Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 161 – 170
ISSN 1896-7094

Fuzzy contrast enhancement for images in the compressed domain

Camelia Popa, Aurel Vlaicu, Mihaela Gordan and Bogdan Orza

Centre for Multimedia Technologies and Distance Education,
Technical University of Cluj-Napoca,
Daicoviciu 15, Cluj-Napoca, Romania
{Camelia.Popa, Aurel.Vlaicu, Bogdan.Orza}@com.utcluj.ro,
Mihaela.Gordan@bel.utcluj.ro
http://www.ctmed.utcluj.ro

Abstract. Our objective is to investigate image processing directly in the compressed domain, without full decompression. Compressed domain image processing algorithms provide a powerful computational alternative to classical (pixel level) based implementations. The field is just emerging and the algorithms reported in the literature are mostly based on linear arithmetic operations between pixels. In this paper, the problem of implementing a non-linear operator, using compressed domain processing is addressed. A new algorithm for digital image enhancement, using fuzzy theory, adapted to the frequency content of each coefficient block in the DCT (Discrete Cosine Transform) encoded JPEG image is developed and proposed.

1 Introduction

The image storage in bitmap format is not so 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 space needed and a better capacity to send information (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. This time economy is due 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 that processing images in the compressed domain means that there are fewer data to process.

Linear processing is easier to implement in the compressed domain because there are no problems to process an image involving linear operations, such as the adding with 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 if much faster due to the reduced number of coefficients (instead of 64 in pixel level processing, corresponding to a 8x8 block of pixels, in the compressed domain just 20% of the pixels are processed).

Implementations of image and video sequence processing in the compressed JPEG/MPEG domain are already presented in the literature. Tang [5] – defined an algorithm based on the contrast measure as the ratio of high-frequency content and low-frequency content in the bands of the DCT matrix. Kebin [6] – use the Tang algorithm but he enhanced each block according to its block classification: smooth block or high activity block. Kim and Peli [9] – developed an image enhancement algorithm, MPEG based, for people with low-vision – contrast enhancement is performed by modifying the quantization matrices for Inter and Intra frames. Sangkeun Lee [4] – use a basic concept of the Retinex theory for enhance the images.

In the paper, a processing algorithm in the compressed domain is presented. The enhancement of images uses fuzzy techniques, involving nonlinear and thresholding operations. The techniques for image enhancement in the compressed domain are based on modifying the frequency transform of an image.

2 Image enhancement using fuzzy algorithm

Image enhancement involves processing an image in order to make it satisfactory to viewers. Our goal is to provide a fast and accurate alternative implementation in the compressed domain, to the classical fuzzy image enhancement algorithm, based on global threshold, which is usually performed at pixel level. 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.

Further we will present the basic steps used to compress/decompress the JPEG images [1, 12]. The image is first divided into 8x8 blocks, and each 8x8 block is computed. After that, a DCT is applied on each block and results the DCT coefficients which are quantized. Many small coefficients, usually high frequency, are quantized to zero. The next step is zig-zag scan of the DCT matrix followed by RLE (Run Length Encoding) coding and entropy coding (Huffman coding). In the decoder, the compressed image is decoded and then dequantized and inverse-DCT-transformed.

There are two ways to enhance the images compressed with JPEG: the compressed domain processing (no decompression/compression) and the pixel level processing (enhance the image after decompression):

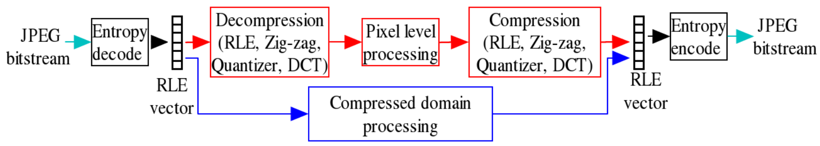


Fig. 1. The two ways to process the image compressed JPEG

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

2.1 Contrast enhancement using an INT-Operator from fuzzy theory

In recent years, many researchers have applied the fuzzy set theory [2] to develop new techniques for contrast improvement. A basic fuzzy algorithm for image enhancement, using a global threshold, will be briefly recalled. Let us consider a gray level digital image, represented by the gray level values of the pixels scaled to the range [0;1]. Let *l* be any gray level of a pixel in this digital image, $l \quad [0;1]$.

Contrast improvement is a basic point processing operation aiming mainly to maximize (increase) the dynamic range of the image. A higher contrast in an image can be achieved by darkening the gray level in the lower luminance range (typically under 0.5 on a [0;1] scale) and brightening the ones in the upper luminance range [13]. This processing usually implies the use of a non-linear function; a typical form of such a function could be the one presented in Figure 2. A possible mathematical expression of such a nonlinear function, $Int:[0;1] \quad [0;1]$, is as follows:

$$Int(l) = \begin{cases} 2\cdot l^2, \ if \ \ 0 \le l \le 0.5 \\ 1-2\cdot(1-l)^2, \ if \ \ 0.5 \le l \le 1 \end{cases}, \quad l \in [0;1]. \tag{1}$$

The expression in (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.

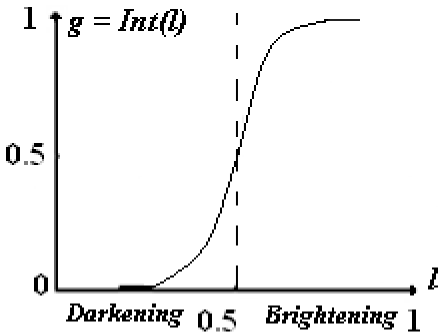


Fig. 2. Fuzzy intensification

Let us denote the resulting gray levels in the contrast enhanced image by *g* given by:

$$g = Int(l), \quad \forall \quad l \in [0;1]. \tag{2}$$

Thus, the contrast enhanced image will have gray levels of its pixels given by the nonlinear point-wise transformation in figure 2, applied to the original gray level image.

2.2 Convolution of the image in the compressed domain

In order to implement a fuzzy algorithm in the compressed domain, linear operations and nonlinear operations are necessary. For example, the square of an image is needed, which is a nonlinear operation and can be obtained using the multiplication algorithm of two images, in the compressed domain, developed by B. Smith [1].

The following notation will be introduced: considering the original HxW gray level image, divided into 8x8 pixels blocks (as in the typical process of JPEG encoding), we denote any such block of pixels by the matrix $U[8x8]$, containing the gray level values of the pixels scaled to the range [0;1]:

$$U = \{u(i, j) \in [0,1],\ where\ i = 0,\ 1,...,\ 7,\ \ j = 0,\ 1,...,\ 7\} \tag{3}$$

Consider $U_{dct}[8x8]$ to be the matrix of the DCT coefficients of any 8x8 pixels block from the original image U (directly available in the JPEG image coding, as shown in figure 1). In order to apply the contrast enhancement algorithm described above, we need to compute the square of each pixel luminance in the 8x8 block, in the compressed domain. That is, we should obtain the DCT of the 8x8 block of squared luminance. We denote this DCT matrix by $U_{dct,sq}[8x8]$. According to [1], $U_{dct,sq}[8x8]$ can be obtaining from $U_{dct}[8x8]$ as follows (using the notations from [1]):

$$\begin{aligned}
&U_{dct,sq}(x_1, x_2) = \frac{1}{4\cdot Q(x_1, x_2)} \sum_i \sum_j C(i, x_1)\cdot C(j, x_2)\cdot [u(i, j)]^2 = \\
&= \sum_{y_1, y_2, w_1, w_2} U_{dct}(y_1, y_2)\cdot U_{dct}(w_1, w_2)\cdot W_Q(y_1, y_2, w_1, w_2, x_1, x_2) \\
&where: \\
&W_Q(y_1, y_2, w_1, w_2, x_1, x_2) = \frac{Q(y_1, y_2)\cdot Q(w_1, w_2)}{256\cdot 64\cdot Q(x_1, x_2)} W(x_1, y_1, w_1)\cdot W(x_2, y_2, w_2) \\
&with:\quad W(x, y, w) = \sum_i C(i, x)\cdot C(i, y)\cdot C(i, w) \\
&C(i, x) = A(x)\cos\frac{(2\cdot i+1)\cdot x\cdot \pi}{16};\quad A(x) = \begin{cases} \frac{1}{\sqrt{2}}, for\ x = 0 \\ 1, for\ x \neq 0 \end{cases}
\end{aligned} \tag{4}$$

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

2.3 Fuzzy image enhancement in the compressed domain

We should take into account that, in the compressed domain, prior to applying the DCT based compression of the JPEG algorithm, the luminance values are scaled from the [0, 255] range to the [-128, 127] range, symmetrical towards 0. Consequently, the resulting DC coefficient will also be scaled towards 0 and the threshold for the fuzzy algorithm in the compressed domain should have the value 0 (instead of 0.5, in the case of the fuzzy algorithm applied on the pixel level).

The algorithm is applied on the DCT coefficients and, in this case only, the DC coefficient will be compared with the threshold, for each and every 8x8 block. The proposed algorithm in order to compute the luminance values in the enhanced image $IntU_{dct}$ is given by the following formula:

$$IntU_{dct}(i,j)=\begin{cases}U_{dct,sq}(i,j)+2\cdot U_{dct}(i,j)-128, & if \quad U_{dct}(0,0)\le 0\\ 2\cdot U_{dct}(i,j)-U_{dct,sq}(i,j)-128, & if \quad U_{dct}(0,0)>0\end{cases} \tag{5}$$

notice that : $U_{dct}(0,0)$ is the DC coefficient coresponding of the 8x8 pixels block of the DCT matrix

where : $i = 0,1,...,7;\ j = 0,1,...,7;$

In case the processing is done in the compressed domain, the threshold is done once for each 8x8 block, meanwhile in pixel level processing, it is performed separately for each pixel inside the block (64 operations).

Although, the use of the threshold only once for an 8x8 block can lead to an eronate classification, because applying the same formula for each pixel from an 8x8 block could be wrong. This is the case of the blocks with a highly non-uniform luminance distribution, which is with large AC energy content. This disadvantage can be avoided by total decompression of all the blocks containing higher energy factor, and by processing them at the pixel level. In this way, processing in the compressed domain is much more efficient, because the number of blocks needing total decompression is small, as it can be seen in the experimental part (table 1).

The fuzzy image enhancement algorithm is defined for each 8×8 pixels block of the JPEG image, as follows:

1) The average amount of the AC coefficients energy from DCT block, denoted E_{AC}, is computed as:

$$E_{AC}=\frac{\sum_{i,j}^{7}\left|U_{dct}(i,j)\right|^{2}}{64} \tag{6}$$

2) If $E_{AC} < e_{th}$ (where the e_{th} value represents the selection threshold between the uniform block and the blocks with a significant number of details) then the block can be considered to have uniform luminance and it will be processed in the compressed domain.
3) If $E_{AC} > e_{th}$, then the block has a significant content of details and it will be decompressed, every pixel from the block being processed separately, using classical enhancement techniques.

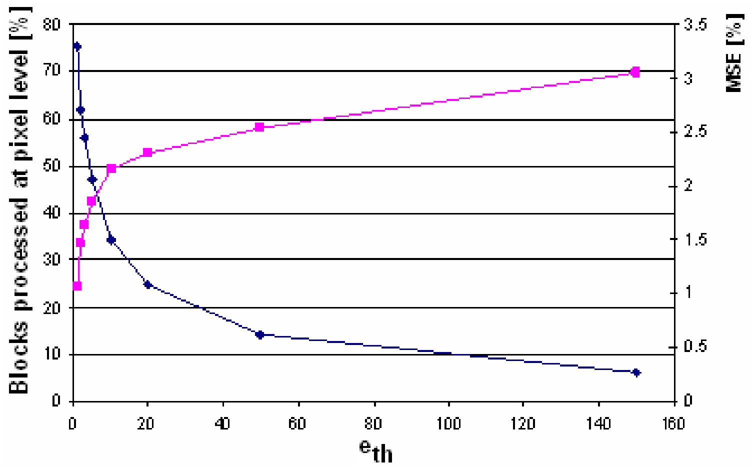


Fig.3. e_{th} influence for an image (e.g. Cars.jpg)

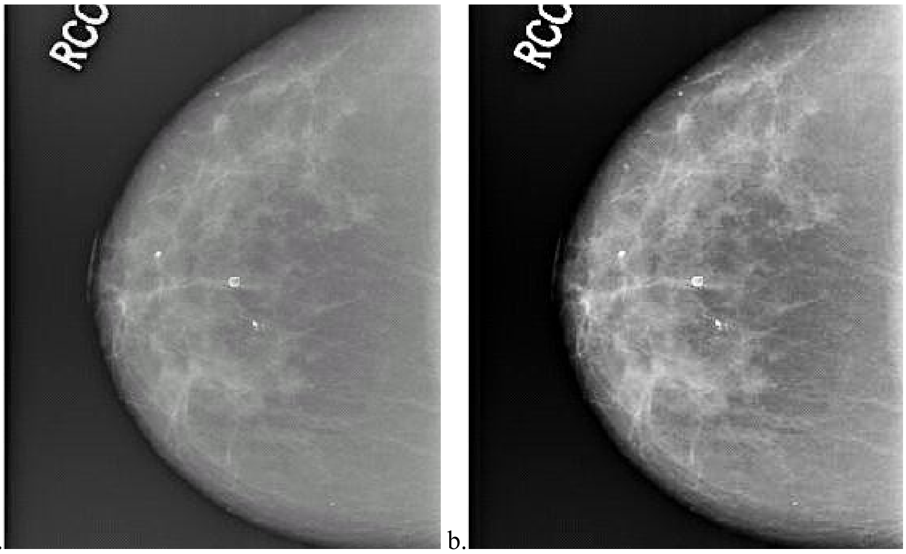
The e_{th} value is chosen from the experiments, taking into account the image type. A very small e_{th} value will always lead to process images of a very good quality, but the number of processed blocks in the compressed domain will be quite small, so the complexity of the computing algorithm is not significantly smaller, compared to the direct processing on the pixel. An appropriate e_{th} value will lead to the increase of the number of blocks processed in the compressed domain and, in this way we can obtain a fast fuzzy algorithm for image enhancement.

3 Experimental results

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, relatively to:

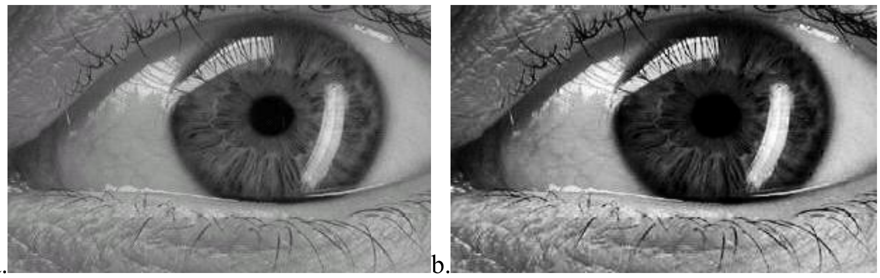
1) the quality of the image resulted from the fuzzy enhancement process, compared to the fuzzy enhancement in the spatial domain (on pixel level);
2) the estimated efficiency of the processing method with 8x8 blocks directly in the compressed domain, the numbers of blocks processed at pixel level;

Results of enhanced images using the proposed algorithm are presented in the next figures:



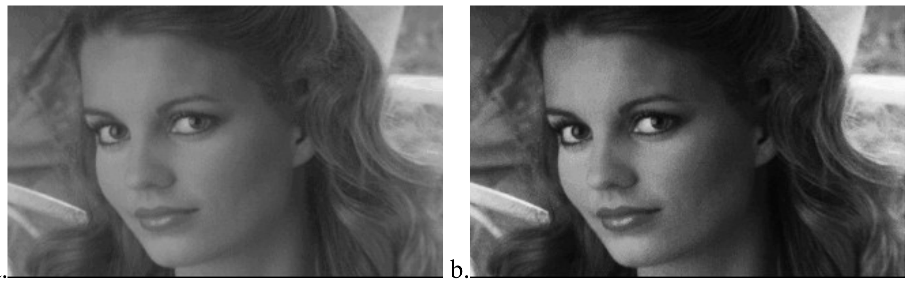
a. b.

Fig. 4. a. Original image Medical.jpg; b. Enhanced image using the proposed algorithm



a. b.

Fig.5. a. Original image Eye.jpg; b. Enhanced image using the proposed algorithm



a. b.

Fig.6. a. Original image Lisa.jpg; b. Enhanced image using the proposed algorithm

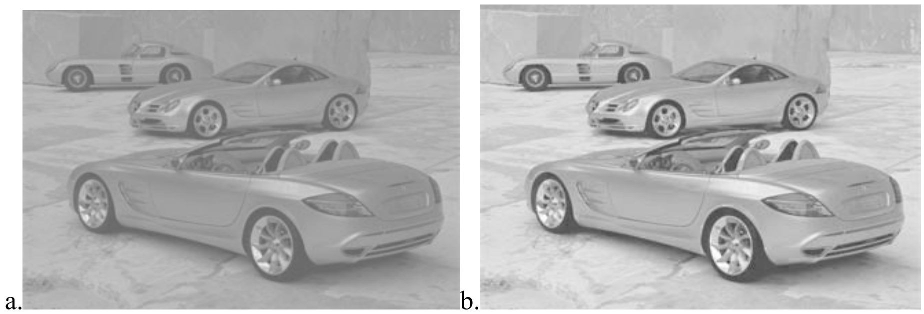


Fig.7. a. Original image Cars.jpg; b. Enhanced image using the proposed algorithm

Our goal is to obtain the same performance as with the standard pixel-level enhancement, but with a significant reduction of the computational complexity. We use as quality performance measure the MSE (Mean Squared Error), between the pixel level process images and the images processed in the compressed domain with our algorithm.

Table 1 . Results for different values e_{th}

Image name	e_{th}	MSE [%]	**Number of blocks processed at pixel level**	**Total number of blocks 8x8**	**Blocks processed at pixel level [%]**
Mouse.jpg	8	9.803	46	1000	4.6
Lisa.jpg	3	5.38	175	1000	17.50
Cars.jpg	10	2.55	61	1200	5.08
Lena.jpg	5	5.716	633	4096	15.45
Nature.jpg	10	1.65	31	1728	1.79
Eye.jpg	10	5.98	103	912	11.29
Medical.jpg	10	9.53	89	1768	5.03

In figure 8 we compare our image enhancement algorithm with other two algorithms: the global histogram equalization and the contrast enhancement by Tang [5]:

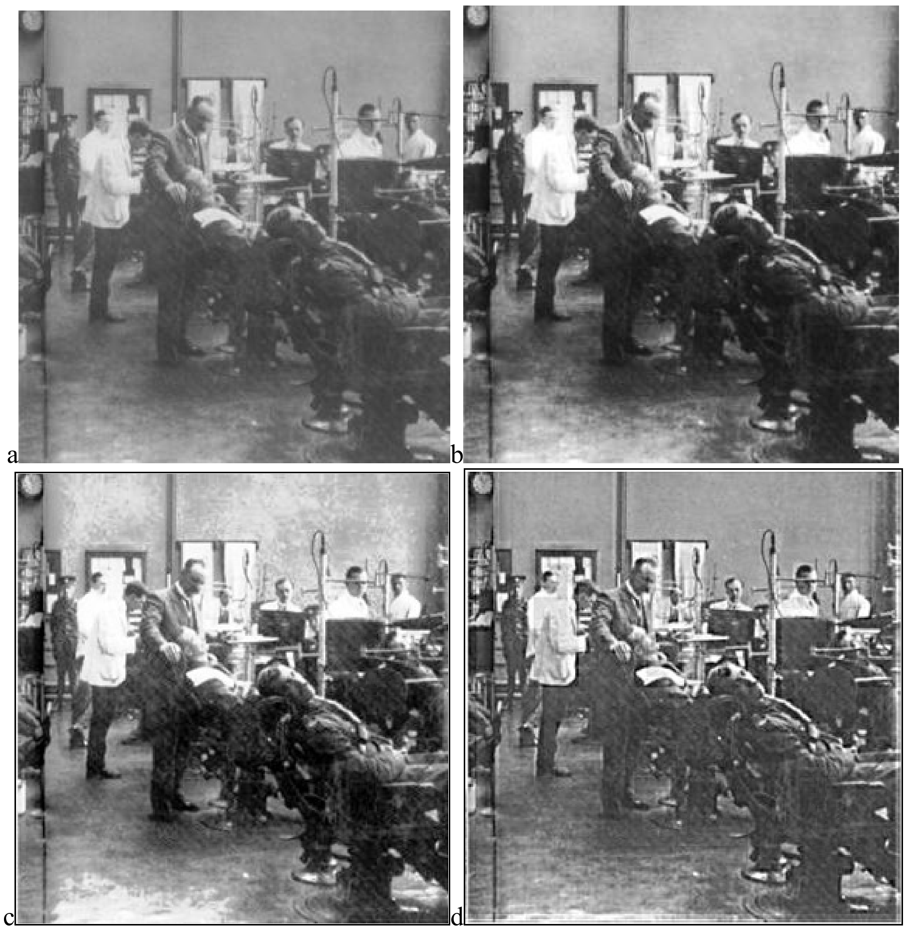


Fig.8. a. Original image; b. enhanced image using the proposed algorithm; c. enhanced image using global histogram equalization; d. enhanced image using contrast enhancement

4 Conclusion

A new algorithm has been proposed and software implemented, in the class of digital image processing in the compressed domain, dedicated to image enhancement using fuzzy theory. The algorithm was tested on different images having different statistics, with different contrast classes and medium luminance. The experimental performance results show a better computation efficiency, compared to the standard processing method (which needs a total decompression of the image), in the same conditions of preservation for the quality of the image.

The algorithm can be optimized with an automatic selection method of the threshold, in order to classify the blocks according to their content of AC, e_{th} energy, so that the best compromise between processing speed and quality of the resulted image is obtained. This issue is currently under investigation, as well as the application of this algorithm to other compressed domain processing of digital images (e.g. segmentation).

References

1. Brian Smith, Lawrence A. Rowe, *A New Family of Algorithms for Manipulating Compressed Images ,* IEEE Computer Graphics and Applications, Sept. 1993, vol.13, (no.5), pp: 34-42.
2. Pal S. K., King R. A., *Image Enhancement Using Smoothing with Fuzzy Sets,* IEEE Trans. Systems Man Cybernet, Vol. SMC-11, No. 7, 1981, pp. 494-501.
3. G. A. Triantafyllidis, M. Varnuska, D. Sampson, D. Tzovaras, M. G. Strintzis, *An efficient algorithm for the enhancement of JPEG-coded images,* Computers and Graphics, Volume 27(4), August 2003, pp. 529-534; Publisher: Elsevier
4. J. Tang, *A contrast based image fusion technique in the DCT domain,* Digital Signal Processing, 2004, Volume 14(3), pp. 218-226.
5. Lee Sangkeun, *Content-based image enhancement in the compressed domain based on multi-scale alpha-rooting algorithm,* PRL (27), No. 10, 15 July 2006, pp. 1054-1066.
6. J. Tang, E. Peli, S. Actonm *Image enhancement using a contrast measure in the compressed domain,* - Signal Processing Letters, IEEE, Vol. 10, Issue: 10, Oct. 2003, pp. 289 – 292.
7. J. Kim, E. Peli, *MPEG based image enhancement for people with low vision,* Soc for Information Display, Digest of Technical Papers, 2003, pp. 1156-1159.
8. V. Mezaris, I. Kompatsiaris, M. G. Strintzis, *Compressed-domain object detection for video understanding,* Proc. Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS), April 2004, Lisbon, Portugal.
9. V. Mezaris, I. Kompatsiaris, E. Kokkinou and M. G. Strintzis, *Real-time compressed-domain spatiotemporal video segmentation,* Proc. Third International Workshop on Content-Based Multimedia Indexing, France, 2003, pp. 373-380.
10. Pedro Miguel Fonseca and Jan Nesvadba – *Face Tracking in the Compressed Domain,* EURASIP Journal on Applied Signal Processing, vol. 2006, artic. ID 59451, 11 pg, 2006.
11. B. K. Gunturk, Yucel Altunbasak, *Multiframe Resolution-Enhancement Methods for Compressed Video,* Ieee Signal Processing Letters, Vol. 9, No. 6, June 2002.
12. Brian C. Smith, *A survey of compressed domain processing techniques,* in Proc. NSF Workshop on Reconnecting Science and Humanities in Digital Libraries, Oct. 1995.
13. Gonzales R. C., Woods R. E, *Digital Image Processing,* Adison Wesley, 1993.