Wavelet–PCA–Based Compression Method for Color Images

Shohreh Kasaei [1] and Arash Abadpour [2]
[1] Computer Engineering Department, Sharif Univ. of Tech., Tehran, Iran, email: skasaei@sharif.edu
[2] Mathematics Science Department, Sharif Univ. of Tech., Tehran, Iran, email: abadpour@math.sharif.edu

Abstract

From the birth of multi–spectral imaging techniques, there has been a tendency to consider and process this new type of data as a set of parallel gray–scale images, (instead of an ensemble of an n–D realization). Although, even now, some researchers make the same assumption, it is proved that using vector geometries leads to more realistic results. In this paper, using a proposed PCA–based eigenimage extraction method, incorporated with a wavelet–based grayscale image compression algorithm, a new compression method is proposed. The paper includes comprehensive performance analysis of the proposed eigenimage extraction and compression methods.

I. Introduction

The fundamental goal of image compression is to obtain the best possible quality, for the available storage or communication capacity. Compression of color images is still a challenge due to its high dimensionality. Color is one of the most important tools for object discrimination by human observers, but it is overlooked in the past [1]. Discarding the intrinsic characteristics of color images, (as *vector geometries* [2]), some researchers have assumed color images as *parallel* gray–scale images [3], [4], [5], [6]. It is proved that *principle component analysis*(PCA) is an appropriate vectorial descriptor for natural color images [7], [8], [9].

The early approach towards color image compression, has been based on decorrelating the color planes using some linear or nonlinear invertible coordinate transformation (*e.g.*, YC_bC_r [10], YIQ [11], and YUV [12]), and then performing one of the standard gray–scale compression methods (like *differential pulse code modulation*(DPCM) [13] or transform coding [14]) on each plane, separately (see also [15]). This approach is inefficient, because none of the available color spaces are able to completely decorrelate the planes in a real image. In [16] using the PCA approach in the neighborhood pixels, the author discusses the idea of separating the spatial and spectral compression stages. As the paper proves, the maximum theoretical compression ratio for an ideal spectral compression method is $1:3$. The main shortcoming of the method in [16] is neglecting the fact that in non–homogenous regions, the PCA does not perform energy compaction [17]. In [17] the author combines the spatial and the spectral information to reach a higher compression ratio. Although, the method is based on expensive computation, the *peak signal to noise ratio* ($PSNR$) results are desperate. The main shortcoming of the method in [17], is the block–wise artifacts produced after decompression.

In this paper, we apply a tree decomposition method using a proper color homogeneity criterion to cut a given image into homogenous patches. Also, some mathematical tools are developed to introduce the proposed eigenimage extraction method and perform. Then, the proposed eigenimage extraction method is completely analyzed. The main contribution of this paper is a new color image compression method which incorporates the proposed eigenimage extraction technique with a novel wavelet–based grayscale image compression method.

Quad–tree decomposition is the well–known method for splitting an image into homogenous sub-blocks, resulting in a very coarse, but fast segmentation [18]. Defining the whole image as a single block, the method is performed according to a problem-specific *homogeneity criterion*. Some works on the generalization of quad–tree decomposition has been performed, regarding the dimension [19] and the shape [20] of the blocks. Using rectangles as the sub–blocks is known to have many benefits. Thus, here we use the *bi–tree* decomposition method [21] which uses rectangular blocks in a more adaptive framework, compared to the original quad–tree decomposition method [18] (for comprehensive comparison of bi–tree and quad–tree decomposition method refer to [21]).

The rest of this paper is organized as follows: Section II discusses the previous works on PCA–based color image processing. Sections III-A and III-B introduce the basis vector polarization and block–wise interpolation schemes which in addition with the concepts introduced in Section II are used in Section III-C to propose a solution to the eigenimage extraction problem. SectionIII-D state the proposed eigenimage–based compression technique. Section IV contains the experimental results and discussions, and finally, Section V concludes the paper.

II. Previous Works on PCA–based Color Processing

In [7], the authors proposed to use the error made by neglecting the two least important principal components (the second and the third) as a likelihood measure for color vectors. As such, the *linear partial reconstruction error* (LPRE) distance of

vector $\vec{c}$ to cluster r is defined as,

$$\tau_r(\vec{c}) = \|\vec{v}^T(\vec{c}-\vec{\eta})\vec{v} - (\vec{c}-\vec{\eta})\|, \tag{1}$$

where $\vec{\eta}$ and $\vec{v}$ denote the expectation vector and the direction of the first principal component of cluster r, and $\|\vec{x}\|$ is the normalized L_1 norm defined as $\|\vec{x}\| = \sum_{i=1}^{3}|x_i|/3$. In [7], the authors proposed to use the following stochastic margin to compute the homogeneity of the selected cluster r,

$$\|f\|_{r,p} = \arg_e\{P_{\vec{x}\in r}\{f(\vec{x}) \le e\} \ge p\}, \tag{2}$$

where p is the inclusion percentage and $P_{\vec{x}\in r}\{f(\vec{x}) \le e\}$ denotes the probability of x being less than or equal to e. Equation (2) leads to the definition of the normalized LPRE measure as,

$$\tilde{\tau}_{r,p}(\vec{c}) = \frac{\tau_r(\vec{c})}{\|\tau_r\|_{r,p}}. \tag{3}$$

It is proved that $\|\tau_r\|_{r,p}$ is a proper homogeneity criterion for usages such as tree decomposition [22], and $\tilde{\tau}_{r,p}(\vec{c})$ serves as a perfect likelihood measure for color images [8]. The comparison of the LPRE–based homogeneity criterion and likelihood measure with the Euclidean and the Mahalanobis–based approaches has proved its absolute superiority over both of them [8]. Also, it is shown that the L_1 norm used in (1) can be converted to the L_2 norm [8].

As mentioned above using any proper homogeneity criterion, the image can be decomposed into a set of non–overlapping blocks. According to the experimental proof of performance of $\|\tau_r\|_{r,p}$, a new tree decomposition method is proposed by the authors [21] that incorporates $\|\tau_r\|_{r,p}$ with a new splitting scheme to reach to a proper tree decomposition method. In the bi–tree decomposition method, starting from the entire image area as a single block, the tree is produced using the homogeneity criterion defined as:

$$h(r) = \begin{cases} 1, & \|\tau_r\|_{r,p} \le \varepsilon_1 \\ 0, & otherwise \end{cases}, \tag{4}$$

where ε_1 is a user-selected parameter, mostly in the range of $[1, 10]$. In this way, any non–homogenous block is split into two non–overlapping blocks according to the rules states below. If the block r is not homogenous enough, rather than the deterministic choice of the sub–blocks in the quad–tree decomposition method, here, r is divided either horizontally or vertically (see Figure 1), in the way that the total non–homogeneity of the result gets the least possible. In this way if,

$$\|e_{r_1}\|_{r_1,p} + \|e_{r'_1}\|_{r'_1,p} < \|e_{r_2}\|_{r_2,p} + \|e_{r'_2}\|_{r'_2,p}, \tag{5}$$

and the depth limitation permits, the block will be split vertically and otherwise (if the depth limitation is met), it is split horizontally.

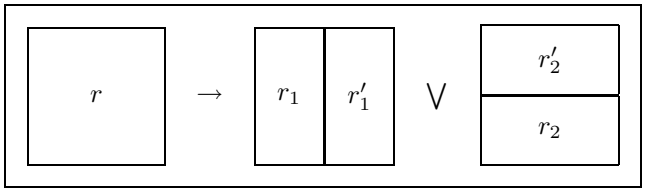


Fig. 1. Bi–tree decomposition method.

In a $W \times H$ image, the *depth* of a $w \times h$ block r is defined as:

$$\varrho_r = \max\left\{\log_2\frac{W}{w}, \log_2\frac{H}{h}\right\}, \tag{6}$$

and no block is permitted to reach to the depth more than a preselected marginal tree depth value ϱ in the range of $[1, 5]$. Note that ϱ_r computes the maximum number of splitting stages needed to produce r out of the entire image.

During the decomposition stage, all the information is saved as a $22 \times N$ matrix called Υ, where N is the number of blocks and each column of Υ consists of x_1, y_1, x_2, y_2, η_1, η_2, η_3, $V_{ij}, i, j = 1, \cdots, 3$ and some reserved parameters. Here, $[\eta_1, \eta_2, \eta_3]^T$ is the expectation vector of color information of the region r and V_{ij}s are the elements of the PCA Matrix V corresponding to block r.

In the new method, the rectangular clipping is reserved while the block shape changes such that it best fits to the image details. Comparing the proposed bi–tree and the quad–tree decomposition methods, bi–tree produces smaller number of blocks that they have almost identical sizes. In the contrary, quad-tree the image more with a more scattered pattern of block sizes [21].

III. Proposed Algorithm

A. Basis Vector Polarization

Consider the space R^n and a set of n basis vectors $\vec{v}_i, i = 1, \cdots, n$. Storing this set of vectors needs n^2 units of memory (when neglecting the redundancy among data). Having in mind that a set of basis vectors are an *ortho–normal* set, the actual needed memory can be reduced. In fact, a set of basis vectors of R^n is a member of R^{n^2}, with n constraints of normality ($\|\vec{v}_i\| = 1, i = 1, \cdots, n$) and $\frac{n(n-1)}{2}$ constraints of orthogonality ($\vec{v}_i \perp \vec{v}_j, i, j = 1, \cdots, n, i \neq j$). Thus, the above–mentioned set of the basis vectors is an unconstrained member of an m–D space, with m equal to:

$$m = n^2 - n - \frac{n(n-1)}{2} = \frac{n(n-1)}{2}. \tag{7}$$

Thus, storing a set of the basis vectors of R^n in $\frac{n(n-1)}{2}$ memory cells contains zero redundancy. To make this representation unique, it is crucial to make the set of the basis vectors *right–rotating* (RR). In 2–D spaces, RR means:

$$(\vec{v}_1 \times \vec{v}_2) \cdot \vec{j} > 0, \tag{8}$$

where, $\times$ and $\cdot$ denote the *outer* and the *inner* products, respectively. In 3–D spaces, RR means:

$$(\vec{v}_1 \times \vec{v}_2) \cdot \vec{v}_3 > 0. \tag{9}$$

Setting $n = 2$ in (7) leads to $m = 1$, which means that any set of RR basis vectors in the xy plane can be specified uniquely by a single parameter (the angle). Similarly, the case of $n = 3$ results in $m = 3$, which is used in this paper. We will prove that the parameters in the 3–D case are angular too. Thus, we call this method of representing a set of basis vectors, the *polarization* method. Here we propose a method for finding these angles.

Consider the three RR vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3$ in R^3. We define the angles θ, ϕ, and ψ as a manipulated version of the well–known set of *Euler* angles. Using $\vec{v}^p$ as the projection of $\vec{v}$ on plane p (*e.g.*, $\vec{v}_1^{xy}$), the three angles are defined as:

$$\theta = \angle(\vec{v}_1^{xy}, [1,0]^T), \phi = \angle((R_\theta^{xy}\vec{v}_1)^{xz}, [1,0]^T), \psi = \angle((R_\phi^{xz}R_\theta^{xy}\vec{v}_2)^{yz}, [1,0]^T). \tag{10}$$

Here, $\angle(\vec{v}, \vec{u})$ denotes the angle between two vectors $\vec{v}, \vec{u} \in R^2$, computed as:

$$\angle(\vec{v}, \vec{u}) = sgn((\vec{v} \times \vec{u}) \cdot \vec{j}) \cos^{-1} \frac{\vec{v} \cdot \vec{u}}{\|\vec{v}\|\|\vec{u}\|}, \tag{11}$$

where $sgn(x)$ is the *signum* function, defined as:

$$sgn(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}. \tag{12}$$

Also, R_α^p is the 3×3 matrix of α radians rotated counter–clock–wise in the p plane:

$$R_\theta^{xy} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, R_\phi^{xz} = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}, R_\psi^{yz} = \begin{pmatrix} 0 & \cos\psi & -\sin\psi \\ 1 & 0 & 0 \\ 0 & \sin\psi & \cos\psi \end{pmatrix}. \tag{13}$$

Composing the 3×3 matrix V with $\vec{v}_i$ as its i–th column, we get:

$$R_\psi^{yz} R_\phi^{xz} R_\theta^{xy} V = I. \tag{14}$$

Having in mind that $(R_\alpha^p)^{-1} = R_{-\alpha}^p$, yields,

$$V = R_{-\theta}^{xy} R_{-\phi}^{xz} R_{-\psi}^{yz}. \tag{15}$$

While equation (10) computes the three angles θ, ϕ, and ψ out of the basis vectors (*polarization*), equation (15) reproduces the base from θ, ϕ, and ψ angles (*depolarization*). Using the notations of the compression theory, this method gives a compression ratio of $2:1$ in 3-D spaces (and $\frac{2n}{n-1}:1$ in n–D spaces).

B. Block–wise Interpolation

Consider a partition of the $N_W \times N_H$ as a set of rectangular regions $\{r_i | i = 1, \cdots, n\}$, with corresponding (given) values of $\{\lambda_i | i = 1, \cdots, n\}$, satisfying:

$$\forall i, \forall \vec{c} \in r_i, f(\vec{c}) \simeq \lambda_i, \tag{16}$$

for an arbitrary function $f : R^2 \rightarrow R$. The problem is to find $\tilde{f}$ as a good approximation of f. We address this problem as *block–wise interpolation* of the set $\{(r_i; \lambda_i) | i = 1, \cdots, n\}$. Note that in the case that the partition is a conventional rectangular grid, the problem reduces to an ordinary 2–D interpolation task. Here, we use a reformulated version of the well–known low–pass *Butterworth* filter, as the interpolation kernel,

$$B_{\tau,N}(x) = \left(1 + \left(\frac{x}{\tau}\right)^{2N}\right)^{-\frac{1}{2}}, N = rnd\left(\log_{\frac{a}{b}}\left(\frac{\beta\sqrt{1-\alpha^2}}{\alpha\sqrt{1-\beta^2}}\right)\right), \tau = a \sqrt[2N]{\frac{\alpha^2}{1-\alpha^2}}. \tag{17}$$

In this definition, the function $B_{\tau,N}(\cdot)$ satisfies the conditions of $B_{\tau,N}(a) = \alpha$ and $B_{\tau,N}(b) = \beta$. The 2–D version of this function is defined as:

$$B_{\tau,N,w,h}(x, y) = B_{w\tau,N}(x) B_{h\tau,N}(y). \tag{18}$$

Thus, w and h control the spread of the function in the x and y directions, respectively. Figure 2 shows the typical shape of the function $B^{w,h}_{\tau,N}(x, y)$ with $a = 0.9$, $\alpha = 0.7$, $b = 1$, $\beta = 0.5$, and $w = h = 16$.

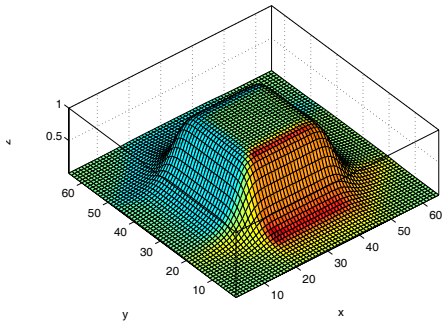


Fig. 2. Typical shape of the proposed interpolation kernel, $B^{w,h}_{\tau,N}(x, y)$.

Assuming that the region r_i is centered on (x_i, y_i), while its height and width are w_i and h_i, respectively, we propose the function $\tilde{f}$ to be defined as:

$$\tilde{f}(x, y) = \frac{\sum_{i=1}^{N} \lambda_i B_{\tau,N,\frac{w_i}{2},\frac{h_i}{2}}(x - x_i, y - y_i)}{\sum_{i=1}^{N} B_{\tau,N,\frac{w_i}{2},\frac{h_i}{2}}(x - x_i, y - y_i)}. \tag{19}$$

Now, $\tilde{f}(x, y)$, is a smooth version of the initial staircase function:

$$\forall [x, y]^T \in r_i : f_\circ(x, y) = \lambda_i. \tag{20}$$

Also, by setting proper values of the parameters a, b, α, and β, the function $\tilde{f}(x, y)$ will satisfy (16). The proper set of the parameters must force $B^{\frac{w_i}{2},\frac{h_i}{2}}_{\tau,N}(x - x_i, y - y_i)$ to become nearly one in entire r_i (except for the borders neighborhoods) and also to prevent r_i to intrude the interior with points of r_j, for $i \neq j$. Selecting a value near 1 (but smaller) for a and α, limits the decline of the ceil of the function, while setting $b = 1$ and a not too big value for β controls the effects of neighbor regions on each other. Note that setting $a = 1^-$, $\alpha = 1$, $b = 1^+$, and $\beta = 0$, is the marginal choice leading to no smoothing (the same as (20)).

As the generalization of the block–wise interpolation problem, assume the set of regions $\{(r_i; \lambda_{ij}) | i = 1, \cdots, n, j = 1, \cdots, m\}$, satisfying:

$$\lambda_{ij} = \arg_\lambda \left(\forall \vec{c} \in r_i, f_j(\vec{c}) \simeq \lambda\right), \tag{21}$$

for a set of arbitrary functions $f_i : R^2 \rightarrow R, i = 1, , \cdots, m$. In a similar manner with (19), we propose:

$$\tilde{f}_j(x,y) = \frac{\sum_{i=1}^{n} \lambda_{ij} B_{\tau,N,\frac{w_i}{2},\frac{h_i}{2}}(x - x_i, y - y_i)}{\sum_{i=1}^{n} B_{\tau,N,\frac{w_i}{2},\frac{h_i}{2}}(x - x_i, y - y_i)}. \tag{22}$$

Here, because the set of the base regions for all $\tilde{f}_j$ are the same, the total performance is increased by computing $B^{\frac{w_i}{2},\frac{h_i}{2}}_{\tau,N}(x - x_i, y - y_i)$ for each value of i, just once. Then, the problem reduces to m times computation of a weighted average.

When working in the polar coordinates, because of the 2π discontinuity, ordinary algebraic operations on the variables lead to spurious results (for example $\frac{o+2\pi}{2} = \pi$, while the average of 0 radians and 2π radians equals $0 \equiv 2\pi$ radians). To overcome this problem, we propose a new method: for the given problem, $\{(r_i; \theta_i) | i = 1, \cdots, n\}$, solve the problem $\{(r_i; \cos\theta_i, \sin\theta_i) | i = 1, \cdots, n\}$ and then find θ_i using ordinary trigonometric methods. Note that interpolating both $\sin\theta_i$ and $\cos\theta_i$ is performed to avoid ambiguity in the polar plane.

C. Eigenimage Extraction

Consider the PCA matrix, V_r, and the expectation vector, $\vec{\eta}_r$, corresponding to the *homogenous* cluster r. Then, for the color vector $\vec{c}$ belonging to r we get the PCA coordinates as,

$$\vec{c}^{\,T} = V_r^{-1}(\vec{c} - \vec{\eta}_r). \tag{23}$$

Assume that we can somehow find the color cluster $r_{\vec{c}}$ for each color vector $\vec{c}$, where $r_{\vec{c}}$ describes the *color vicinity* of $\vec{c}$, in the sense that,

$$\vec{c}^{\,T} = \begin{pmatrix} c'_1 \\ c'_2 \\ c'_3 \end{pmatrix} = V_{r_{\vec{c}}}^{-1}(\vec{c} - \vec{\eta}_{r_{\vec{c}}}), \tag{24}$$

satisfies:

$$\sigma_{c'_1} \gg \sigma_{c'_2} \gg \sigma_{c'_3}. \tag{25}$$

We call the c'_1, c'_2, and c'_3 images as the pc_1, pc_2, and pc_3, respectively. The original image can be perfectly reconstructed using these three channels, except for the numerical errors:

$$\vec{c} \simeq \vec{c}_3 = V_{r_{\vec{c}}}\vec{c}^{\,T} + \vec{\eta}_{r_{\vec{c}}}. \tag{26}$$

It is proved in [7] that for homogenous swatches, neglecting pc_3 (or even both pc_2 and pc_3), gives good approximations of the original image. Here, we generalize the approach. Note that the perfect reconstruction scheme of (26) does not rely on (24), while the partial reconstructions defined as:

$$\vec{c}_2 = V_{r_{\vec{c}}} \begin{pmatrix} c'_1 \\ c'_2 \\ 0 \end{pmatrix} + \vec{\eta}_{r_{\vec{c}}}, \tag{27}$$

and,

$$\vec{c}_1 = V_{r_{\vec{c}}} \begin{pmatrix} c'_1 \\ 0 \\ 0 \end{pmatrix} + \vec{\eta}_{r_{\vec{c}}}, \tag{28}$$

do rely on (24).

Although (28) gives a 1–D representation of a given color image, if the computation of $V_{r_{\vec{c}}}$ and $\vec{\eta}_{r_{\vec{c}}}$ gets expensive the scheme although being theoretically promising it actually is not applicable. Thus, we seek for a method for describing $V_{r_{\vec{c}}}$ and $\vec{\eta}_{r_{\vec{c}}}$ in a simple way. The case for defining $r_{\vec{c}} = N_{\vec{c}}$ (the neighborhood pixels) is automatically rejected (because to compute $V_{r_{\vec{c}}}$ and $\vec{\eta}_{r_{\vec{c}}}$ we need all the neighborhood points of $\vec{c}$ leading to a high redundancy and computation cost).

Here, we propose a fast and reliable method to compute the corresponding $V_{r_{\vec{c}}}$ and $\vec{\eta}_{r_{\vec{c}}}$ for all image pixels. Assume feeding the given image I to the bi–tree (or equivalently the quad–tree) decomposition method. The output of the decomposition method is the matrix Υ containing the coordinates of r_i along with the expectation matrix $\vec{\eta}_i$ and the polarized version of the PCA matrix $(\theta_i, \phi_i, \psi_i)$. Storing this portion of the Υ matrix needs $10n$ bytes. For ordinary values of $n \simeq 200$ in a 512×512 image, Υ will take about $\frac{1}{400}$ of the original image data. Now assume solving the problem $\{(r_i; \xi_i) | i = 1, \cdots, n\}$ using the block–wise interpolation, where ξ_i is the row vector containing η_{i_1}, η_{i_2}, η_{i_3}, θ_i, ϕ_i, and ψ_i. Note that the three values of θ_i, ϕ_i, and ψ_i are of angular type. Assume the solutions of the problem as the functions $\tilde{\eta}_1$, $\tilde{\eta}_2$, $\tilde{\eta}_3$, $\tilde{\theta}$, $\tilde{\phi}$, and $\tilde{\psi}$. Now we compute the functions $\tilde{\vec{\eta}} : R^2 \rightarrow R^3$ and $\tilde{V} : R^2 \rightarrow R^9$, as the value of the expectation vector and the PCA matrix in each pixel,

respectively. This leads to the computation of the three *eigenimages* pc_1, pc_2 and pc_3 are computed (using (24)). We call the function $\tilde{\vec{\eta}}: R^2 \rightarrow R^3$ as the *expectation map* (Emap) and the polarized version of $\tilde{V}: R^2 \rightarrow R^9$ as the *rotation map* (Rmap), respectively. As the PCA theory states [23], we expect the standard deviation of the three planes to be in descending order.

From linear algebra we know that for orthonormal matrices V_r the transformation of equation (24) satisfies:

$$\sigma_{pc_1}^2 + \sigma_{pc_2}^2 + \sigma_{pc_3}^2 = \sigma_r^2 + \sigma_g^2 + \sigma_b^2. \tag{29}$$

Thus,

$$\kappa_x = \frac{\sigma_x^2}{\sigma_r^2 + \sigma_g^2 + \sigma_b^2}, \tag{30}$$

shows the amount of information available in the x channel, where x is one of pc_1, pc_2, pc_3, r, g, and b. Note that,

$$\kappa_1 + \kappa_2 + \kappa_3 = \kappa_r + \kappa_g + \kappa_b = 1. \tag{31}$$

D. Color Image Compression

Consider the image I and its corresponding eigenimages pc_1, pc_2, and pc_3. Due to the energy compaction condition (see (29)), this scheme is actually an spectral image compression method. Reconstructing the image using just one or two eigenimage(s) gives the compression ratios of $\frac{3}{2}:1$ and $3:1$ (the theoretical margin) respectively. To add the spatial compression ability to the proposed method, we use the PU–PLVQ wavelet–based gray–scale image compression technique, that was proposed by *Kasaei et al.* in [24], for each eigenimage (with different compression ratios).

Given the highly nonstationary characteristic of images, the wavelet transform is used in the gray-scale image compression stage. Some advantages of using the wavelet transform include providing: multiscale analysis, flexible time–frequency tiling, good ability to extract signal information from a nonstationary signal, good energy compaction, matching human vision, fast algorithms ($O(N)$), orthogonal wavelets (to preserve length in transform domain), short filters (to prevent the Gibbs ringing effects to propagate far from edges), absence of block artifacts, frequently more sparse representations than the discrete cosine transform (higher lossy compression gains at low bit rates). Furthermore, as the wavelet packets give the possibility of applying further decompositions at any node of the decomposition tree that results in a structure that matches the characteristics of input data, and thus increases the compression performance, here the wavelet packets are utilized.

As Figure 3–(a) shows, the transmitted information contains the compressed versions of the pc_1, pc_2, and pc_3, along with the Υ, α, a, β, and b (for block–wise interpolation). Assuming that the image to be compressed is a $H \times W$ color image, decomposed into n blocks, the total amount of information to be sent equals: $10n$ bytes for storing x_{i1}, x_{i2}, y_{i1}, y_{i2}, η_{i1}, η_{i2}, η_{i3}, θ_i, ϕ_i, and ψ_i plus $WH(\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1})$ for storing pc_1, pc_2 and pc_3 eigenimages compressed with compression ratios of λ_1, λ_2, and λ_3, respectively (where $\lambda_1 > \lambda_2 > \lambda_3$). Thus, the total compression ratio equals,

$$\lambda \simeq \frac{3}{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \frac{10n}{WH}} \tag{32}$$

A nominal value of $\lambda_2 = \lambda_1$ and $\lambda_3 = \infty$ leads to $\lambda \simeq 1.5\lambda_1$. Note that using a pure spatial compression, all three channels must be compressed with almost the same compression ratios, resulting in a total compression ratio of,

$$\tilde{\lambda} = \frac{3WH}{\frac{WH}{\lambda_1} + \frac{WH}{\lambda_1} + \frac{WH}{\lambda_1}} = \lambda_1. \tag{33}$$

As shown in Figure 3–(b), in the decompression process the Emap and the Rmap are computed just like what performed in the encoding process. Using these information along with the decoded versions of pc_1, pc_2, and pc_3, the original image is reconstructed.

IV. Experimental Results

The proposed algorithms are developed in *MATLAB 6.5*, on an *1100 MHz Pentium III* personal computer with $256MB$ of *RAM*. A database of color images (140 samples) including the standard images of *Lena*, *Mandrill*, *Airplane*, *Peppers*, *Girl*, and *Couple* and some professional color photographs [25] are used. All images have the size of 512×512, in RGB color space, and compressed using standard jpeg compression with compression ratio of about $3:1$.

A. Block–wise Interpolation

Figure 4–(a) shows a sample problem set given to the proposed block–wise Interpolation. Figure 4–(b) illustrates the corresponding $\tilde{f}$. In this sample the resulting *signal to noise ratio* (SNR) is more than $22dB$.

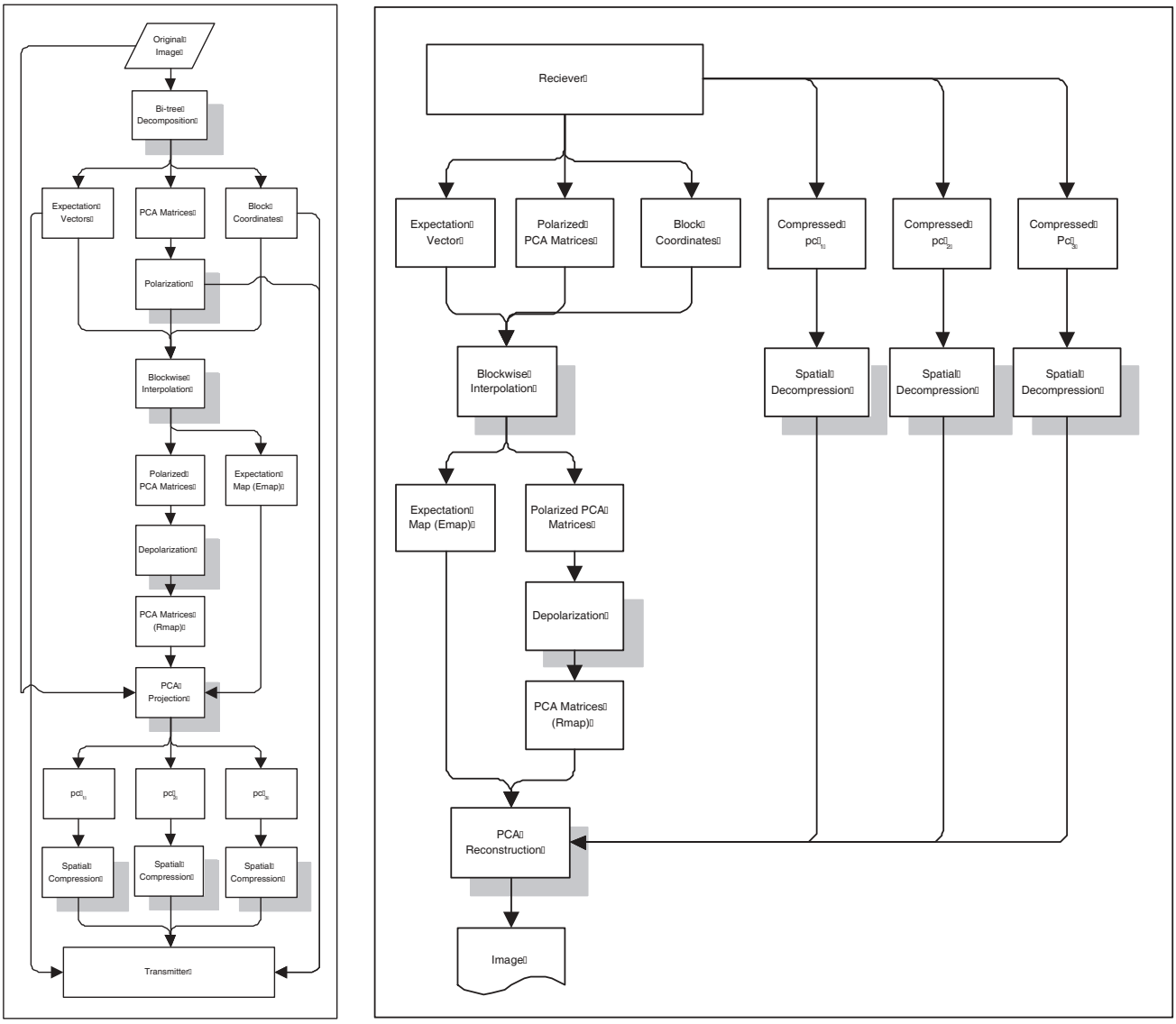


Fig. 3. Flowchart of the proposed methods. (a) Compression. (b) Decompression.

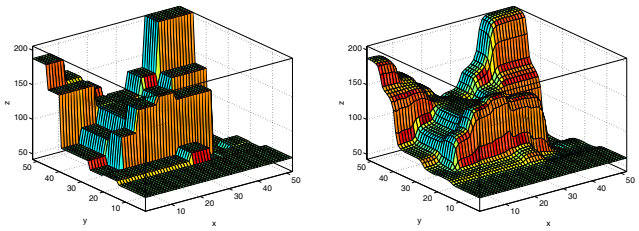
B. Eigenimage

Consider the image shown in Figure 5–(a), which is decomposed with parameters of $p = 0.5$, $\varepsilon_1 = 5$, and $\varrho = 5$ into 91 blocks (see Figure 5–(b)). Figures 5–(c) and 5–(d) show the corresponding EMap and RMap. Figure 6 shows the three pc_i channels corresponding to the image shown in figures 5–(a).In all eigenimages the dynamic range of the image is exaggerated to give a better visualization.

The stochastic distribution of pc_i is investigated in Figure 6–(a) which shows the histogram of the three produced planes for the case of the image shown in Figure 5–(a). In this example the standard deviations of the pc planes is computed as: $\sigma_{pc_1} = 52$, $\sigma_{pc_2} = 12$, and $\sigma_{pc_3} = 6$. Note the perfect compaction of the energy in pc_1.

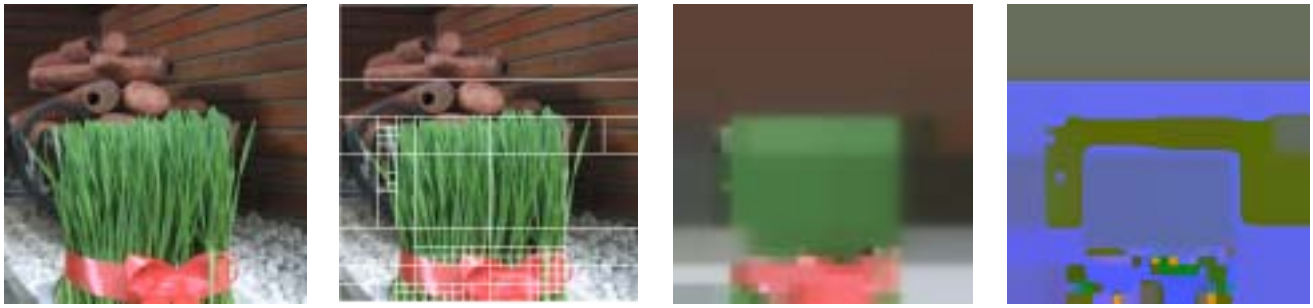
Figure 7–(a), 7–(b), and 7–(c) show the values of κ_1, κ_2, and κ_3 for the image in Figure 5–(a) for different values of ε_1 and ϱ. Note that rather than the trivial cases of $\varrho \leq 2$ and $\varepsilon_1 > 9$ (which are never used actually), more than 90% of the image energy is stored in pc_1, while pc_2 and pc_3 hold about 9% and 1% of the energy, respectively. Having in mind that in the original image $\kappa_r = 38\%$, $\kappa_g = 32\%$, and $\kappa_b = 30\%$, the energy compaction of the eigenimages are considerable.

Figure 8 shows the results of reconstructing the image of Figure 8–(a) from the eigenimages. While Figure 8–(b) shows the result of reconstructing the image using all three eigenimages, Figures 8–(c) and 8–(d) show the results of ignoring pc_3 and both pc_3 and pc_2, respectively. The resulting $PSNR$ values are $60dB$, $38dB$, $31dB$. Note that $PSNR = 60dB$ (instead of infinity), for reconstructing the image using all eigenimages is caused by the numerical errors, while the two other $PSNR$ values ($38dB$,



(a) (b)

Fig. 4. Block–wise interpolation, (a) sample problem and (b) the solution with $SNR > 22dB$.



(a) (b) (c) (d)

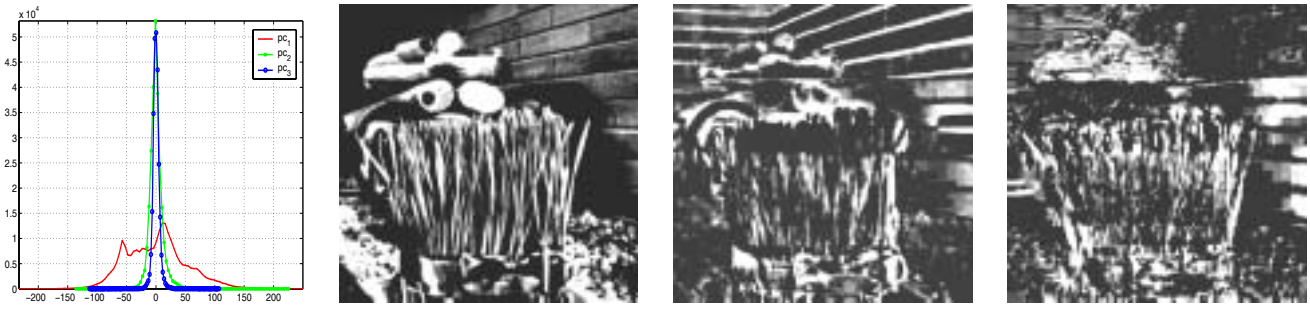
Fig. 5. (a) Original image adopted from [25], (b) result of the bi–tree decomposition method, (c) Emap, and (d) Rmap.

$31dB$) show some loss of information. As indicated recently $PSNR$ values of above $38dB$ are visually satisfactory even for professionals [26].

Figure 9 shows the $PSNR$ values obtained by reconstructing the image using all the three channels (Figure 9–(a)), only two channels (Figure 9–(b)), and just one channel (Figure 9–(c)), for different values of ε_1 and ϱ. Note that for values of $\varepsilon_1 \leq 8$ and $\varrho \geq 3$, reconstructing the image using all eigenimages gives the high $PSNR$ value of about $60dB$, while neglecting one and two eigenimages results in $PSNR \geq 35dB$ and $PSNR \geq 28dB$, respectively.

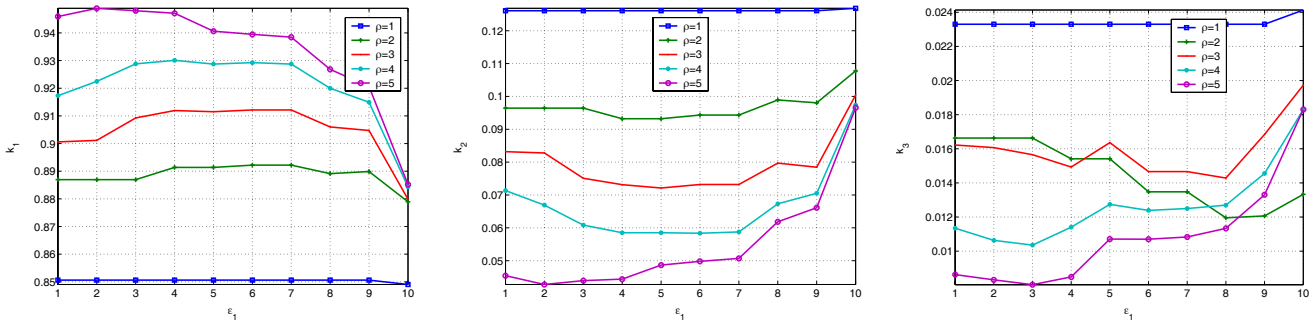
C. Color Image Compression

Figure 11 shows the results of the proposed compression method applied on the images shown in Figure 10. Table I shows the compression ratio used for compressing the eigenimages and the resulting compression ratio and $PSNR$ values. These results has been acquired while setting $p = \frac{1}{2}$, $\varepsilon_1 = 5$, and $\varrho = 5$. Figure 12 shows the exaggerated difference between the reconstructed images shown in Figure 11 and the original images shown in Figure 10. Here, the exaggeration scheme is defined as $x^\star = (x - [\eta - \sigma])/(2\sigma)$, where η and σ denote the expectation and the standard deviation of x, respectively. Note the high compression ratio of about $70:1$ in all cases, while the $PSNR$ is mostly above $25dB$. Among other region–based coding approaches the method by *Carveic et. al.* is one of the best [27]. They mixed the color and texture information into a single vector and performed the coding using a massively computationally expensive algorithm. The final results show $PSNR$



(a) (b) (c) (d)

Fig. 6. Exaggerated eigenimages of the image shown in Figure 5–(a) and their histograms. (a) Histograms. (b) pc_1, (c) pc_2, (d) pc_3.



(a) (b) (c)

Fig. 7. Distribution of the energy between the three eigenimages for different values of ε_1 and ϱ. (a)κ_{pc_1}. (b)κ_{pc_2}. (c)κ_{pc_3}.

(a) (b) (c) (d)

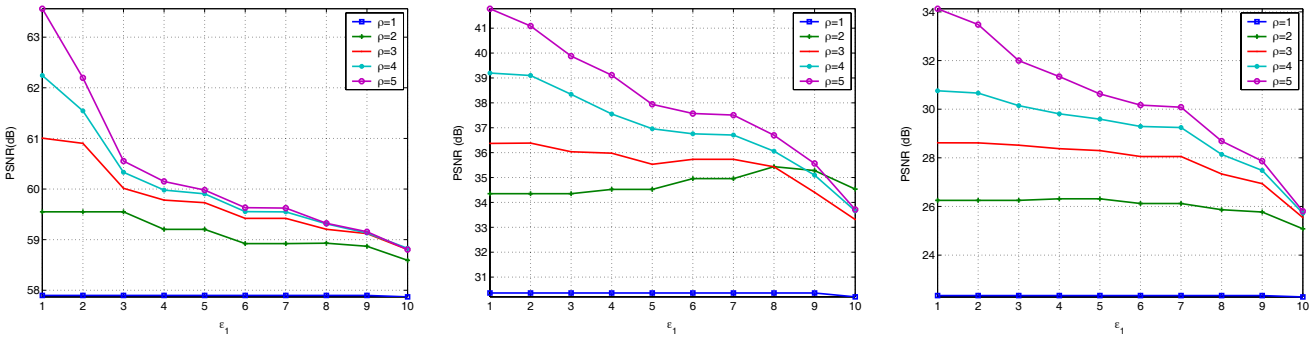
Fig. 8. Results of reconstructing an image from its eigenimages. (a) Original image adopted from [25]. (b) Using all eigenimages ($PSNR = 60dB$). (c) ignoring one eigenimage ($PSNR = 38dB$). (c) ignoring two eigenimages ($PSNR = 31dB$).

values of about $20:1$ for compression ratios of about $40dB$. In [28] the researchers use the same separation scheme between compression in the two disjoint domains of spectral and spatial redundancy using a PCA neural network. They reached the compression ratio of $3.7:1$ with value of $PSNR$ around $25dB$, while almost all test samples are homogenous. In [29] the method gives the compression ratio of about $14.5:1$ but with the same range of $PSNR$ as ours.

The only drawback of the proposed compression method is some rectangular artifacts in ultra simple images, as seen in Figure 11–(f). Our future plan is to overcome this minority of problem.

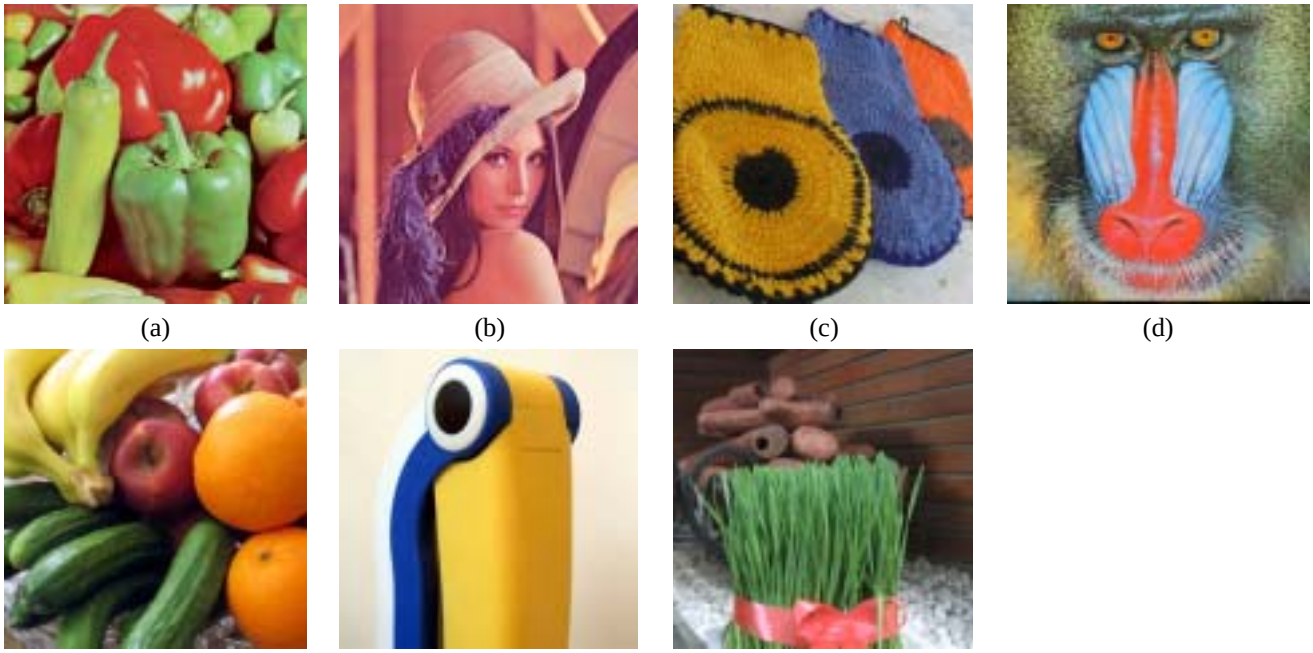
V. Conclusions

In this paper, using the proposed PCA–based eigenimage extraction method, incorporated with a wavelet–based gray–scale image compression algorithm, a new color image compression method was proposed. The proposed eigenimage extraction method was proved to be highly efficient in energy compaction and partial reconstruction of the image, both quantitatively



(a) (b) (c)

Fig. 9. $PSNR$ values of image reconstruction using. (a)All three eigenimages. (b)Only two eigenimages. (c)Just one eigenimage for different values of ε_1 and ϱ.



(a) (b) (c) (d)

(e) (f) (g)

Fig. 10. Sample images.

TABLE I

NUMERICAL RESULTS AND PARAMETERS USED FOR THE RESULTS SHOWN IN FIGURE 11. λ: COMPRESSION RATIO, n: NUMBER OF BLOCKS.

Sample	n	pc_1		pc_2		pc_3		**Image**		
		λ	**Bitrate**	λ	**Bitrate**	λ	**Bitrate**	λ	**Bitrate**	$PSNR$
Figure 10–(a)	334	$1:42.6$	$0.19bpp$	$1:58.8$	$0.14bpp$	$1:\infty$	$0bpp$	$1:71.1$	$0.34bpp$	$28.4dB$
Figure 10–(b)	60	$1:43.1$	$0.19bpp$	$1:56.2$	$0.14bpp$	$1:\infty$	$0bpp$	$1:72.6$	$0.33bpp$	$33.6dB$
Figure 10–(c)	161	$1:39.8$	$0.20bpp$	$1:55.1$	$0.15bpp$	$1:\infty$	$0bpp$	$1:68.0$	$0.35bpp$	$30.4dB$
Figure 10–(d)	311	$1:44.4$	$0.18bpp$	$1:58.0$	$0.14bpp$	$1:\infty$	$0bpp$	$1:72.5$	$0.33bpp$	$24.5dB$
Figure 10–(e)	203	$1:41.3$	$0.19bpp$	$1:56.9$	$0.14bpp$	$1:\infty$	$0bpp$	$1:70.1$	$0.34bpp$	$33.3dB$
Figure 10–(f)	32	$1:45.5$	$0.18bpp$	$1:62.8$	$0.13bpp$	$1:\infty$	$0bpp$	$1:78.8$	$0.30bpp$	$26.1dB$
Figure 10–(g)	91	$1:41.3$	$0.19bpp$	$1:57.7$	$0.14bpp$	$1:\infty$	$0bpp$	$1:71.4$	$0.34bpp$	$35.4dB$

and subjectively. The comprehensive performance analysis showed the efficiency of the proposed color image compression algorithm.

ACKNOWLEDGEMENT

The first author wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas. We also thank the generous photographers Ms. *Shohreh Tabatabaii Seifi* and *Ali Qanavati* for their incorporation of their rich digital image archive.

REFERENCES

[1] N. Papamarkos, C. Strouthopoulos, and I. Andreadis, "Multithresholding of color and gray–level images through a neural network technique," *Image and Vision Computing*, vol. 18, pp. 213–222, 2000.
[2] D. Tschumperle, "Pde's based regularization of multivalued images and applications," Ph.D. dissertation, University of Nice, Sophia Antipolis, 2002.
[3] H. Cheng and J. Li, "Fuzzy homogeneity and scale–space approach to color image segmentation," *Pattern Recognition*, vol. 36, pp. 1545–1562, 2003.
[4] J. Bruce, T. Balch, and M. Veloso, "Fast and cheap color image segmentation for interactive robots," in *Proceedings of IROS–2000*, Japan, 2000.
[5] T. Chaira and A. Ray, "Fuzzy approach for color region extraction," *Pattern Recognition*, vol. 24, pp. 1943–1950, 2003.
[6] L. Lucchese and S. Mitra, "Colour segmentation based on separate anisotropic diffusion of chromatic and achromatic channels," *Vision, Image, and Signal Processing*, vol. 148(3), pp. 141–150, 2001.
[7] A. Abadpour and S. Kasaei, "A new parametric linear adaptive color space and its pca–based implementation," in *The 9th Annual CSI Computer Conference, CSICC*, Tehran, Iran, 2004, pp. 125–132.
[8] ——, "Performance analysis of three homogeneity criteria for color image processing," in *IPM Workshop on Computer Vision*, Tehran, Iran, 2004.
[9] S.-C. Cheng and S.-C. Hsia, "Fast algorithm's for color image processing by principal component analysis," *Journal of Visual Communication and Image Representation*, vol. 14, pp. 184–203, 2003.
[10] *ITU–R Recommendation BT–601–5: Studio Encoding Parameters of Digital Television for Standard 4:3 and Widescreen 16:9 Aspect Ratios*. Geneva: http://www.itu.ch/, 1994.

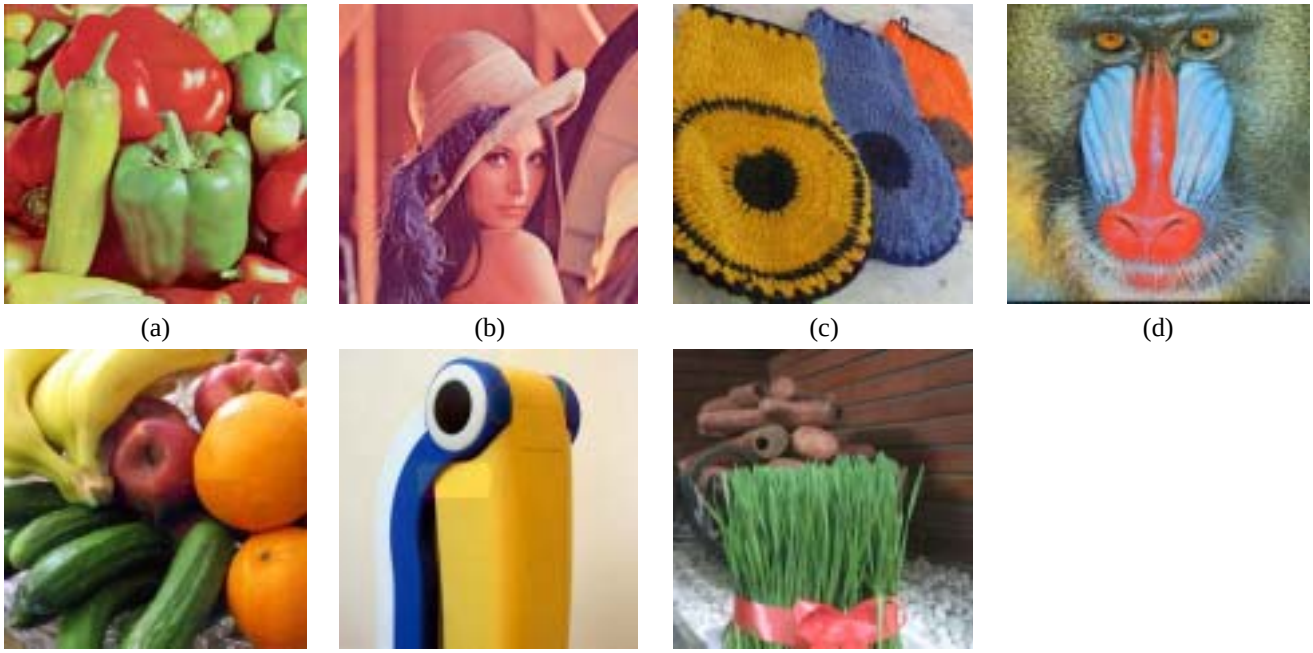


Fig. 11. Results of applying the proposed compression method on the images shown in Figure 10. For details see Table I.

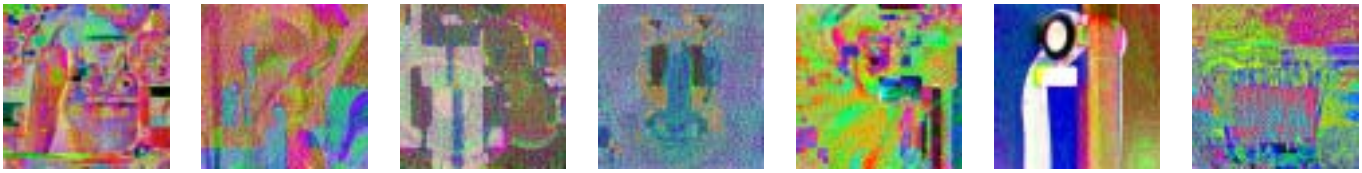


Fig. 12. Exaggerated error of the proposed compression method for the images shown in Figure 10.

[11] K. Benson, *Television Engineering Handbook*, J. C. Whitaker, Ed. New York, London: Mc Graw–Hill, 1992.
[12] J. Slater, *Modern Television System to HDTV and Beyond*. London: Pitman, 1991.
[13] J. O. Limb, C. Rubinstein, and J. Thompson, "Digital coding of color video signals– a review," *IEEE Transaction of Communication*, vol. 25(11), pp. 1349–1385, 1977.
[14] W. Pratt, "Spatial transform coding of color images," *IEEE Transaction on Communication Technology*, vol. 19(6), pp. 980–992, 1971.
[15] S. Fukuma, M. Iwahashi, and N. Kambayashi, "Lossless color coordinate transform for lossless color image coding," 1998, pp. 595–598.
[16] C. Clausen and H. Wechsler, "Color image compression using pca and backpropagation learning," *Pattern Recognition*, vol. 33, pp. 1555–1560, 2000.
[17] D. Carevic and T. Caelli, "Region–based coding of color images using karhunen–loeve transform," *Graphical Models and Image Processing*, vol. 59(1), pp. 27–38, 1997.
[18] H. Samet, "Region representation: Quadtrees from boundary codes," *Comm. ACM*, vol. 21, p. 163:170, 1980.
[19] C. Faloutsos, H. Jagadish, and Y. Manolopoulos, "Analysis of the n–dimensional quadtree decomposition for arbitrary hyperrectangles," *IEEE Transaction on Knowledge and Data Engineering*, vol. 9, No. 3, pp. 373–383, 1997.
[20] M. Yazdi and A. Zaccarin, "Interframe coding using deformable triangles of variable size," in *ICIP'97*, 1997, pp. 456–459.
[21] A. Abadpour and S. Kasaei, "A new tree decomposition method for color images," in *10th Annual CSI Computer Conference (CSICC2005)*, Tehran, Iran, 2005.
[22] ——, "A new pca–based robust color image watermarking method," in *the 2nd IEEE Conference on Advancing Technology in the GCC: Challenges, and Solutions*, Manama, Bahrain, 2004.
[23] H. Hotteling, "Analysis of a complex of statistical variables into principal components." *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.
[24] S. Kasaei, M. Dericher, and B. Biashash, "A novel fingerprint image compression technique using wavelet packets and pyramid lattice vector quantization," *IEEE Transactions on Image Processing*, vol. 11(12), pp. 1365–1378, 2002.
[25] S. T. Seifi and A. Qanavati, "Digital color image archive," Qnavati@mehr.sharif.edu.
[26] S. Katzenbeisser and A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House Inc., 2000.
[27] D. Carevic and T. Caelli, "Region–based coding of color images using karhunen–loeve transform," *Graphical Models and Image Processing*, vol. 59(1), pp. 27–38, 1997.
[28] C. Clausen and H. Wechsler, "Color image compression using pca and backpropagation learning," *Pattern Recognition*, vol. 33, pp. 1555–1560, 2000.
[29] Y.-D. Yu, D.-S. Kang, and D. Kim, "Color image compression based on vector quantization using pca and lebd," in *IEEE IENCON'99*, 1999, pp. 1259–1262.