} = [k_2^s \ 0 \ 0 \ \dots \ 0]$, and
- $c_m = k_1^s$, computed previously.

The fuzzy rule-based contrast enhancement algorithm, from equation (8) taking in account the equation (9), reformulated in the compressed domain as a block level processing, can be described by the following formula (we denote the DCT matrix for the 8×8 enhanced block from the image with the $U_{dct, Int}[1 \times 128]$):

$$U_{dct, Int} = D_{dct}^{Dark} \cdot (I_o^d - 128) + D_{dct}^{Gray} \cdot (I_o^g - 128) + D_{dct}^{Bright} \cdot (I_o^b - 128). \quad (27)$$

The adaptive implementation was used for processing: the decompression was performed for the blocks with high energy and pixel level processed using the equation (8); for the blocks with uniform variances the processing is done directly in the compressed domain, using the equation (27).

We can see in the experimental results that the visual effect is more powerful using Takagi-Sugeno fuzzy algorithm comparing with results obtained using INT operator for contrast enhancement. The problem in Takagi-Sugeno algorithm was that we have two thresholds and this leads to a bigger e_{thd} value (for not having block artifacts), which means more blocks decompressed. But, how is shown in Table 2, the *EffBlocks* is still less than 20% (blocks decompressed and processed at pixel level).

The algorithm was tested on different images having different statistics, with different contrast factors and different average luminance, collected from different sources. The experimental results show a much better computational efficiency, compared to the standard processing method, which needs a total decompression of the image.

Results of enhanced images using the adaptive algorithm are presented in next figures: 14, 15, 16, 17 below. In Fig.14 we have the original image *frog.jpg* with the histogram from Fig.13.a and the enhanced image with the histogram in Fig.13.b.

Fig. 13. a. Input membership function superimposed on the DC histogram of *frog.jpg*; b. DC histogram of *frog.jpg* after fuzzy contrast enhancement.

Fig. 14. Original and enhanced image *frog.jpg*

Fig. 15. Original and enhanced image *keyboard.jpg*

Fig. 16. Original and enhanced image *Cars.jpg*

Fig. 17. Original and enhanced image *Monarch.jpg*

Image	e_{thd}	T_1	T_2	T_3	EffBlocks [%]	MSE
<i>frog.jpg</i>	5	66	104	166	20	1.9
<i>woman.jpg</i>	5	78	141	191	15	0.44
<i>Lena.jpg</i>	5	33	126	223	20.1	0.004
<i>Lena.jpg</i>	10	33	126	223	15.03	0.005
<i>Lena.jpg</i>	20	33	126	223	10.41	0.02
<i>keyboard.jpg</i>	10	0	34	78	16.02	0.01
<i>eye.jpg</i>	10	3	117	199	14.6	1.78
<i>medical.jpg</i>	10	5	90	208	10.07	0.93
<i>butterfly.jpg</i>	10	32	117	207	10.58	0.011
<i>cars.jpg</i>	5	106	161	200	8.16	0.5
<i>bee.jpg</i>	10	7	170	48	15.4	0.55
<i>fingerPrint.jpg</i>	10	45	117	190	18	0.29
<i>children.jpg</i>	10	60	128	228	13.1	0.28
<i>monarch.jpg</i>	10	29	101	223	16.15	0.63
<i>lisa.jpg</i>	10	33	88	223	14.9	1.5
<i>girl.jpg</i>	10	76	104	174	7.24	2.1

Table 2. Results for different values e_{thd}

6. Conclusion

An adaptive approach for image enhancement using fuzzy sets theory and fuzzy logic, in the class of digital image processing in the compressed domain, is presented in this chapter. Fuzzy rule-based contrast enhancement and fuzzy intensification operator are by default nonlinear, with one or few threshold comparisons as main nonlinearity; they are not straightforward applicable on the JPEG bitstream. We have presented a strategy for implementation of the brightness thresholding, for which an adaptive solution which takes into account the frequency content of each block in the JPEG compress domain is suggested. To obtain the best enhancement possible with these approaches, we chose to use, in the grey levels fuzzification step of the methods, a parameterized membership function, with the parameters adapted to the grey level statistics in the image. The selection of the fuzzy set membership function parameters was done directly in the compressed domain, using the histogram of the DC coefficients of the compressed blocks as an approximation of the grey level statistics of the image. The algorithm was tested on different images having different statistics, with different contrast factors and different average luminances. The algorithms are applied only on the luminance component, but they can be applied for color image enhancement as well, with no change of the chrominance components. The experimental results show a better computational efficiency, compared to the standard processing method (which needs a total decompression of the image), practically at no processing error as compared to the reference algorithm (i.e., the same fuzzy image enhancement algorithm formulated at pixel level).

We also provide a principled formulation of a procedure to select the optimal value of the threshold for the AC energy content of each block, in order to guarantee the best possible computational efficiency with no visually noticeable processing error; embedding this principle in a fully automatic procedure and its verification for several types of images will make the object of our future work.

7. References

- Agaian, S.S.; Silver, B.; Panetta, K.A. (2007). Transform Coefficient Histogram-Based Image Enhancement Algorithms Using Contrast Entropy, *IEEE Transactions On Image Processing*, Vol. 16, No. 3, March 2007, pp.741-758.
- An, K.; Ni, Q.; Sun, J. (2004). A contrast enhancement method for compressed images, *IEICE Electronic Express* Vol. 1, No. 18, pp. 582-587.
- Bezdek, J.C.; Keller, J.M.; Krishnapuram, R.; Pal, N.R. (1999). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Boston.
- Florea, C.; Vlaicu, A.; Gordan, M.; Orza, B.; (2009). Fuzzy intensification operator based contrast enhancement in the compressed domain, *Applied Soft Computing*, Elsevier, ISSN: 1568-4946.
- Gonzales, R.C.; Woods R.E, (2008). *Digital Image Processing, 3rd edition*, Prentice Hall, ISBN 9780131687288, United States.
- Gunturk, B.K.; Altunbasak, Y. (2002). Multiframe Resolution-Enhancement Methods for Compressed Video, *Signal Processing Letters*, IEEE, Vol. 9, No. 6, pp. 170-174.
- Kim, J.; Peli, E. (2003). MPEG based image enhancement for people with low vision, *Soc for Information Display, Digest of Technical Papers*, pp. 1156-1159.

- Mezaris, V.; Kompatsiaris, I.; Kokkinou, E.; Strintzis, M.G. (2003). Real-time compressed-domain spatiotemporal video segmentation, *Proceedings of Third International Workshop on Content-Based Multimedia Indexing*, France, 2003, pp. 373-380.
- Pal, S.K.; King, R.A. (1981). Image Enhancement Using Smoothing with Fuzzy Sets, *IEEE Transaction Systems Man Cybernet*, Vol. SMC-11, No. 7, pp. 494-501.
- Popa, C.; Gordan, M.; Vlaicu, A.; Orza, B.; Oltean, G. (2008). Computationally efficient algorithm for fuzzy rule-based enhancement on JPEG compressed color images, *Transactions on Signal Processing*, WSEAS, Vol. 4, No.5, May 2008, pp. 310 – 319, ISSN: 1790-5052.
- Pedrycz, W.; Gomide, F. (1998). An Introduction to Fuzzy Sets: Analysis and Design, *The MIT Press*, 1998.
- Sangkeun, L. (2006). Content-based image enhancement in the compressed domain based on multi-scale alpha-rooting algorithm, *Pattern Recognition Letters*, Vol. 27, No. 10, 15 July 2006, pp. 1054-1066.
- Smith, B.C.; Rowe, L.A. (1993). A New Family of Algorithms for Manipulating Compressed Images, *IEEE Computer Graphics and Applications*, vol.13, (no.5), pp: 34-42, Sept. 1993.
- Smith, B.C. (1995). A survey of compressed domain processing techniques, *Proceedings of NSF Workshop on Reconnecting Science and Humanities in Digital Libraries*, Oct. 1995.
- Smith, B.C.; Rowe, L.A. (1996). Compressed Domain Processing of JPEG-encoded Images, *Real-Time Imaging Journal*, Vol. 2, July, pp 3-17.
- Tang, J. (2004). A contrast based image fusion technique in the DCT domain, *Digital Signal Processing*, Vol. 14, No. 3, pp. 218-226.
- Tang, J.; Peli, E.; Acton, S. (2003). Image enhancement using a contrast measure in the compressed domain, *Signal Processing Letters*, IEEE, Vol. 10, No. 10, Oct. 2003, pp. 289 – 292.
- Tizhoosh, H.R. (2000). Fuzzy Image Enhancement: An Overview, In: Kerre, E., Nachtegaal, M. (Eds.): "Fuzzy Techniques in Image Processing", *Studies in Fuzziness and Soft Computing*, Springer, pp. 137-171, ISBN: 3-7908-1304-4.
- Triantafyllidis, G.A.; Varnuska, M.; Sampson, D. ; Tzovaras, D. ; Strintzis, M.G. (2003). An efficient algorithm for the enhancement of JPEG-coded images, *Computers and Graphics*, Vol. 27, No. 4, August 2003, pp. 529-534, Publisher: Elsevier.
- Zadeh, L.A. (1965). Fuzzy sets, *Information and Control*, Vol. 8, No. 3, pp. 338-353.

Advances in Computer Science and IT

Edited by D M Akbar Hussain

ISBN 978-953-7619-51-0

Hard cover, 420 pages

Publisher InTech

Published online 01, December, 2009

Published in print edition December, 2009

The book presents some very interesting and excellent articles for this divergent title. The 22 chapters presented here cover core topics of computer science such as visualization of large databases, security, ontology, user interface, graphs, object oriented software developments, and on the engineering side filtering, motion dynamics, adaptive fuzzy logic, and hyper static mechanical systems. It also covers topics which are combination of computer science and engineering such as meta computing, future mobiles, colour image analysis, relative representation and recognition, and neural networks. The book will serve a unique purpose through these multi-disciplined topics to share different but interesting views on each of these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Camelia Florea, Aurel Vlaicu, Mihaela Gordan and Bogdan Orza (2009). Adaptive Implementation of Nonlinear Fuzzy Image Enhancement Algorithms in the Compressed JPEG Images, *Advances in Computer Science and IT*, D M Akbar Hussain (Ed.), ISBN: 978-953-7619-51-0, InTech, Available from:
<http://www.intechopen.com/books/advances-in-computer-science-and-it/adaptive-implementation-of-nonlinear-fuzzy-image-enhancement-algorithms-in-the-compressed-jpeg-image>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen