Color PCA Eigenimages and their Application to Compression and Watermarking

Arash Abadpour [a], Shohreh Kasaei [b,*]

[a] *Mathematics Science Department, Sharif University of Technology, Tehran, Iran*

[b] *Computer Engineering Department, Sharif University of Technology, Tehran, Iran*

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. However, it has been proved that using vector–based tools leads to a more appropriate understanding of color images and thus more efficient algorithms for processing them. Such tools are able to take into consideration the high correlation of the color components and thus to successfully carry out energy compaction. In this paper, a novel method is proposed to utilize the principal component analysis in the neighborhoods of an image in order to extract the corresponding eigenimages. These eigenimages exhibit high levels of energy compaction and thus are appropriate for operations such as compression and watermarking. Subsequently, two such methods are proposed in this paper and their performance comparison with respect to available approaches is presented.

1 Introduction

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

* Corresponding author: Computer Engineering Department, Sharif University of Technology, Azadi Ave., Tehran, Iran, P.O. Box 11155–9517, Telephone: (+98) 21 6616 4631.

Email addresses: abadpour@math.sharif.edu (Arash Abadpour), skasaei@sharif.edu (Shohreh Kasaei).

URLs: math.sharif.edu/∼abadpour (Arash Abadpour), sharif.edu/∼skasaei (Shohreh Kasaei).

characteristics of color images, as vector geometries [2], some researchers have considered them as parallel gray–scale components [3–5]. However, it has been proved that the application of *principal component analysis* (PCA) leads to appropriate descriptors for natural color images [6,7]. These descriptors work in vector domains and take into consideration the statistical dependence of the components of color images taken from the nature [8].

In this paper, we use a PCA–based distance function and the related homogeneity criterion. This criterion will be used in a tree decomposition method which splits a given image into homogeneous patches. The patches will be further analyzed using the PCA to provide energy compaction through a proposed eigenimage extraction method. These eigenimages are then used in order to provide color image processing tools such as compression and watermarking.

The rest of this paper is organized as follows. First, Section 2 briefly reviews the related literature. Then, Section 3 discusses the tools used in this paper and follows with introducing the proposed method. The paper continues with Section 4 which contains the experimental results and discussions. Finally, Section 5 concludes the paper.

2 Literature Review

This section briefly reviews the related literature. First, in Section 2.1, a few distance functions and their application to color image processing are discussed. This section also looks at PCA–based color image processing and its implications. Then, Sections 2.2, 2.3, and 2.4 look at the key issues in color image decomposition, compression, and watermarking.

2.1 Distance Functions and PCA–Based Color Image Processing

The Euclidean distance is generally assumed to be a proper distance function in color applications [1,9–14]. Accordingly, it has been applied in different color spaces (*e.g.* CIE–Lab [15]). However, research has shown that, although being very convenient to use, this measure is not a proper color descriptor [16].

A theoretical analysis of highlights in color images was presented in a classical paper published by *Klinker*, *Shafer*, and *Kanade* in 1988 [6]. In that work, the light reflected from an arbitrary point of a dielectric object was modeled. In 1990, the same group applied their approach to color image understanding [16]. These developments were left mostly untouched by the research community, till more than a decade later, in 2003, without paying too much attention to

the theoretical aspects, *Cheng* and *Hsia* used the PCA for color image processing [7]. Then, in 2004, *Nikolaev* and *Nikolayev* started the work back from the theory and proved that PCA is a proper tool for color image processing [17]. The next necessary step was introduced in 1991, when *Turk* and *Pentland* proposed their eigenface method [18], in a completely different context. There, they developed a novel idea which connected the eigenproblems in the color domain and the spatial domain, together. Although, there is this rich theoretical background for the linear local models of color, it is quite common to see research procedures which are based on the old color space paradigm, even published in 2007 (see [19] for example). For a more comprehensive discussion of this topic refer to [20].

Aside from the theoretical evidence for the superiority of PCA–based color descriptors, an extensive empirical analysis, given in [21], shows that a certain PCA–based color distance produces more appropriate results, compared to the conventional Euclidean and Mahalanobis distances. The latter one is included in the analysis because some researchers tend to use weighted Euclidean distance, which when tuned well converges to the well–known Mahalanobis distance (*e.g.* [22]). Here, we will briefly introduce the utilized PCA–based distance function. The interested reader is referred to [21] for details about the two others as well as a comprehensive comparison between the three. The chosen distance function will be used in a homogeneity criterion for the utilized tree decomposition, which is discussed next.

2.2 Tree Decomposition

Quad–tree decomposition is a well–known method for splitting an image into homogeneous sub–blocks, resulting in a very coarse, but fast, segmentation [23]. Defining the whole image as a single block, the method is performed according to a problem–specific homogeneity criterion. Generalizations of quad–tree include working on dimension [24] and shape [25] of the blocks. In conventional quad–tree decomposition, a non–homogeneous block is split into four sub–blocks. The one–split–to–four rule has been optimized by using a novel one–split–to–two rule, resulting in a method called bi–tree decomposition [26].

2.3 Color Image Compression

The early approach to color image compression is based on decorrelating the color components using a linear or nonlinear invertible coordinate transformation (*e.g.*, YC_bC_r [27], YIQ [28] and YUV [29]). Then, one of the standard gray–scale compression methods (such as *differential pulse code modulation*

(DPCM) [30] or transform coding [31]) will be applied on each component, separately (see also [32]).

This approach is inefficient, because none of the available color spaces is able to completely decorrelate the color components in a real image [6]. In [33], using the PCA approach in the neighboring pixels, the author discusses the idea of separating the spatial and the 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 introduced in [33] is neglecting the fact that in non–homogeneous patches, the PCA does not perform energy compaction [9]. In [9], the author combines the spatial and spectral information to reach a higher compression ratio. Although, that method is based on expensive computation, the *peak signal to noise ratio* (PSNR) results are desperate.

2.4 Color Image Watermarking

Ease procedure of copy and transfer of digital files has made the urge for effective watermarking schemes to protect the fidelity of the media [34]. On top of the general–purpose data watermarking approaches (*e.g.* [35,36]), there is an extensive literature for watermarking images. The main concern in this field is to locate spots in an image, in different domains, where data can be added without being perceptible by human observers. The early approaches taken in this regard use the redundancy of the natural images in the spatial domain [34,37]. With the introduction of color imaging, some of these grayscale watermarking techniques have been exported to color images. For example, in [36], the authors embed the watermark in the blue component. In another work [38], the $CIE-Lab$ coordinates are quantized and then the watermark is copied into the least important bits. As mentioned by the authors in [39], most available color image watermarking approaches work on the luminance or independent color components. This ignores the correlation between color components, and the regarding opportunity for effective embedding of information in color images.

3 Proposed Method

3.1 Distance Functions and Homogeneity Criteria

In [40], the authors proposed to use the error made by neglecting the two least important principal components (the second and the third) as a distance

function, called the *linear partial reconstruction error* (LPRE)

$$\tau_{\mathbf{r}}\left(\vec{\mathbf{c}}\right) = \left\| \vec{\mathbf{v}}_{\mathbf{r}}^{T}\left(\vec{\mathbf{c}} - \vec{\eta}_{\mathbf{r}}\right)\vec{\mathbf{v}}_{\mathbf{r}} - \left(\vec{\mathbf{c}} - \vec{\eta}_{\mathbf{r}}\right)\right\|. \tag{1}$$

Here, $\vec{\mathbf{v}}_{\mathbf{r}}$ denotes the direction of the first principal component of $\mathbf{C}_{\mathbf{r}}$, the covariance matrix of the color vectors of the patch $\mathbf{r}$

$$\mathbf{C}_{\mathbf{r}} = E_{\vec{\mathbf{c}}\in\mathbf{r}}\left\{\left(\vec{\mathbf{x}}_i - \vec{\eta}\right)\left(\vec{\mathbf{x}}_i - \vec{\eta}\right)^{T}\right\}. \tag{2}$$

Also, $\|\cdot\|$ is the Euclidean norm and $\vec{\mathbf{x}}^T$ denotes transposition. Furthermore, $\vec{\eta}_{\mathbf{r}}$ is the expectation vector of $\mathbf{r}$, $\vec{\eta}_{\mathbf{r}} = E_{\vec{\mathbf{c}}\in\mathbf{r}}\left\{\vec{\mathbf{c}}\right\}$. With these definitions, $\tau_{\mathbf{r}}\left(\vec{\mathbf{c}}\right)$ is the distance from the arbitrary color vector $\vec{\mathbf{c}}$ to the given set $\mathbf{r}$. Comparison of this measure with other conventional ones can be found in [21].

Having defined a proper distance function, a related homogeneity criterion can be derived by defining

$$\tau_{\mathbf{r}} = E_{\vec{\mathbf{c}}\in\mathbf{r}}\left\{\tau_{\mathbf{r}}\left(\vec{\mathbf{c}}\right)\right\}. \tag{3}$$

This way, the LPRE–based homogeneity criterion is defined. As mentioned in Section 2.1, the empirical analysis carried out in [21] shows that, compared to the Euclidean and Mahalanobis distances, LPRE performs the three below mentioned primitive tasks more efficiently. First, it yields a more appropriate classification of pixels into relevant and irrelevant ones, in regards to a given homogeneous patch. Furthermore, when asked to decide whether or not a patch is homogeneous, LPRE's results comply better with the human perception. Finally, in the existence of outliers, LPRE shows more robustness than both the Euclidean and the Mahalanobis distances [21] (also see [41]).

3.2 Bi–Tree Decomposition

As discussed in Section 2.2, having chosen a proper homogeneity criterion, an image can be decomposed into homogeneous regions (here, based on $\tau_{\mathbf{r}}$). Starting with the entire image, the tree is produced by splitting non–homogeneous regions (the ones for which $\tau_{\mathbf{r}} > \varepsilon_1$, where $\varepsilon_1 \in [1, 10]$ is a threshold). In a $W \times H$ image, the depth of a $w \times h$ block $\mathbf{r}$ is defined as

$$\varrho_{\mathbf{r}} = \max\left\{\log_2\frac{W}{w}, \log_2\frac{H}{h}\right\} \tag{4}$$

and no block is permitted to reach the depth more than a preselected marginal tree depth $\varrho \in [1, 5]$.

Here, we use a novel tree decomposition method called bi–tree [26]. In bi–tree, if the block $\mathbf{r}$ is not homogeneous enough, rather than the deterministic choice of the sub–blocks in quad–tree, $\mathbf{r}$ is divided either horizontally or vertically (see Figure 1). This is accomplished in the way that the total non–homogeneity of the output gets the least possible. Extensive empirical comparison of bi–tree with quad–tree, given in [26], shows that bi–tree produces fewer blocks, with a more uniform distribution of block sizes. On the contrary, quad–tree splits the image more and produces a more scattered pattern of block sizes. The total homogeneity of the resulting blocks are almost the same in both methods. The achievements of bi–tree have the cost of demanding about four times more computational resources than the quad–tree. As the proposed method works better when blocks have similar sizes, we use bi–tree in this paper.

3.3 Basis Vector Polarization

There is a manipulated form of the well–known Euler angles that relates any right–rotating orthonormal matrix with three angles, in a one–to–one revertible transformation [42]. As such, for the matrix $\mathbf{V}$, with its i–th column denoted as $\vec{\mathbf{v}}_i$, we write $\mathbf{V} \sim (\theta, \phi, \psi)$ when

$$\begin{cases} \theta = \angle\left(\vec{\mathbf{v}}_1^{xy}, [1,0]^T\right) \\ \phi = \angle\left(\left(\mathbf{R}_\theta^{xy}\vec{\mathbf{v}}_1\right)^{xz}, [1,0]^T\right) \\ \psi = \angle\left(\left(\mathbf{R}_\phi^{xz}\mathbf{R}_\theta^{xy}\vec{\mathbf{v}}_2\right)^{yz}, [1,0]^T\right) \end{cases} \tag{5}$$

Here, $\angle(\vec{\mathbf{v}}, \vec{\mathbf{u}})$ is the angle between the two vectors $\vec{\mathbf{v}}, \vec{\mathbf{u}} \in \mathbf{R}^2$ and $\vec{\mathbf{v}}_p$ denotes the project of the vector $\mathbf{v}$ on the plane p. Also, $\mathbf{R}_\alpha^p$ is the 3×3 matrix of α radians counter–clock–wise rotation in the p plane

$$\begin{cases} \mathbf{R}_\theta^{xy} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_\phi^{xz} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \\ \mathbf{R}_\psi^{yz} = \begin{bmatrix} 0 & \cos\psi & -\sin\psi \\ 1 & 0 & 0 \\ 0 & \sin\psi & \cos\psi \end{bmatrix}. \end{cases} \tag{6}$$

It can be proved that to produce $\mathbf{V}$ out of the triple (θ, ϕ, ψ) one may use [43]

$$\mathbf{V} = \mathbf{R}^{xy}_{-\theta}\mathbf{R}^{xz}_{-\phi}\mathbf{R}^{yz}_{-\psi}. \tag{7}$$

Note that $(\mathbf{R}^p_\alpha)^{-1} = \mathbf{R}^p_{-\alpha}$. While equation (5) computes the three angles θ, ϕ and ψ out of $\mathbf{V}$, equation (7) reconstructs $\mathbf{V}$ from θ, ϕ and ψ. These methods are called polarization and depolarization of a right–rotating orthonormal matrix, respectively [42].

3.4 Block–Wise Interpolation

Consider a partition of the $H \times W$ square as a set of rectangular regions $\{\mathbf{r}_i \,|i = 1, \cdots, n\}$, while the values of $\{\lambda_i | i = 1, \cdots, n\}$ are given, satisfying

$$f(\vec{\mathbf{c}}) \simeq \lambda_i, \forall i, \forall \vec{\mathbf{c}} \in \mathbf{r}_i \tag{8}$$

for an unknown function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. The problem is to properly approximate f as $\tilde{f}$. We address this problem as block–wise interpolation of the set $\{(\mathbf{r}_i; \lambda_i) \,|i = 1, \cdots, n\}$.

Note that, for a conventional rectangular grid as the partition, the problem reduces to an ordinary 2–D interpolation task. Here, we use a reformulated version of the well–known lowpass Butterworth filter, as the interpolation kernel, to carry out the interpolation in the more general case.

$$\begin{cases} B_{\tau,N}(x) = \left(1 + \left(\frac{x}{\tau}\right)^{2N}\right)^{-\frac{1}{2}} \\ \begin{cases} N = rnd\left(\log_{\frac{a}{b}}\left(\dfrac{\beta\sqrt{1-\alpha^2}}{\alpha\sqrt{1-\beta^2}}\right)\right) \\ \tau = a\sqrt[2N]{\dfrac{\alpha^2}{1-\alpha^2}} \end{cases} \end{cases} \tag{9}$$

Here, a, b, α, and β are given and the function $B_{\tau,N}(\cdot)$ satisfies $B_{\tau,N}(a) = \alpha$ and $B_{\tau,N}(b) = \beta$. The 2–D version of this function can be defined as, $B^{w,h}_{\tau,N}(x, y) = B_{w\tau,N}(x)B_{h\tau,N}(y)$, where 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 $w = h = 16$.

Assuming that the region $\mathbf{r}_i$ is centered at (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

$$\tilde{f}(x,y) = \frac{\sum_{i=1}^{N} \lambda_i B_{\tau,N}^{\frac{w_i}{2},\frac{h_i}{2}}(x-x_i, y-y_i)}{\sum_{i=1}^{N} B_{\tau,N}^{\frac{w_i}{2},\frac{h_i}{2}}(x-x_i, y-y_i)}. \tag{10}$$

Note that $\tilde{f}$ is a smooth version of the staircase function f.

As the generalization of the block–wise interpolation problem, assume that the set of regions and the corresponding λ_{ij}s are given as $\{(\mathbf{r}_i; \lambda_{i1}, \cdots, \lambda_{im}) \,|i = 1, \cdots, n\}$, satisfying $f_j(\vec{\mathbf{c}}) \simeq \lambda_{ij}, \forall i, j, \vec{\mathbf{c}} \in \mathbf{r}_i$, for a set of m unknown functions $f_j : \mathbb{R}^2 \to \mathbb{R}$. To solve this problem, in a way similar to (10), we calculate

$$\tilde{f}_j(x,y) = \frac{\sum_{i=1}^{n} \lambda_{ij} B_{\tau,N}^{\frac{w_i}{2},\frac{h_i}{2}}(x-x_i, y-y_i)}{\sum_{i=1}^{n} B_{\tau,N}^{\frac{w_i}{2},\frac{h_i}{2}}(x-x_i, y-y_i)}. \tag{11}$$

Here, because the set of base regions for all $\tilde{f}_j$s are the same, the total performance is increased by computing the kernel just once, for each value of i. Then, the problem reduces to m times computation of a weighted average.

When working in the polar coordinates, ordinary algebraic operations on the variables lead to spurious results, because of the 2π discontinuity. To overcome this problem, rather than directly solving $\{(\mathbf{r}_i; \theta_i) \,|i = 1, \cdots, n\}$, we solve the problem $\{(\mathbf{r}_i; \cos\theta_i, \sin\theta_i) \,|i = 1, \cdots, n\}$ and then find θ_i using ordinary trigonometric methods.

3.5 Eigenimage Theory

Consider the PCA matrix, $\mathbf{V}_{\mathbf{r}}$ and the expectation vector, $\vec{\eta}_{\mathbf{r}}$, corresponding to the homogeneous set $\mathbf{r}$. For the arbitrary color vector $\vec{\mathbf{c}}$, which belongs to $\mathbf{r}$, the PCA coordinates are calculated as, $\vec{\mathbf{c}}' = \mathbf{V}_{\mathbf{r}}^{-1}(\vec{\mathbf{c}} - \vec{\eta}_{\mathbf{r}})$ [44]. Assume that there exists a method for finding an appropriate color set $\mathbf{r}_{\vec{\mathbf{c}}}$ for each color vector $\vec{\mathbf{c}}$. Here, $\mathbf{r}_{\vec{\mathbf{c}}}$ describes the color content of $\vec{\mathbf{c}}$, in the sense that

$$\vec{\mathbf{c}}' = \mathbf{V}_{\mathbf{r}_{\vec{\mathbf{c}}}}^{-1}(\vec{\mathbf{c}} - \vec{\eta}_{\mathbf{r}_{\vec{\mathbf{c}}}}) \tag{12}$$

satisfies

$$\sigma_{c_1'} \gg \sigma_{c_2'} \gg \sigma_{c_3'} \tag{13}$$

where, $\vec{\mathbf{c}}' = [c'_1, c'_2, c'_3]^T$. We call the c'_1, c'_2 and c'_3 components as the $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$, principal components, respectively. The original image can be perfectly reconstructed using these three components, except for numerical errors, as

$$\vec{\mathbf{c}} \simeq \vec{\mathbf{c}}_3 = \mathbf{V}_{\mathbf{r}_{\vec{\mathbf{c}}}} \vec{\mathbf{c}}' + \vec{\eta}_{\mathbf{r}_{\vec{\mathbf{c}}}}. \tag{14}$$

It is proved in [6], theoretically, and in [40], empirically, that for homogeneous swatches, neglecting $\mathbf{pc}_3$ (or even both $\mathbf{pc}_2$ and $\mathbf{pc}_3$), gives a good approximation of the original image. These are called partial reconstructions. Note that the perfect reconstruction scheme of (14) does not rely on (13), while the properness of the partial reconstructions defined as

$$\vec{\mathbf{c}}_2 = \mathbf{V}_{\mathbf{r}_{\vec{\mathbf{c}}}} \begin{bmatrix} c'_1 \\ c'_2 \\ 0 \end{bmatrix} + \vec{\eta}_{\mathbf{r}_{\vec{\mathbf{c}}}} \tag{15}$$

and

$$\vec{\mathbf{c}}_1 = \mathbf{V}_{\mathbf{r}_{\vec{\mathbf{c}}}} \begin{bmatrix} c'_1 \\ 0 \\ 0 \end{bmatrix} + \vec{\eta}_{\mathbf{r}_{\vec{\mathbf{c}}}} \tag{16}$$

do rely on (13).

Now, the goal is to devise an efficient method for calculating, and storing, $\mathbf{V}_{\mathbf{r}_{\vec{\mathbf{c}}}}$ and $\vec{\eta}_{\mathbf{r}_{\vec{\mathbf{c}}}}$ for all the pixels in an image. In the following, a method for doing this task is proposed.

After feeding an image $\mathbf{I}$ into the bi–tree method, The output is the matrix $\boldsymbol{\Upsilon}$ which contains the coordinates of $\mathbf{r}_i$ along with the expectation matrix $\vec{\eta}_i$ and the polarized version of the PCA matrix $(\theta_i, \phi_i, \psi_i)$. Now, consider solving the problem $\{(\mathbf{r}_i; \xi_i) \,|i = 1, \cdots, n\}$ using the block–wise interpolation, where ξ_i is the row vector containing η_{i1}, η_{i2}, η_{i3}, θ_i, ϕ_i and ψ_i. The solution will be the functions $\tilde{\eta}_1, \tilde{\eta}_2, \tilde{\eta}_3, \tilde{\theta}, \tilde{\phi}, \tilde{\psi} : \mathbb{R}^2 \to \mathbb{R}$. Now, we compute the functions $\tilde{\vec{\eta}} : \mathbb{R}^2 \to \mathbb{R}^3$ (expectation map, Emap) and $\tilde{\mathbf{V}} : \mathbb{R}^2 \to \mathbb{R}^9$ (rotation map, Rmap), as the expectation vector and the PCA matrix in each pixel, respectively. This leads to the computation of the three eigenimages $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$ (using (12)).

It can be shown that [45]

$$\sigma^2_{\mathbf{pc}_1} + \sigma^2_{\mathbf{pc}_2} + \sigma^2_{\mathbf{pc}_3} = \sigma^2_{\mathbf{r}} + \sigma^2_{\mathbf{g}} + \sigma^2_{\mathbf{b}}. \tag{17}$$

Here, $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{b}$ are the original RGB components of $\mathbf{I}$. Thus, we define

$$\kappa_i = \frac{\sigma^2_{\mathbf{pc}_1}}{\sigma^2_{\mathbf{pc}_1} + \sigma^2_{\mathbf{pc}_2} + \sigma^2_{\mathbf{pc}_3}} \tag{18}$$

as the energy ratio of the i–th eigenimage. Now, if the PCA matrix is chosen properly

$$\kappa_1 \gg \kappa_2 \gg \kappa_3. \tag{19}$$

In the next two sections, we will show how the eigenimage can be used for effective compression and watermarking of color images.

3.6 Color Image Compression

Consider the image $\mathbf{I}$ and its corresponding eigenimages $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$. Due to the energy compaction condition, given in (19), this scheme can lead to a spectral compression of an image. As such, reconstructing the image using just one or two eigenimage(s) gives the compression ratios of $\frac{3}{2} : 1$ and $3 : 1$, respectively. To add the spatial compression to this scheme, we use the PU–PLVQ gray–scale image compression technique [46] for each eigenimage, independently, with different compression ratios. This leads to the color image compression method proposed in Figure 3.

As Figure 3–a shows, the transmitted information consists of the compressed versions of $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$, along with $\mathbf{\Upsilon}$, α, a, β and b. Assume that the compression is to be applied on an $H \times W$ image, decomposed into n blocks by the bi–tree method. 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 $WH(\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1})$ bytes for storing $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$ 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

$$\lambda \simeq \frac{3}{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \frac{10n}{WH}}. \tag{20}$$

Picking the values of $\lambda_2 = \lambda_1$ and $\lambda_3 = \infty$ leads to $\lambda \simeq 1.5\lambda_1$. Note that using a pure spatial compression approach, all three color components must be compressed with almost the same compression ratios, resulting in a total compression ratio of λ_1. Thus, in the worst case, the proposed method multiples the compression ratio by at least 1.5.

As shown in Figure 3–b, in the decompression process, first, the Emap and the Rmap are computed. Using this information, along with the decoded versions of $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$, the original image is then reconstructed.

3.7 Color Image Watermarking

Consider the image $\mathbf{I}$ with the three corresponding eigenimages of $\mathbf{pc}_1$, $\mathbf{pc}_2$ and $\mathbf{pc}_3$. Although, there is no orthogonality constraint in the eigenimage theory, the eigenimage approach can be adapted for watermarking purposes. Assume that the gray–scale watermark image $\mathbf{W}$ is to be embedded into $\mathbf{I}$. Also, assume that $\mathbf{I}$ and $\mathbf{W}$ are of the same size $(H \times W)$. First, the dynamic range of $\mathbf{W}$ is fitted into that of $\mathbf{pc}_3$, as

$$\tilde{\mathbf{W}} = \frac{\sigma_{\mathbf{pc}_3}}{\sigma_W} \left(\mathbf{W} - \eta_{\mathbf{W}} \right). \tag{21}$$

Now, replacing $\mathbf{pc}_3$ with the scaled version of the watermark signal, $\tilde{\mathbf{W}}$, the watermarked image, $\mathbf{I}'$, is constructed.

The process of extracting the watermark is through computation of the eigenimages corresponding to $\mathbf{I}'$, which we call $\mathbf{pc}'_1$, $\mathbf{pc}'_2$ and $\mathbf{pc}'_3$. Now, $\mathbf{pc}'_3$, when normalized, contains the reconstructed watermark signal

$$\mathbf{W}' = 255 \frac{\mathbf{pc}'_3 - (\eta_{\mathbf{pc}'_3} - \sigma_{\mathbf{pc}'_3})}{2\sigma_{\mathbf{pc}'_3}}. \tag{22}$$

4 Experimental Results

The proposed algorithms are developed in MATLAB 6.5, on an 1100 MHz Pentium III personal computer with 256MB of RAM. For a typical 512×512 image it takes 8.3 seconds to extract the eigenimages. Then, another 4.6 seconds are needed to reconstruct the image. Adding the less than one second needed to do the spatial compression, the total operation takes slightly less than 16 seconds for a typical image. It takes a similar amount of time to embed a watermark in a 512×512 image and about 9 seconds to extract it. Note that these values are measured using MATLAB code and can be reduced drastically if the code is implemented using a higher level programming language.

A database of color images which includes the standard images of *Lena*, *Mandrill*, *Peppers*, and *Couple* as well as some professional color photographs [47],

a total of 140 images, is used in this research. All images are of the size 512×512, in *RGB* color space, and are compressed using standard JPEG with qualities above 95. Prior to the operation, all the images are converted to the standard BMP format.

4.1 Block–Wise Interpolation

To illustrate the results of the proposed block–wise Interpolation method, a sample problem is given in Figure 4–a, with the associated outcome in Figure 4–b, at the *signal to noise ratio* (SNR) of more than 22dB.

4.2 Eigenimages

Figure 5 shows the process of extracting the eigenimages corresponding to the image shown in Figure 5–a, using the proposed method. First, using the bi–tree method with the parameters $\varepsilon_1 = 5$ and $\varrho = 5$, the image is split into 91 blocks (see Figure 5–b). These blocks are then used to generate the EMap and the RMap, as shown in Figures 5–c and (d), respectively. Subsequently, the eigenimages, as shown in Figures 5–e, f, and g, are extracted. Notice that the dynamic ranges of all eigenimages are exaggerated to give a better visualization. The stochastic distributions of the eigenimages are investigated in Figure 5–h. These histograms show the compaction of the energy as discussed next. Using numerical measures, the standard deviations of the eigenimages are computed as $\sigma_{\mathbf{pc}_1} = 52$, $\sigma_{\mathbf{pc}_2} = 12$, and $\sigma_{\mathbf{pc}_3} = 6$. This results in $\kappa_1 = 94\%$, $\kappa_2 = 5\%$, and $\kappa_3 = 1\%$, a clear compaction of the energy in the first eigenimage.

Figure 5–i shows the result of reconstructing the image shown in Figure 5–a using all three eigenimages. Similarly, Figures 5–j and k show the results of ignoring $\mathbf{pc}_3$ and both $\mathbf{pc}_3$ and $\mathbf{pc}_2$, respectively. The resulting PSNR values are 60dB, 38dB, and 31dB, respectively. Note that PSNR of 60dB (instead of infinity), for reconstructing the image using all eigenimages, is caused by numerical errors, while the two other PSNR values (38dB, 31dB) show some loss of information. Note that, in the literature, PSNR values of above 38dB are considered visually satisfactory even for professionals [48].

Figures 6–a, b, and c show the values of κ_1, κ_2 and κ_3 for the image shown in Figure 5–a, at different values of ε_1 and ϱ. Note that except for the trivial cases of $\varrho \leq 2$ and $\varepsilon_1 > 9$, which are never used in practice, more than 90% of the image energy is stored in $\mathbf{pc}_1$, while $\mathbf{pc}_2$ and $\mathbf{pc}_3$ 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 7 shows the PSNR values obtained by reconstructing the image shown in Figure 5–a using all three eigenimages (Figure 7–a), two eigenimages (Figure 7–b), and only one eigenimage (Figure 7–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.

4.3 Color Image Compression

Figure 8 shows the results of the proposed compression method, with the associated numerical values mentioned in Table 1. These results are acquired using $\varepsilon_1 = 5$ and $\varrho = 5$. To compare the compressed images with the original ones, the exaggerated differences are carried in Figure 9. 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 [9]. In that work, the authors mixed the color and texture information into a single vector and then performed the coding using a massively computationally–expensive algorithm. However, the final results show PSNR values of about 40dB for compression ratios of about $20 : 1$. In [33], the researchers separate the compression in the two disjoint domains of spectral and spatial redundancy, by using a PCA neural network. They reach the compression ratio of $3.7 : 1$ with PSNR of around 25dB, while almost all test samples are homogeneous. In [49], the method gives the compression ratio of about $14.5 : 1$ but with the same range of PSNR as ours. As a more recent approach, we compare the PSNR of 31.93dB reported in [19] for *Lena* at 1.17bpp with the results of the proposed method in which PSNR of 33.6dB is achieved at 0.33bpp.

Table 1 also compares the proposed algorithm with JPEG98 and JPEG2000. To do this comparison, each sample image is compressed using each one of these algorithms with the exact compression ratio acquired from the proposed method. As expected, Table 1 shows that JPEG2000 is always giving a better result compared to JPEG98. Comparing the proposed method with JPEG2000, we understand that, except for the case of Figure 8–f, the proposed method is giving a higher PSNR than both others. Numerically, the proposed method is 14% and 7% better than JPEG98 and JPEG2000, respectively, in terms of PSNR. This means more than 3.6dB and 1.9dB increase in the quality when comparing the results of the proposed method with JPEG98 and JPEG2000, respectively.

To demonstrate the utilization of the proposed method in embeding a watermark in an image, we use the example shown in Figure 10. Here, Figure 10–b shows the results of embedding the watermark data shown in Figure 10–a into the image shown in Figure 5–a. The resulting PSNR of this operation is 35dB. Figure 10–c shows the exaggerated difference between the original image and the watermarked image, and Figure 10–d shows the extracted watermark. Investigating Figure 10–c shows where the proposed method hides the data; at each pixel, the direction of the third principal component shows the direction in which data can be put such that it does not affect the visual perception of the image.

To test the robustness of the proposed watermarking method against invasive attacks, the watermark shown in Figure 11 is embedded in the sample image shown in Figure 5–a. Then, the watermarked image is attacked, using *Adobe Photoshop 6.0*, as stated in Table 2. Figure 12 shows some of the attacked watermarked images and Figure 13 show the corresponding extracted watermark data. Investigating Figure 13 shows that the proposed watermarking method is robust against linear and nonlinear geometrical transformations, including rotation, scaling, trimming, and other geometrical distortions. Also, it is robust against occlusion, artistic effects, noise, enhancement operations such as brightening and increasing contrast (even when performed locally), and also lossy compression for ratios up to $15:1$.

Table 3 compares the proposed watermarking method with some of the approaches available in the literature. This table lists the watermark capacity of each method when embedding to a 512×512 color image along with the domain in which the data is embedded. Also, the attack resistance of different approaches is compared here.

It is observed in different experiments that the standard deviation of $\mathbf{pc}_3$ in a typical image is more than 4. Thus, using the proposed watermarking method on a 512×512 color image, at least a same–sized 2bpp image can be used as the watermark signal. This makes the watermark capacity of the proposed method equal to 64KB. This is four times more than the highest capacity of the best available approach [50]. It is worth to mention that only the approaches discussed in [38,51] use color vectors and that only [38] exhibits levels of resistance to linear point operations such as brightening and contrast enhancement. It must be emphasized that the resistance of that method is limited to global operations, while the method proposed in this paper is resistant even to local linear point operations (see Figures 12–f and g). Unfortunately, in the literature, minor attention is paid to non–linear geometrical operations, such as elastic and perspective transformation, and image editing processes,

such as adding text, artistic effects, occlusion and the forth, whilst the copyrighted images can be used in books, posters, and websites where they appear in manipulated forms.

5 Conclusion

In this paper, a PCA–based distance function and the corresponding homogeneity criterion are utilized within a tree decomposition method. The resulting decomposition is then used in order to extract the eigenimages associated with a given image. The proposed eigenimage extraction method is proved to be highly efficient in compacting the image energy as well as in providing proper partial reconstructions of color images. The eigenimages are then used in a color image compression method. Furthermore, the fact that manipulating the third eigenimage has minor effect on the quality of an image is used in a novel watermarking method. Both methods are compared with the available literature. Comparisons show that the proposed compression method is superior to the available approaches (in average 1.9dB more than JPEG2000 in PSNR for the samples used in this paper). The analysis of the proposed watermarking method shows that it is resistant to local alterations as well as to the manipulations performed in publications and websites, such as adding text and non–linear geometrical distortions. No other watermarking approach in the literature is found to claim similar resistance.

Acknowledgement

We wish to thank the respected referees for their constructive suggestions. The first author wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas. We also thank the photographers Ms. *Shohreh Tabatabaii Seifi* and *Mr. Ali Qanavati* for allowing us to use their rich digital image archive.

References

[1] N. Papamarkos, C. Strouthopoulos, I. Andreadis, Multithresholding of color and gray–level images through a neural network technique, Image and Vision Computing 18 (2000) 213–222.

[2] D. Tschumperle, PDE based regularization of multivalued images and applications, Ph.D. thesis, University of Nice, Sophia Antipolis (2002).

[3] H. Cheng, J. Li, Fuzzy homogeneity and scale–space approach to color image segmentation, Pattern Recognition 36 (2003) 1545–1562.

[4] T. Chaira, A. Ray, Fuzzy approach for color region extraction, Pattern Recognition 24 (2003) 1943–1950.

[5] L. Lucchese, S. Mitra, Colour segmentation based on separate anisotropic diffusion of chromatic and achromatic channels, Vision, Image, and Signal Processing 148(3) (2001) 141–150.

[6] G. J. Klinker, S. A. Shafer, T. Kanade, The measurement of highlights in color images, International Journal of Computer Vision 2 (1988) 7–32.

[7] S.-C. Cheng, S.-C. Hsia, Fast algorithm's for color image processing by principal component analysis, Journal of Visual Communication and Image Representation 14 (2003) 184–203.

[8] J. Limb, C. Rubinstein, Statistical dependence between components of a differentially quantized color signal, IEEE Transactions on Communications 20 (1971) 890–899.

[9] D. Carevic, T. Caelli, Region–based coding of color images using karhunen–loeve transform, Graphical Models and Image Processing 59(1) (1997) 27–38.

[10] F. Masulli, A. Massone, A. Schenone, Fuzzy clustering methods for the segmentation of multimodal medical volumes (invited), in: P. Szczepaniak, P. Lisboa, S. Tsumoto (Eds.), Fuzzy Systems in Medicine. Series Studies in Fuzziness and Soft Computing, editor J. Kacprzyk, Springer–Verlag, Heidelberg (Germany), 2000, pp. 335–350.

[11] L. Lucchese, S. Mitra, Color image segmentation: A state–of–the–art survey (invited paper), Image Processing, Vision, and Pattern Recognition, Proc. of the Indian National Science Academy (INSA–A) 67, A, No. 2 (2001) pp. 207–221.

[12] T. Q. Chen, Y. Lu, Color image segmentation – an innovative approach, Pattern Recognition 35 (2002) 395–405.

[13] X. Dai, I. Maeda, Fuzzy–based segmentation of color natural images, in: IEEE ICSP'02, 2002, pp. 969–972.

[14] S.-H. Leung, S.-L. Wang, W.-H. Lau, Lip image segmentation using fuzzy clustering incorporating an elliptic shape function, IEEE Transaction on Image Processing 13(1) (2004) 51–62.

[15] K.-K. Ma, J. Wang, Color distance histogram: A novel descriptor for color image segmentation, in: Seventh International Conference on Control, Automation, Robotics, and Vision (ICARCV'02), Singapore, 2002, pp. 1228–1232.

[16] G. J. Klinker, S. A. Shafer, T. Kanade, A physical approach to color image understanding, International Journal of Computer Vision 4 (1990) 7–38.

[17] D. O. Nikolaev, P. O. Nikolayev, Linear color segmentation and its implementation, Computer Vision and Image Understanding 94 (2004) 115–139.

[18] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3(1) (1991) 71–86.

[19] B. C. Dharaa, B. Chanda, Color image compression based on block truncation coding using pattern fitting principle, Pattern Recognition 40 (2007) 2408–2417.

[20] A. Abadpour, Color image processing using principal component analysis, Master's thesis, Sharif University of Technology, Mathematics Science Department, Tehran, Iran (June 2005).

[21] A. Abadpour, S. Kasaei, Performance analysis of three likelihood measures for color image processing, in: IPM Workshop on Computer Vision, Tehran, Iran, 2004.

[22] Q. Ye, W. Gao, W. Zeng, Color image segmentation using density–based clustering, in: Proceedings of 2003 IEEE International Conference on Acoustic and Signal Processing, Hog Kong, 2003, pp. 401–404.

[23] H. Samet, Region representation: Quadtrees from boundary codes, Comm. ACM 21 (1980) 163:170.

[24] C. Faloutsos, H. Jagadish, , Y. Manolopoulos, Analysis of the n–dimensional quadtree decomposition for arbitrary hyperrectangles, IEEE Transaction on Knowledge and Data Engineering 9, No. 3 (1997) 373–383.

[25] M. Yazdi, A. Zaccarin, Interframe coding using deformable triangles of variable size, in: ICIP'97, 1997, pp. 456–459.

[26] A. Abadpour, S. Kasaei, A new tree decomposition method for color images, in: 10th Annual CSI Computer Conference (CSICC2005), Tehran, Iran, 2005.

[27] ITU–R Recommendation BT–601–5: Studio Encoding Parameters of Digital Television for Standard 4:3 and Widescreen 16:9 Aspect Ratios, www.itu.ch/, Geneva, 1994.

[28] K. Benson, Television Engineering Handbook, Mc Graw–Hill, New York, London, 1992.

[29] J. Slater, Modern Television System to HDTV and Beyond, Pitman, London, 1991.

[30] J. O. Limb, C. Rubinstein, J. Thompson, Digital coding of color video signals–a review, IEEE Transaction of Communication 25(11) (1977) 1349–1385.

[31] W. Pratt, Spatial transform coding of color images, IEEE Transaction on Communication Technology 19(6) (1971) 980–992.

[32] S. Fukuma, M. Iwahashi, N. Kambayashi, Lossless color coordinate transform for lossless color image coding, 1998, pp. 595–598.

[33] C. Clausen, H. Wechsler, Color image compression using PCA and back propagation learning, Pattern Recognition 33 (2000) 1555–1560.

[34] M. Swanson, M. Kobayashi, A. Tewfik, Multimedia data–embedding and watermarking technologies, in: Proceedings of the IEEE, Vol. 86(6), 1998, pp. 1064–1087.

[35] M. Kutter, F. Jordan, F. Bossen, Digital watermarking of color images using amplitude modulation, Electronic Imaging 7(2) (1998) 326–332.

[36] P.-T. Yu, H.-H. Tasi, J.-S. Lin, Digital watermarking based on neural networks for color images, Signal Processing 81 (2001) 663–671.

[37] E. Shahinfard, S. Kasaei, Digital image watermarking using wavelet transform, in: Iranian Conference of Mechatronics Engineering, ICME, Qazvin, Iran, 2003, pp. 363–370.

[38] C.-H. Chou, T.-L. Wu, Embedding color watermarks in color images, EURASIP Journal on Applied Signal Processing 1 (2001) 327–332.

[39] C.-S. C. P.S. Huang, C.-P. Chang, T.-M. Tu, Robust spatial watermarking technique for colour images via direct saturation adjustment, IEE Proceedings Vision, Image & Signal Processing 152 (5) (2005) 561–574.

[40] A. Abadpour, 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.

[41] A. Abadpour, S. Kasaei, A new principle component analysis based colorizing method, in: Proceedings of the 12th Iranian Conference on Electrical Engineering (ICEE2004), Mashhad, Iran, 2004.

[42] A. Abadpour, S. Kasaei, 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.

[43] A. Abadpour, S. Kasaei, Unsupervised, fast and efficient color image copy protection, IEE Proceedings Communications 152 (5) (2005) 605–616.

[44] H. Hotteling, Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology 24 (1933) 417–441.

[45] A. Abadpour, S. Kasaei, An efficient PCA–based color transfer, Visual Communication & Image Representation 18 (1) (2007) 15–34.

[46] S. Kasaei, M. Dericher, B. Biashash, A novel fingerprint image compression technique using wavelet packets and pyramid lattice vector quantization, IEEE Transactions on Image Processing 11(12) (2002) 1365–1378.

[47] S. T. Seifi, A. Qanavati, Digital color image archive, Qnavati@mehr.sharif.edu.

[48] S. Katzenbeisser, A. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Inc., 2000.

[49] Y.-D. Yu, D.-S. Kang, D. Kim, Color image compression based on vector quantization using PCA and LEBD, in: IEEE IENCON'99, 1999, pp. 1259–1262.

[50] M. Barni, F. Bartolini, A. Piva, Multichannel watermarking of color images, IEEE Transaction on Circuits and Systems for Video Technology 12(3) (2002) 142–156.

[51] P. Tsai, Y.-C. Hu, C.-C. Chang, A color image watermarking scheme based on color quantization, Signal Processing 84 (2004) 95–106.

[52] J. Ruanaidh, W. Dowling, F. Boland, Watermarking digital images for copyright protection, IEE Proceedings on Vision, Signal and Image Processing 143(4) (1996) 250–256.

[53] A. Nikoladis, I. Pitas, Robust watermarking of facial images based on salient geometric pattern matching, IEEE Transaction on Multimedia 2(3) (2000) 172–184.

[54] M.-S. Hsieh, D.-C. Tseng, Y.-H. Huang, Hiding digital watermarks using multiresolution wavelet transform, IEEE Transaction on Industrial Electronics 48(5) (2001) 875–882.

[55] M. Kutter, S. Winkler, A vision–based masking model for spread–spectrum image watermarking, IEEE Transaction on Image Processing 11(1) (2002) 16–25.

[56] C.-W. Tang, H.-M. Hang, A feature–based robust digital image watermarking scheme, IEEE Transaction on Signal Processing 51(4) (2003) 950–959.

[57] X. Kang, J. Huang, Y. Q. Shi, Y. Lin, A dwt–dft composite watermarking scheme robust to both affine transform and jpeg compression, IEEE Transactions on Circuits and Systems for Video Technology 13(8) (2003) 776–786.

[58] Shih-Hao, Y.-P. Lin, Wavelet tree quantization for copyright protection watermarking, IEEE Transactions on Image Processing 13(2) (2004) 154–165.

6 Response to the Comments

We wish to thank the respected editor-in-chief and the respected referee for their constructive suggestions. In the following the comments are discussed accompanied by descriptions on how they are addressed.

6.1 Comments of the Respected Editor–in–Chief

- "*revise the manuscript in the context of the more recent work in the literature.*",

The manuscript is completely rewritten and the results are compared with more recent works in the literature.

6.2 Comments of the Respected Referee

- "*In my opinion the paper may be made more compact with the proper references to author's previous work*",
  Performed as suggested.
- "*in section 2.1 the requirement of fuzzy membership function needs slightly more explanation*",
  The section is removed in the revised version.
- "*section 2.5 may be elaborated more*",
  Performed as suggested.
- "*in proposed algorithm part few sections may begin with a few lines of subjective explanation.*", Performed as suggested.

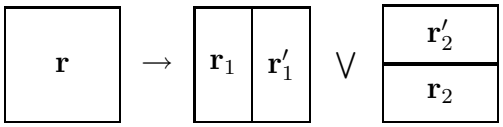


Fig. 1. Bi–tree decomposition method.

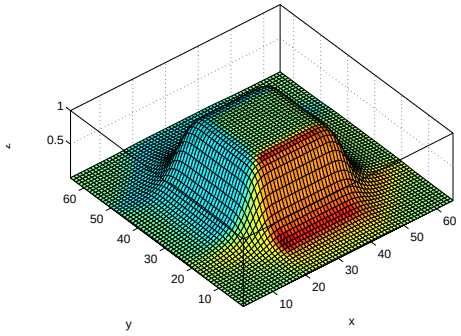
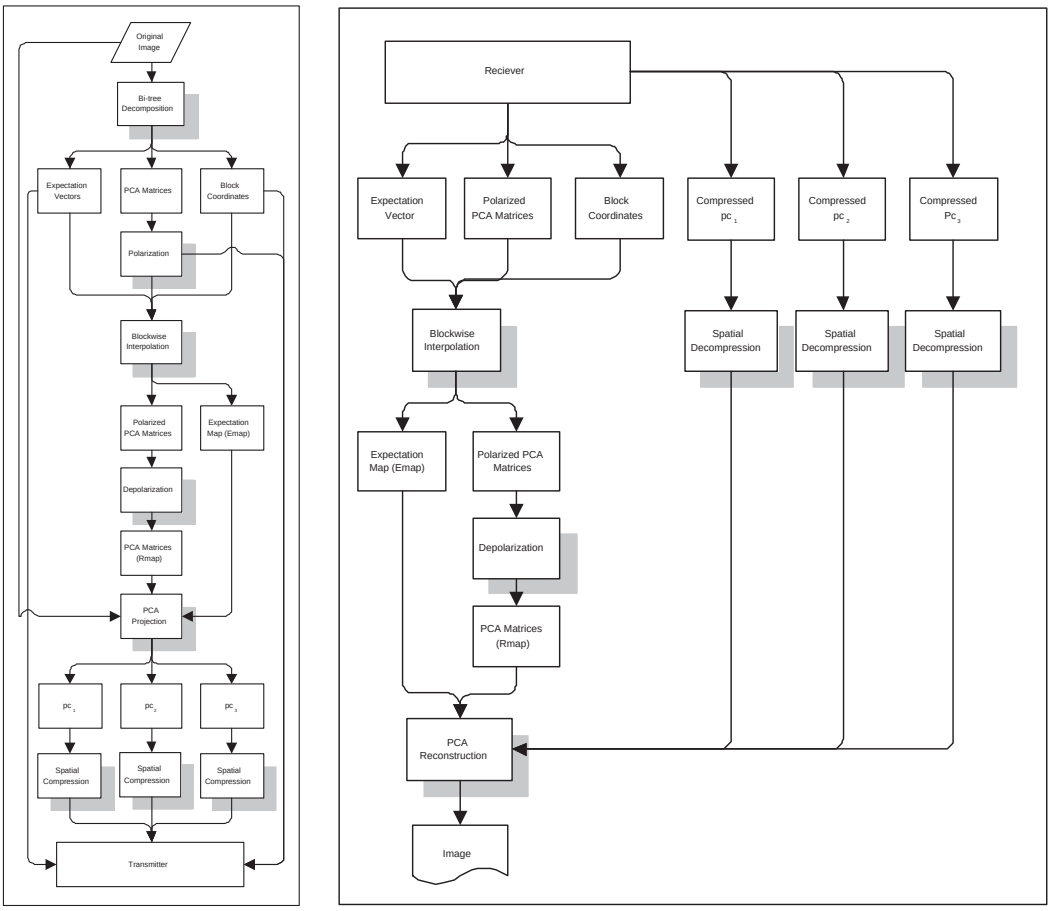


Fig. 2. Typical shape of the proposed interpolation kernel $B^{w,h}_{\tau,N}(x,y)$.

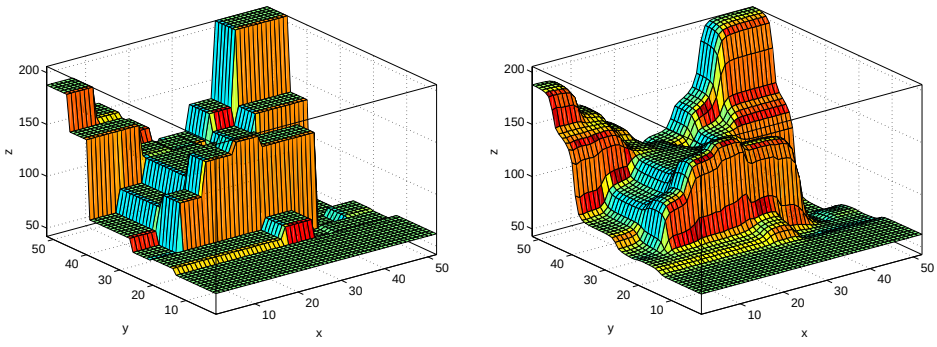
Table 1
Numerical information corresponding to the results shown in Figure 8. [n: block count. λ: Compression Ratio, *bpp*: bit per pixel.]

Sample	**Proposed Method**										**JPEG**	**JPEG2000**
	n	$\mathbf{pc}_1$		$\mathbf{pc}_2$		$\mathbf{pc}_3$		**Total**				
		λ	*bpp*	λ	*bpp*	λ	*bpp*	λ	*bpp*	PSNR (dB)	PSNR (dB)	PSNR (dB)
8–a	334	42.6 : 1	0.19	58.8 : 1	0.14	∞	0	71.1 : 1	0.34	28.4	26.8	27.1
8–b	60	43.1 : 1	0.19	56.2 : 1	0.14	∞	0	72.6 : 1	0.33	33.6	28.9	30.7
8–c	161	39.8 : 1	0.20	55.1 : 1	0.15	∞	0	68.0 : 1	0.35	30.4	27.7	29.5
8–d	311	44.4 : 1	0.18	58.0 : 1	0.14	∞	0	72.5 : 1	0.33	24.5	19.8	21.7
8–e	203	41.3 : 1	0.19	56.9 : 1	0.14	∞	0	70.1 : 1	0.34	33.3	30.5	32.4
8–f	32	45.5 : 1	0.18	62.8 : 1	0.13	∞	0	78.8 : 1	0.30	26.1	38.4	42.4
8–g	91	41.3 : 1	0.19	57.7 : 1	0.14	∞	0	71.4 : 1	0.34	35.4	30.2	32.7



(a) (b)

Fig. 3. Flowchart of the proposed color image compression method. (a) Compression. (b) Decompression.



(a) (b)

Fig. 4. Proposed block–wise interpolation. (a) Sample problem. (b) Solution provided by the proposed block–wise interpolation (SNR> 22dB).

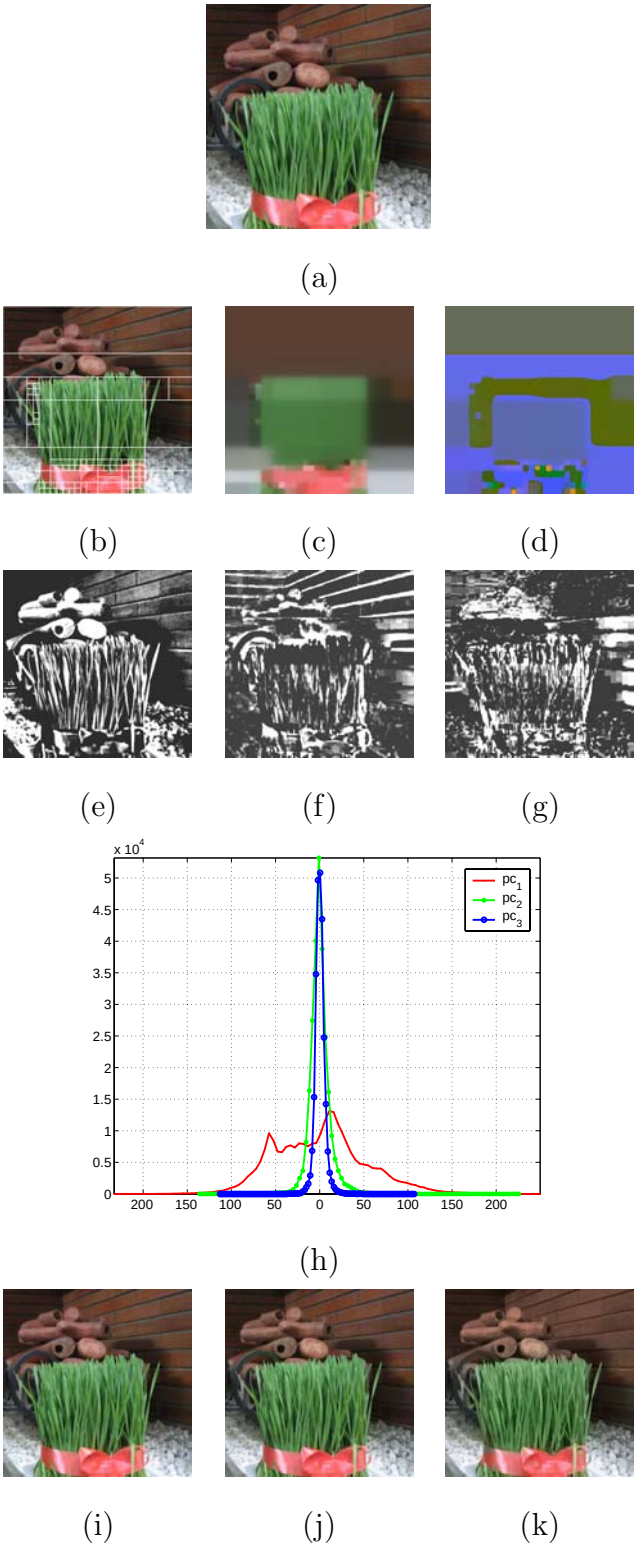
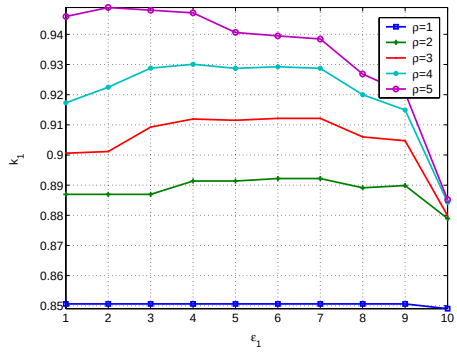
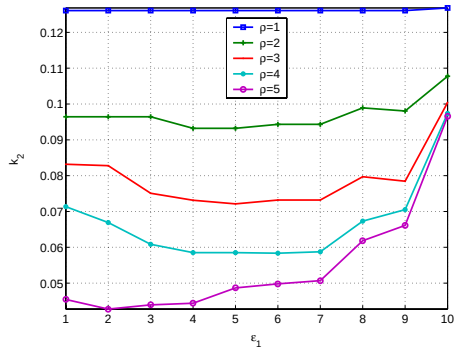


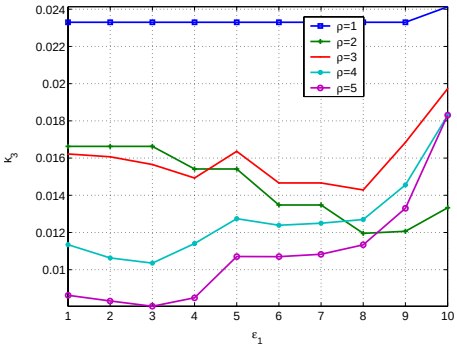
Fig. 5. Process of extracting eigenimages from an arbitrary image. (a) Original image adopted from [47]. (b) Outcome of bi–tree decomposition method. (c) Emap. (d) Rmap. (e) $\mathbf{pc}_1$. (f) $\mathbf{pc}_2$. (g) $\mathbf{pc}_3$. (h) Histograms of eigenimages. (i) Reconstruction using all eigenimages (PSNR=60dB). (j) Ignoring one eigenimage (PSNR=38dB). (k) Ignoring two eigenimages (PSNR=31dB).



(a)

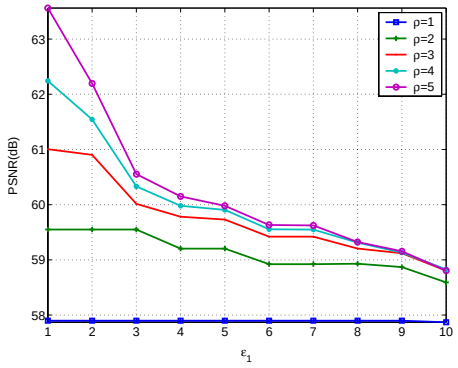


(b)

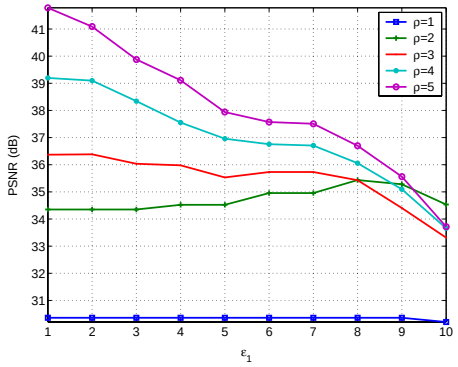


(c)

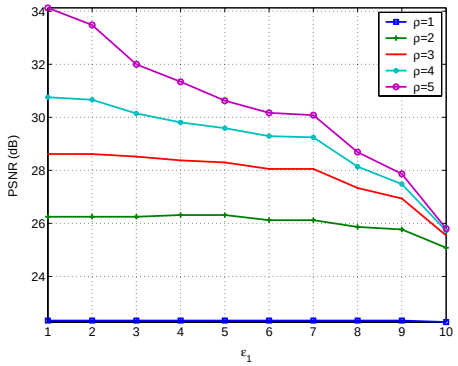
Fig. 6. Distribution of energy between three eigenimages, for different values of ε_1 and ϱ. (a)κ_1. (b)κ_2. (c)κ_3.



(a)



(b)



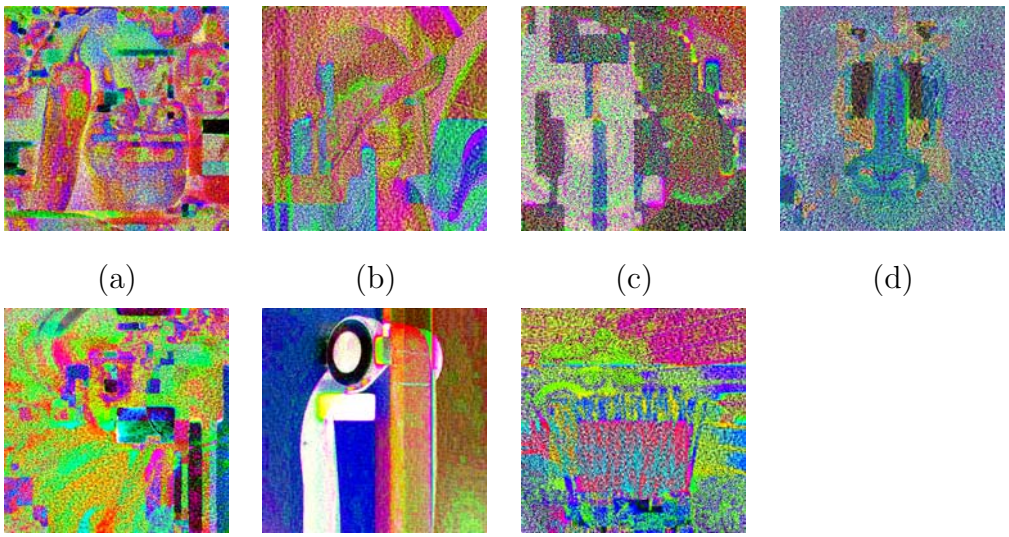
(c)

Fig. 7. PSNR values corresponding to partially–reconstructed images, for different values of ε_1 and ϱ. (a) Using three eigenimages. (b) Using two eigenimages. (c) Using only one eigenimage.

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

(e) (f) (g)

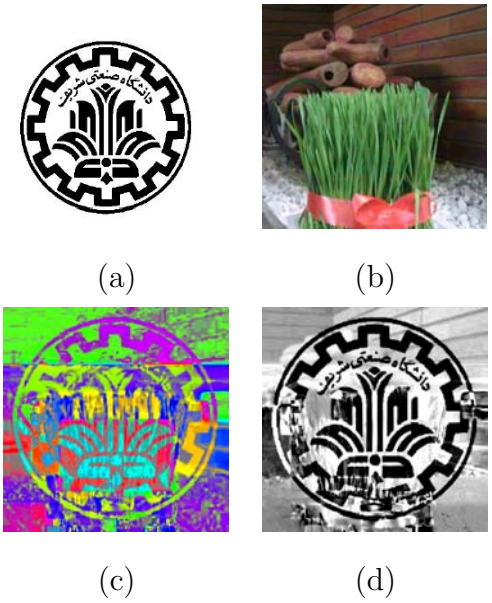
Fig. 8. Results of proposed compression method. For details about compression ratio and PSNR see Table 1.



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

(e) (f) (g)

Fig. 9. Exaggerated error of the proposed compression method.

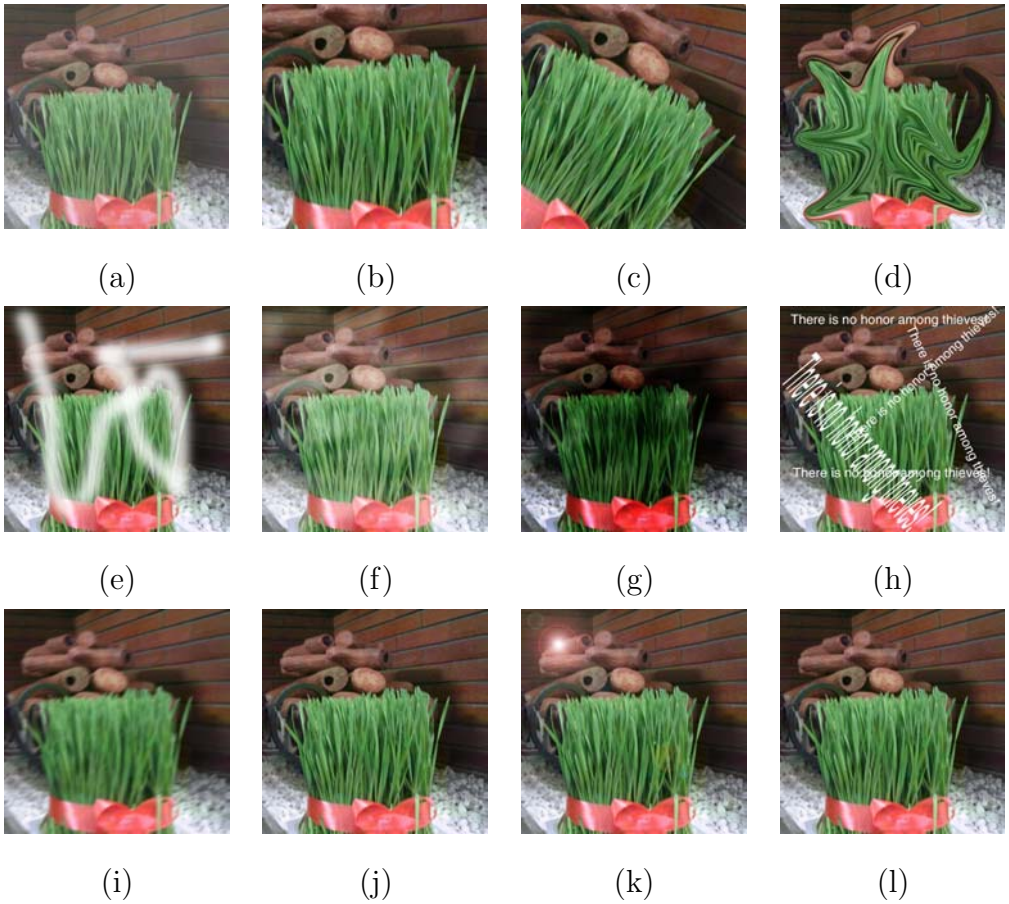


(a) (b)

(c) (d)

Fig. 10. Results of the proposed watermarking method performed on the image shown in Figure 5–a. (a) Watermark data. (b) Watermarked image at PSNR=35dB. (c) Exaggerated difference between the original and watermarked images. (d) Extracted watermark data.

Fig. 11. Sample watermark.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 12. Some attacked watermarked images, corresponding items in Table 2 are: (a):1, (b):2, (c):3, (d):5, (e):9, (f):11, (g):12, (h):13, (i):14, (j):23, (k):28, and (l):29

Table 2
Some of the attacks performed on watermarked images to analyze the robustness of the proposed watermarking method.

index	Attack
0	–
1	Enhancement
2	Scaling and trimming
3	Rotation, scaling and trimming
4–7	Nonlinear Geomtrical
8	Gaussian Lowpass
9,10	Scratch
11,12	Local brightening/darkening
13	Text
14	Motion Blur
15	Radial Blur
16–22	Geometrical Distortion
23	Uniform Color Noise
24	Gaussian Color Noise
25	Uniform Gray Noise
26	Gaussian Gray Noise
27	Median
28	Artistic Lens Flare
29	Lossy Compression

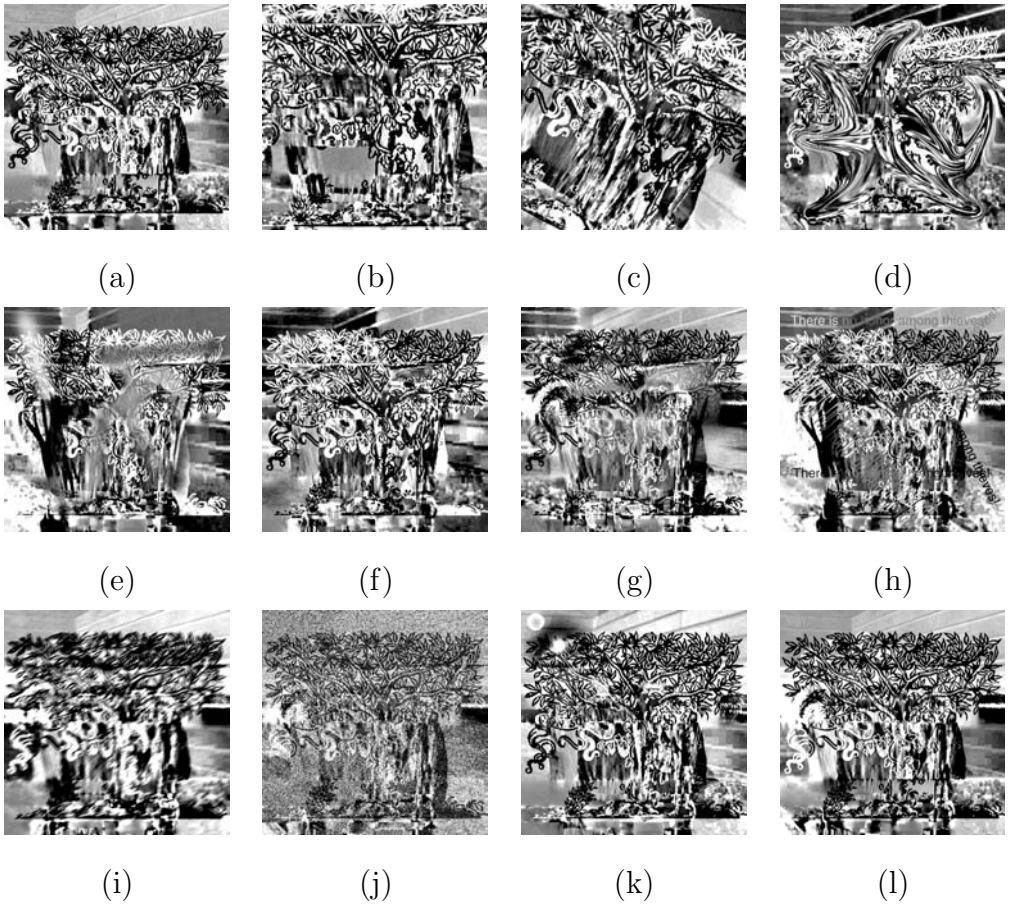


Fig. 13. Extracted watermarks from attacked images shown in Figure 12.

Table 3
Comparison of different watermarking approaches with the proposed method, all used in a 512 × 512 image. –: Not Resistant. ∼: Partially Resistant. √: Completely Resistant. [Abbreviations: G: Grayscale, SCC: Single Color Component, CV: Color Vector, LG: Linear Geometrical Transformation, NLG: Nonlinear Geometrical Transformation, LPO: Linear Point Operations, NLPO:Nonlinear Point Operations, SO: Spatial Domain Operations, EO: Editing Operations, CMP: JPEG Compression].

Method		[52]	[53]	[54]	[38]	[36]	[50]	[55]	[56]	[57]	[58]	[51]	Proposed
Capacity		4KB	8KB	8KB	2KB	1KB	16KB	8B	0.5KB	60B	64B	2KB	64KB
Domain		G	G	G	CV	SCC	SCC	SCC	G	G	G	CV	CV
Attack	LG	∼	√	∼	∼	∼	√	–	√	√	∼	∼	√
	NLG	–	–	–	–	–	–	–	–	–	–	–	√
	LPO	–	–	–	∼	–	–	–	–	–	–	–	√
	NLPO	–	∼	∼	∼	∼	∼	–	∼	–	–	∼	√
	SO	–	∼	∼	∼	∼	∼	–	∼	∼	∼	∼	√
	EO	–	–	–	–	–	–	–	–	–	–	–	√
	CMP	√	√	√	√	√	√	∼	√	√	√	√	√