10th Annual
Conference of Computer Society of Iran
February 15-17, 2005

A New Tree Decomposition Method for Color Images

Arash Abadpour and Shohreh Kasaei
Sharif University of Technology, Tehran, Iran

***Abstract*—Two decades ago, quad-tree decomposition was proposed as a fast tool for decomposing an image into a set of homogenous rectangular regions. Since then, many researchers have tried to generalize the method to enhance the results. In this paper, a new tree decomposition method is proposed which uses the rectangular regions in a more adaptive way. This paper gives a comprehensive performance comparison between the quad-tree and the new proposed bi-tree decomposition methods.**

I. Introduction

Quad-tree decomposition is the well-known method for splitting an image into homogenous sub-blocks, resulting in a very coarse, but fast, segmentation [1]. Defining the whole image as a single block, the method is performed according to a problem-specific *homogeneity criteria*. Work on the generalization of quad-tree decomposition has been performed, both on the dimension [2] and the shape [3] of the blocks. Using rectangles as the sub-blocks is known to have many benefits. Firstly, performing block-wise operations in rectangular regions is computationally cheap. Also, more complicated shapes of blocks, such as triangles, perform *division* operator on spatial variables, which leads to more *round-off* and *misalignment* errors. In the conventional quad-tree decomposition method, when a non-homogenous block is encountered, it is split into four sub-blocks. The *one-split-to-four* rule is optimized in this paper by using a novel *one-split-to-two* rule, resulting in a method called *bi-tree decompotision*.

Color is one of the most important tools for object discrimination by human observers, but it is overlooked in the past [4]. Discarding the intrinsic characteristics of color images (as *vector geometries* [5]), some researchers have assumed color images as *parallel* gray-scale images [6], [7], [8], [9]. It has been proved that the *principle component analysis* (PCA) is an appropriate vectorial descriptor for natural color images [10], [11], [12]. To use any tree decomposition, a suitable homogeneity criteria is needed. In [10], 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, called the *linear partial reconstruction error* (LPRE). The LPRE distance of vector $\vec{c}$ to cluster r is defined as:

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

where $\vec{v}$ denotes the direction of the first principal component and $\|\vec{x}\|$ is the normalized L_1 norm $\|\vec{x}\| = \sum_{i=1}^{N}|x_i|/N$. In [10], the authors proposed to use the following stochastic margin to compute the homogeneity of the selected region r,

$$\|f\|_{r,p} = arg_e\{P_{\vec{x}\in r}\{f(\vec{x}) \leq e\} \geq p\} \quad (2)$$

where p is the inclusion percentage and $P_{\vec{x}\in r}\{f(\vec{x}) \leq e\}$ denotes the probability of x being less than or equal to e. It is proved that $\|\tau_r\|_{r,p}$ is a proper homogeneity criteria for the quad-tree decomposition [11]. The comparison of the LPRE-based homogeneity criteria with *Euclidean* and *Mahalanobis* has proved its superiority [11].

In this paper, we compare the performance of the two decomposition methods of quad-tree and bi-tree. The performance analysis is performed using color images and the LPRE-based homogeneity criteria. The rest of this paper is organized as follows: Section II introduces the proposed bi-tree decomposition method. Section III holds the experimental results. Finally, Section IV concludes the paper.

II. Proposed Method

As discussed in Section I, having a suitable homogeneity criteria, the image can be decomposed into homogenous regions (here we use $\|e_r^*\|_{r,p}$). Starting with the entire image area, the tree is produced using the homogeneity criteria defined as:

$$h(r) = \begin{cases} 1, & \|\tau_r\|_{r,p} \leq \varepsilon_1 \\ 0, & otherwise \end{cases} \quad (3)$$

where, ε_1 is a user-selected parameter, mostly in the range of $[1\ldots10]$ and $\|\tau_r\|_{r,p}$ is the LPRE-based homogeneity criteria discussed in Section I. In an $W \times H$ image, the *depth* of a $w \times h$ block r is defined as:

$$\varrho_r = \max\left\{\log_2\frac{W}{w}, \log_2\frac{H}{h}\right\} \quad (4)$$

where 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 $\{x_1, y_1, x_2, y_2, \eta_1, \eta_2, \eta_3, V_{ij}, i, j = 1, \cdots, 3\}$ and some reserved parameters. Where, $[\eta_1, \eta_2, \eta_3]^T$ is the expectation of color information of the region r and V_{ij}s are the elements of the PCA Matrix V corresponding to block r.

Here, we propose a new tree decomposition method called bi–tree decomposition. Assume that the image I is fed into the *bi–tree* decomposition method. 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. Thus, if:

$$\|\tau_{r_1}\|_{r_1,p} + \|\tau_{r_1'}\|_{r_1',p} < \|\tau_{r_2}\|_{r_2,p} + \|\tau_{r_2'}\|_{r_2',p} \tag{5}$$

and the depth limitation permits, the block is split vertically and otherwise (if the depth limitation is met) it is split horizontally. In the new method, the rectangular clipping is reserved while the block shape changes such that it best fits into the image details.

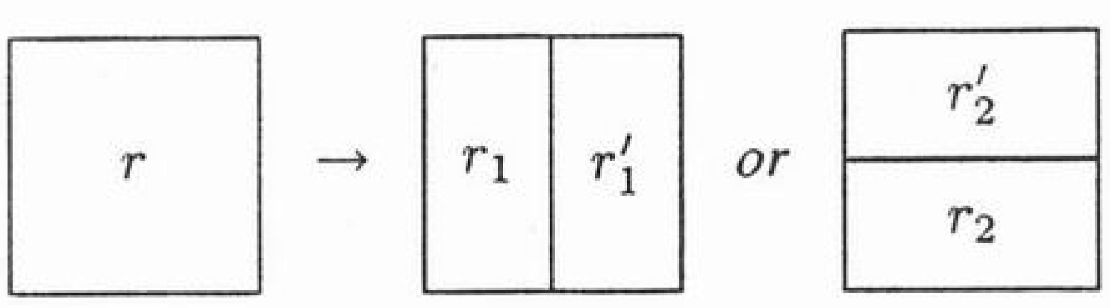

Fig. 1. The proposed bi–tree decomposition method.

Any tree decomposition method (containing the quad–tree and the bi–tree), when considered as a black box, processes the input image (in terms of given values of ε_1 and ϱ), resulting in n blocks. No quantitative measure for tree decomposition performance analysis exists in the literature. Here, we propose three measures, concerning the non–homogeneity of the resulting blocks, the average size of them, and the size spread.

It is crucial to measure the performance of a tree decomposition method in terms of giving blocks with the non–homogeneity values not much bigger than the initial value of ε_1. Here, we propose the *average homogeneity factor* defined as,

$$\bar{\varepsilon}_1 = \sum_{i=1}^{n} \frac{\|r_i\|}{\|I\|} \|\tau_{r_i}\|_{r_i,p} \tag{6}$$

where $\|r\|$ denotes the area of r. Note that $\bar{\varepsilon}_1$ measures the average homogeneity of regions weighted by their areas.

Also, it is important to reject methods resulting in many small regions, even if they are all homogenous to great extents. One step of both horizontally and vertically splitting of R (as performed in the quad–tree decomposition method) results in block r, which satisfies $\|r\| = \frac{1}{4}\|R\|$. Thus, it is reasonable to define,

$$\tilde{\varrho}_r = \log_4 \frac{\|I\|}{\|r\|} \tag{7}$$

as the number of splitting steps needed to acquire region r out of image I. Deriving (7) yields:

$$\tilde{\varrho}_r = \log_4 \frac{WH}{wh} = \frac{1}{2}(\log_2 \frac{W}{w} + \log_2 \frac{H}{h}). \tag{8}$$

Comparing (8) with (4) shows the similarity between ϱ_r and $\tilde{\varrho}_r$. It is worth to mention that we are forced to use the max operator in (4) (to avoid ultra thin and tall blocks) but we use the *mean* operator in (8) to increase the precision of $\tilde{\varrho}_r$; the two 3×10 and 3×20 blocks in no sense have the same depth while their corresponding values of ϱ are identical. We propose the expectation of $\tilde{\varrho}_r$ as the average depth of the tree, defined as:

$$\tilde{\varrho} = \frac{1}{n} \sum_{i=1}^{n} \log_4 \frac{\|I\|}{\|r_i\|}. \tag{9}$$

Assume Deriving (9) as:

$$4^{-\tilde{\varrho}} = \sqrt[n]{\prod_{i=1}^{n} \frac{\|r_i\|}{\|I\|}}. \tag{10}$$

Having in mind that the sum of the areas of all sub–blocks equals the area of the original image we have:

$$\sum_{i=1}^{n} \frac{\|r_i\|}{\|I\|} = 1. \tag{11}$$

It shows that $4^{-\tilde{\varrho}}$ is the *geometrical median* of n values with summation equal to one. Note that for a fixed value of n, $4^{-\tilde{\varrho}}$ gives its largest value (or similarly $\tilde{\varrho}$ gives its smallest value), for the case of equality we have:

$$\frac{\|r_i\|}{\|I\|} = c \rightarrow \|r_i\| = \frac{1}{n}\|I\| \tag{12}$$

which results in,

$$\tilde{\varrho} = \frac{1}{n} n \log_4 n = \log_4 n. \tag{13}$$

Now, defining,

$$\tilde{\varphi} = \frac{\tilde{\varrho}}{\log_4 n} = \frac{\sum_{i=1}^{n} \log_4 \frac{\|r_i\|}{\|I\|}}{n \log_4 n} \tag{14}$$

$\tilde{\varphi}$ gets to its minimum value, unity, when all blocks have the same size. Also, for values of $\tilde{\varphi} \simeq 1$, we have $\tilde{\varrho} \simeq \log_4 n$. We propose computing $\tilde{\varphi}$ as a criteria for the proper distribution of the areas of the tree nodes. Note that we prefer tree decomposition methods leading to $\tilde{\varepsilon}_1 \leq \varepsilon_1$ and $\tilde{\varrho} \leq \varrho$ with values of $\tilde{\varphi}$ near unity.

III. Experimental Results

The proposed algorithms are developed in *MATLAB 6.5*, on an *1100 MHz*, *Pentium III*, personal computer with $256MB$ of *RAM*. The code is available online at *http://math.sharif.edu/~abadpour*.

A database of color images (140 samples) including the standard images of *Lena*, *Mandrill*, *Airplane*, *Peppers*, *Girl*, and *Couple* and also some professional color photographs [13] is used. All images are of size 512×512, in RGB color space, and compressed using standard jpeg compression with compression ratio of about 3 : 1.

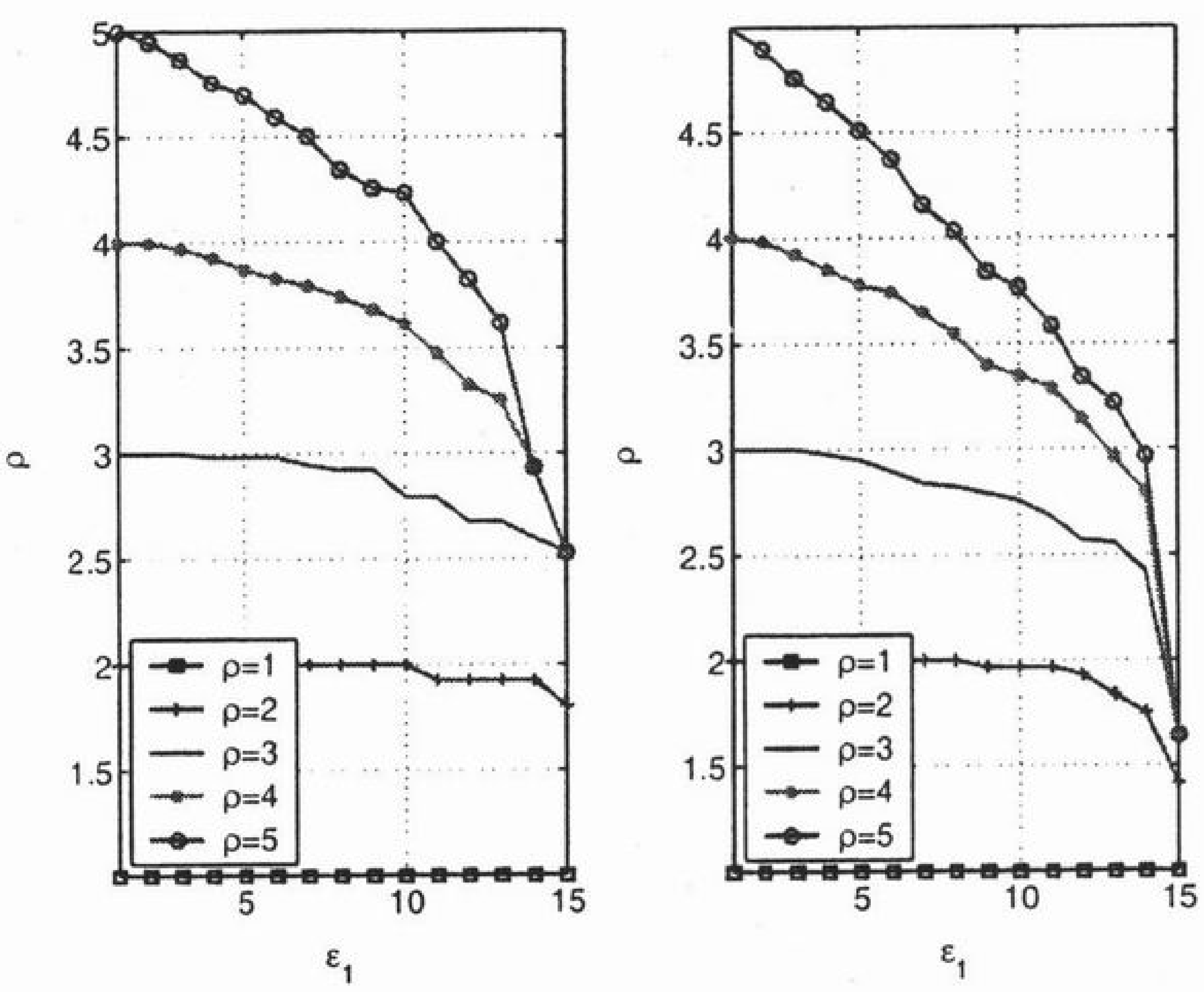

Fig. 2. Values of $\tilde{\varrho}$ for different values of ϱ and ε_1, left: quad–tree method and right: proposed bi–tree method.

To compare the performance of the quad–tree and the bi–tree decomposition methods, each sample image is decomposed by each method using values of $\varepsilon_1 = 1, \cdots, 15$ and $\varrho = 1, \cdots, 5$ and the results are recorded. Figure 2 shows the values of $\tilde{\varrho}$ for different values of ϱ and ε_1 in the quad–tree and bi–tree methods. Figure 2, shows that in both algorithms in all sets of parameters we have $\tilde{\varrho} \geq \varrho$, with the equality in smaller values of ε_1. Also, as desired, $\tilde{\varrho}$ is a monotonic non–increasing function of ε_1. Inspecting the falling tender of the $\tilde{\varrho}$ with increasing ε_1 gives the promising result that although in both methods by increasing the marginal acceptable non–homogeneity (relieving the method) the average splitting decreases, but bi–tree gives less splitting in average. For example, compare the number of blocks at $\varepsilon_1 = 10$ for each set of corresponding curves in the left and right graphs. For the quad–tree the values are approximately $\{1.00, 2.00, 2.80, 3.61, 4.23\}$ while the corresponding results for the bi–tree method are $\{1.00, 1.97, 2.76, 3.35, 3.77\}$, about 9% better than that of the quad–tree for the largest ρ.

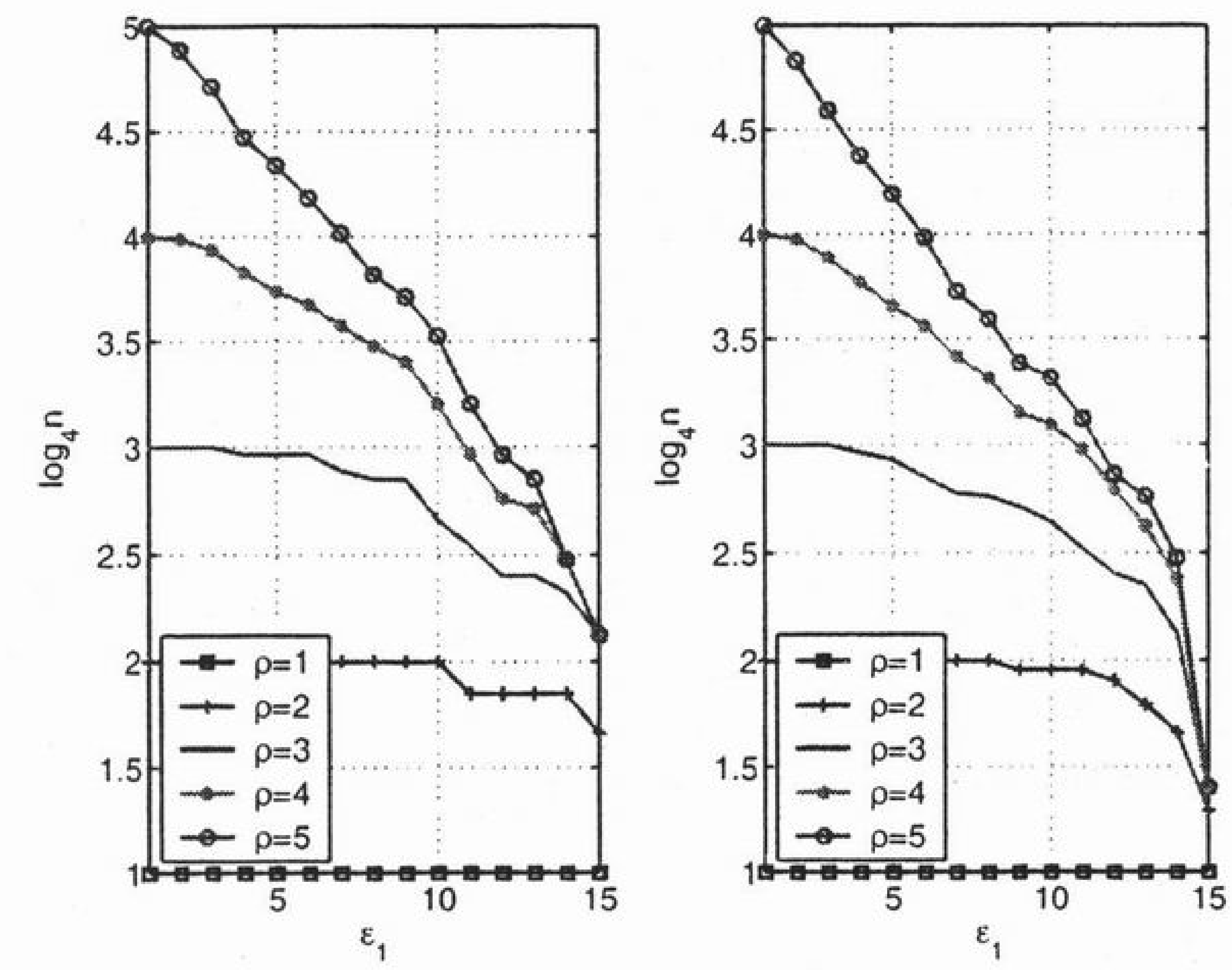

Fig. 3. Values of $\log_4 n$ for different values of ϱ and ε_1, left: quad–tree method and right: proposed bi–tree method.

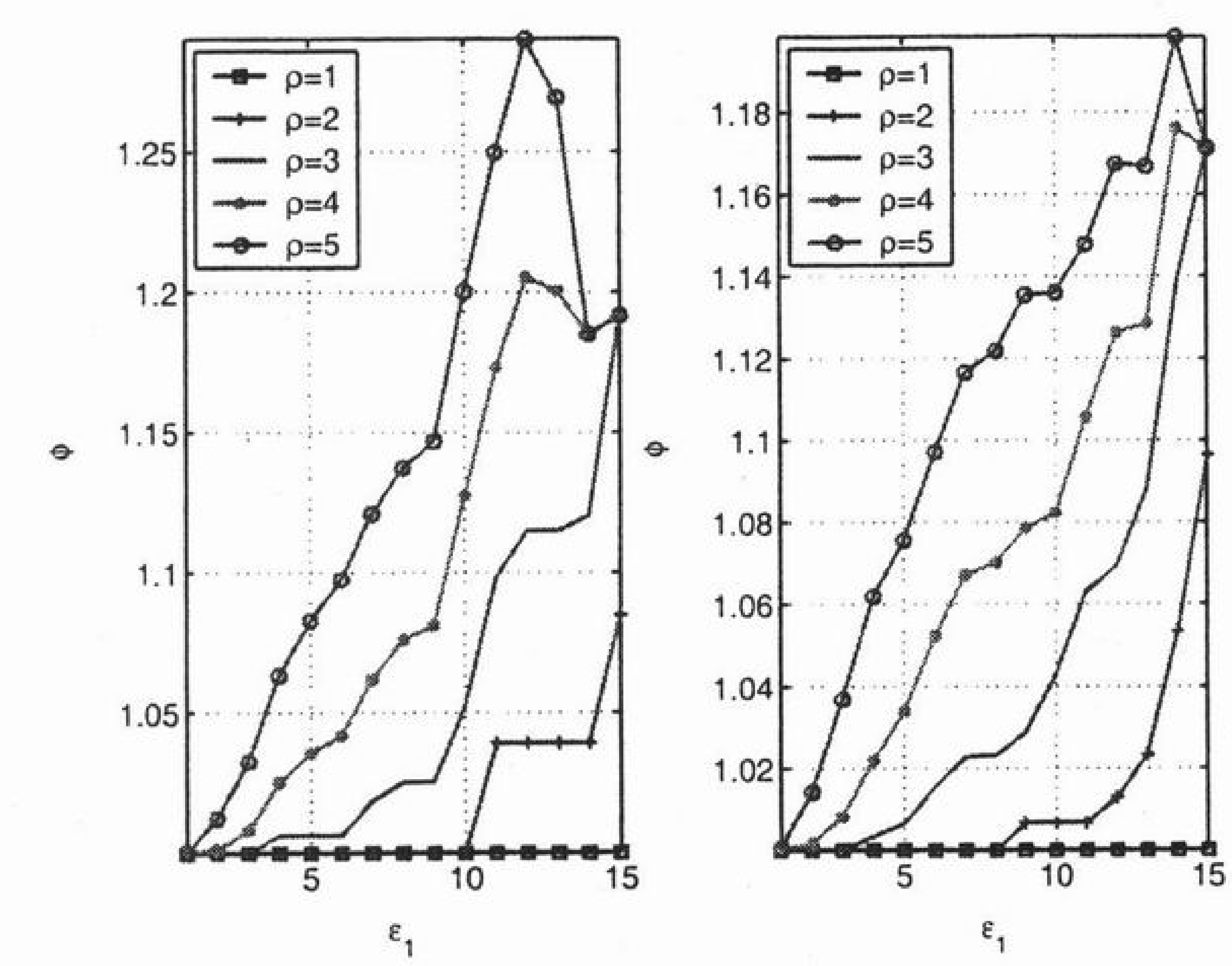

Fig. 4. Values of $\tilde{\varphi}$ for different values of ϱ and ε_1, left: quad–tree method and right: proposed bi–tree method.

Figure 3 shows the values of $\log_4 n$ for different values of ϱ and ε_1 in the quad–tree and the bi–tree methods. While, as expected, n is a monotonic non–increasing function of ε_1, there are less blocks produced by the bi–tree compared with the quad–tree, when used in the same conditions. Namely, in the case of $\varepsilon_1 = 10$, the bi–tree has produced $\{4, 15, 39, 73, 99\}$ number of blocks for consecutive values of ρ, while quad–tree has produced $\{4, 16, 40, 85, 133\}$ number of blocks (averagely 12% more). When tree depth increases, the quad–tree results in much more block numbers (34% more for $\rho = 5$). This inflated number of blocks has a negative impact on the speed performance of the afterwards stages of the process.

Figure 4 shows the values of $\tilde{\varphi}$ for different values of ϱ and ε_1 in the two methods. As shown in Figure 4, for both methods $\tilde{\varphi}$ is always less than 1.3. While the range of $\tilde{\varphi}$ in bi–tree occupies the [1, 1.2] interval, the quad–tree produces $\bar{\varphi}$ values in the range of [1, 1.3]. Thus, the size scatter is about 50% wider in the quad–tree when compared to the bi–tree. This means that the result of the quad–tree is more non–homogenous in its size pattern.

Figure 5 shows the values of $\frac{\bar{\varepsilon}_1}{\varepsilon_1}$ for different values of ϱ and ε_1 using the two decomposition methods of quad–tree and bi–tree. The mean square difference between corresponding points of these two sets of curves is 0.059. Thus, in terms of final homogeneity, the two tree decomposition methods are almost identical.

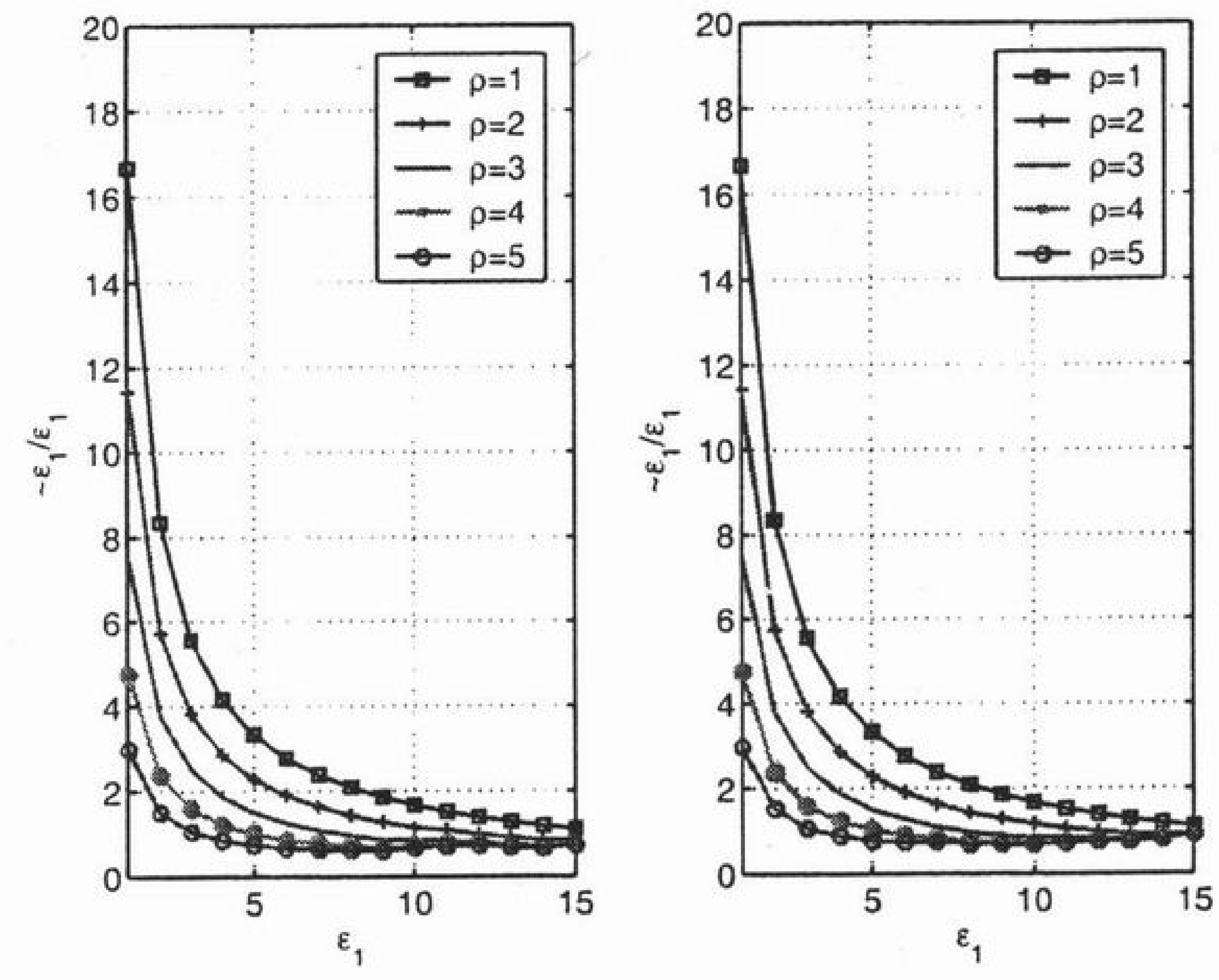

Fig. 5. Values of $\frac{\bar{\varepsilon}_1}{\varepsilon_1}$ for different values of ϱ and ε_1, left: quad–tree method and right: proposed bi–tree method.

Figure 7 shows the time elapsed by the two decomposition methods with different values of ϱ and ε_1. The proposed bi–tree elapses almost 3.5 times more computation cost. This is due to the decision stage in the proposed bi–tree method. Figure 6 shows some typical results of the quad–tree and proposed bi–tree decomposition methods.

IV. Conclusions

A novel tree decomposition method is proposed. Comparing the proposed bi–tree and the quad–tree decomposition methods, bi–tree method produces less blocks with almost identical sizes, while the quad–tree splits the image more with a more scattered pattern of block sizes. The total homogeneity of the resulting blocks are almost the same in both methods, while bi–tree is more time consuming. Further optimizations must be performed on the bi–tree implementation to increase its time performance.

Acknowledgement

The first author wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas.

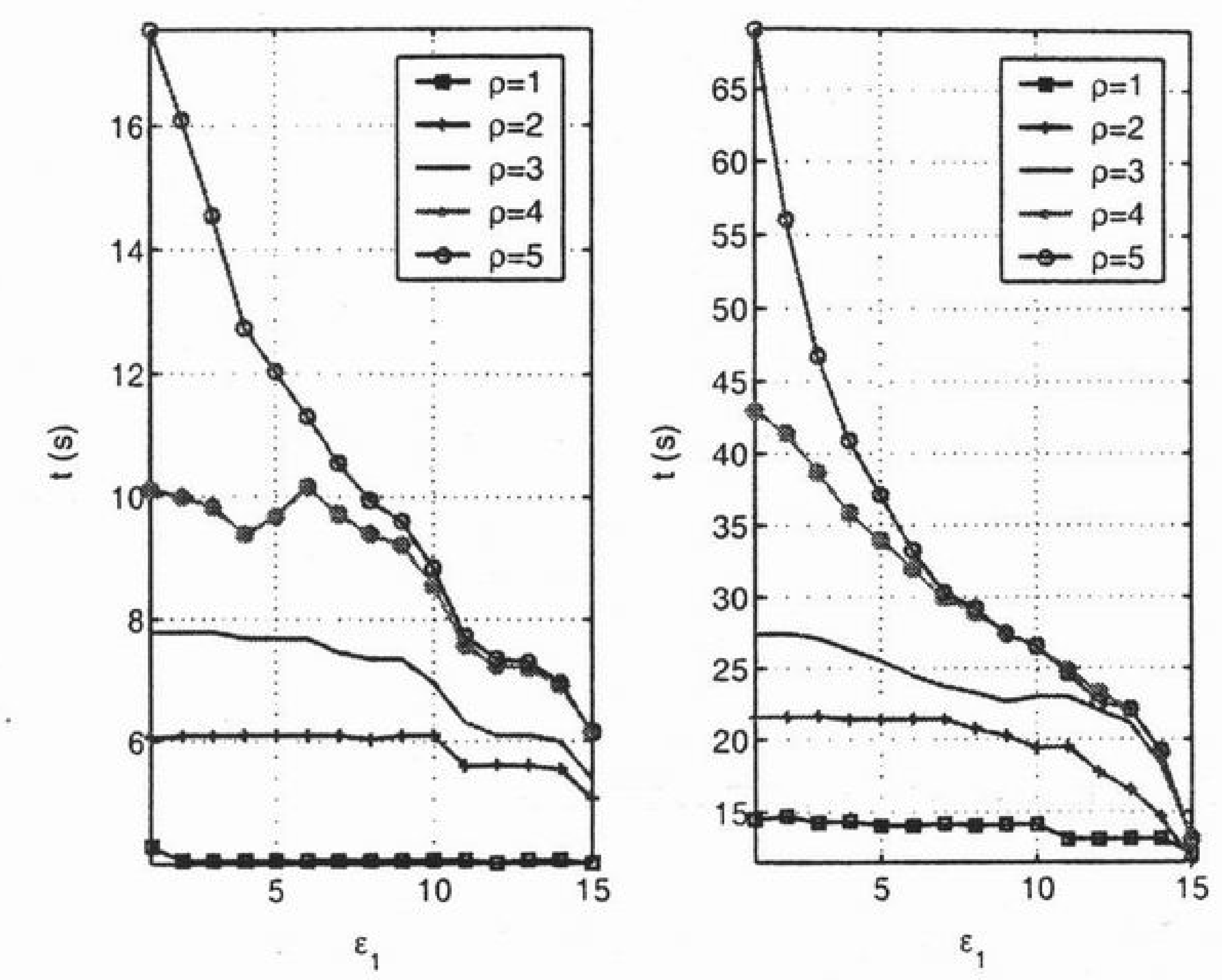

Fig. 7. The elapsed time by the decomposition methods for different values of ϱ and ε_1, left: quad–tree method and right: proposed bi–tree method.

References

[1] H. Samet, "Region representation: Quadtrees from boundary codes," *Comm. ACM*, vol. 21, p. 163:170, 1980.

[2] 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.

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

[4] 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.

[5] D. Tschumperle, "Pde's based regularization of multivalued images and applications," Ph.D. dissertation, University of Nice, Sophia Antipolis, 2002.

[6] H. Cheng and J. Li, "Fuzzy homogeneity and scale–space approach to color image segmentation," *Pattern Recognition*, vol. 36, pp. 1545–1562, 2003.

[7] J. Bruce, T. Balch, and M. Veloso, "Fast and cheap color image segmentation for interactive robots," in *Proceedings of IROS–2000*, Japan, 2000.

[8] T. Chaira and A. Ray, "Fuzzy approach for color region extraction," *Pattern Recognition*, vol. 24, pp. 1943–1950, 2003.

[9] 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.

[10] 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.

[11] ——, "Performance analysis of three homogeneity criteria for color image processing," in *IPM Workshop on Computer Vision*, Tehran, Iran, 2004.

[12] 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.

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

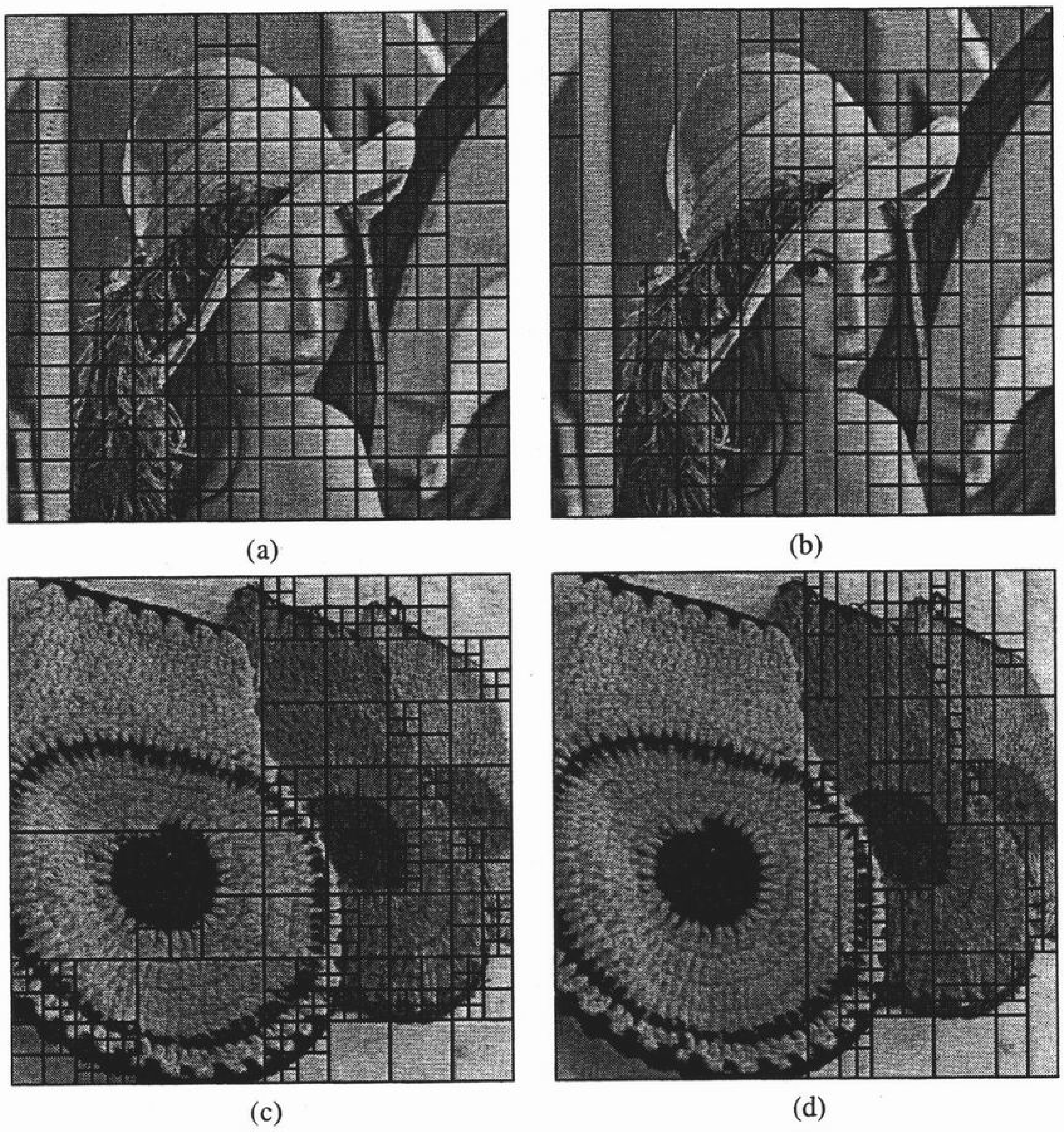

Fig. 6. Results of the quad-tree (left) and the proposed bi–tree (right) decomposition methods. (a),(b) $\varepsilon_1 = 10$, $\varrho = 4$. (c),(d) $\varepsilon_1 = 15$, $\varrho = 5$.