

# Improved fuzzy transform based image compression and fuzzy median filter based its artifact reduction: pairFuzzy

Deepak Gambhir<sup>1</sup> · Navin Rajpal<sup>1</sup>

Received: 28 November 2014 / Accepted: 2 May 2015 / Published online: 15 May 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** During image compression, visually significant edges should be well preserved for human perception. Despite existence of many image compression standards, joint photographic experts group (JPEG) is the most popularly used standard for image compression. However at low bit rate, JPEG compressed images exhibit blocking artifacts that adversely affect the visual image quality. Thus, to produce a high visual quality image at low bit rate, pairFuzzy algorithm that is simple and more efficient as compared to JPEG alongwith the capability of reducing artifact is proposed. The proposed algorithm is carried out in three steps. First, an image is preprocessed using competitive fuzzy edge detection which efficiently detects the edge pixels contained in the image. Second, based on the edge information the image is compressed and decompressed using improved fuzzy transform. Third, the reconstructed image is postprocessed using fuzzy switched median filter for artifact reduction. The subjective as well as objective analysis alongwith the comparison to recent methods proves the superiority of proposed algorithm.

**Keywords** Image compression/decompression · Edge detection · Fuzzy transform · Median filter · Artifact reduction

## 1 Introduction

Image compression reduces the amount of data required to represent a digital image. A large size of digital image requires large storage space, large bandwidth and more time for uploading and downloading on the internet. The problem can be solved by reducing the redundant and/or irrelevant information from the image. Image compression is widely used in medical imaging, remote sensing, video conferencing, high definition television, fax etc. Many image compression methods have been developed in literature. Among these commonly used compression techniques include discrete cosine transform (DCT) based such as joint photographic experts group (JPEG), wavelet based methods such as JPEG2000, fuzzy based, neural networks based, optimization techniques based image compression methods [3, 13, 21, 30, 38, 39].

*Motivation* JPEG based on DCT is the most popularly used image compression standard. However when low bit rate is to be achieved, JPEG produces compressed images that suffer from annoying blocking artifacts. To reduce the blocking artifacts onto the compressed images fuzzy transform is utilized by various researchers [25, 26]. Since fuzzy transform converts a complex problem into a respective problem of linear algebra that deals with vectors making computations easier and also possess an important property of preserving monotonicity [28] that can be utilized significantly to improve the quality of reconstructed image, hence in this paper we propose pairFuzzy algorithm based on improved fuzzy transform to compress the images.

PairFuzzy based image compression algorithm consists of three main steps:

---

✉ Deepak Gambhir  
gambhir.deepak@gmail.com  
Navin Rajpal  
navin\_rajpal@yahoo.com

<sup>1</sup> School of Information and Communication Technology,  
Guru Gobind Singh Inderprastha University, Dwarka,  
New Delhi, India

1. *Preprocessing using competitive fuzzy edge detection* In this step, input image blocks are classified into low-variation (LV) block, medium variation (MV) block or a high variation (HV) block using competitive fuzzy edge detector.
2. *Compression and decompression using improved fuzzy transform* The LV, MV and HV block are compressed and decompressed using improved F-transform.
3. *Postprocessing (Artifact reduction with fuzzy switched median filter)* The resultant decompressed image may contain some blocking artifacts, that are reduced using fuzzy switched median filter during postprocessing.

**Contribution** It is well known that image blocks with many edge pixels have more information and hence should be less compressed, whereas image blocks with smooth regions should be more compressed. Fuzzy transform based image compression method [7] compresses each image block to the same level without taking into the consideration whether containing the edge information or not. Whereas, the algorithm proposed in this paper called pairFuzzy algorithm compresses each image block taking into account the amount of edge information contained in it. The proposed approach gives better visual quality of compressed images with well-preserved edges and reduced artifacts. The proposed pairFuzzy algorithm provides substantial improvement in the visual quality of compressed image as compared with the state-of-art techniques [7, 27] as shown in Figs. 8, 9, 10, and 11 and their quantitative results are also reported in Tables 2, 3, 4 and 5 with well-defined parameters.

The rest of paper is organized as follows: Sect. 2 gives literature review of image compression and edge detection alongwith review of artifact reduction methods. Section 3 and its subsections explain the proposed method in detail. Section 4 provide results and discussions and finally the conclusions are drawn in Sect. 5.

## 2 Literature review

Image compression means reduction of irrelevant information with the aim to reduce space and thereby transmission time. These techniques may either be classified as either lossless or lossy [35]. In lossless compression techniques, an image can be compressed and decompressed without any loss of information. These techniques are widely used in medical imaging, computer aided design, video containing text etc. The standard formats using lossless compression techniques are portable network graphics (PNG), graphics interchange format (GIF), tagged image file format [TIFF (lossless)]. In lossy compression

techniques, an image is compressed by permanently discarding some redundant information and therefore the uncompressed image is not exactly the same as the original image, but is similar, in sense, that the change is not perceived by human eye. Lossy image compression techniques are used in video conferencing, mobile applications, and natural images over web etc. The standard file format using lossy image compression technique is JPEG, etc. Lossy compression techniques can provide high compression ratio as compared to lossless compression techniques. Lossy image compression techniques generally introduce artifacts when the image is compressed at low bit rate [38, 39].

According to human visual system for depth perception of an image [34], edges play an important role in image processing and computer vision. Edges contain important structural information [5, 17] and their detection forms a basis for representing an image from pixel level to feature level. Literature shows various works on edge detection based image compression. Horn et al. [6] proposed an edge based compression scheme by extracting edge and mean information for very low bit rate applications. Avramovic [2] presents a lossless image compression algorithm that uses edge information for eliminating spatial redundancy. This algorithm is based on gradient edge detection predictor that strongly depends on prediction, entropy coding and context modeling. Markus et al. [23, 24] proposed edge detection based image compression algorithm for cartoon-like images. This algorithm exploits the information that is present on both sides of edges extracted using Marr-Hildreth operator for compressing these images. In past few years, many edge detection image compression algorithm using fuzzy logic have been developed that are more robust and flexible compared to the classical approaches. Amarunnishad et al. [1] proposed a fuzzy based block truncation coding algorithm using the Yager involutive fuzzy complement edge operator (YIFCEO). The method uses fuzzy logical bit block (LBB) for encoding the input image along with block mean and the standard deviation. Gambhir et al. [9] proposed image compression algorithm that detects edge pixels based on entropy optimized fuzzy edge detector and uses adaptive quantization based compression and decompression. Petrosino et al. [29] proposed an image compression algorithm based on the rough fuzzy sets that depend on fuzzy vector quantization (FVQ) to determine optimal code vectors. The decoder proposed uses specific properties of rough fuzzy sets to reconstruct the original image blocks.

However, compressing an image at low bit rate introduces blocking artifact that seriously degrades the visual quality of the image [15, 31]. Gambhir et al. [10] proposed artifact reduction of DCT compressed images using the values obtained from fuzzy s-shaped membership function that obtains the edge class of pixel from the Gaussian fuzzy

edge detector. Zou et. al [41] proposed a projection onto convex sets (POCS) based blocking artifact reduction algorithm by considering the intensity variation across and along the edges. The algorithm uses two convex sets, a smoothness constraint set and a narrow quantization constraint set. Luo et al. [22] proposed a method of reducing blocking artifact of block based DCT compressed images which performs artifact reduction in transform domain as well as in spatial domain. The method takes advantage of the continuity provided by original pixels belonging to the same block and also the correlation between the neighboring blocks to reduce the pixel discontinuity along the boundaries. Kim et al. [14] proposed a signal adaptive weighted sum (SAWS) technique to reduce the blocking artifact obtained in highly compressed images. The technique is based on adaptively adjusting the weights based on correlation as well as on the activities across the boundaries. Gambhir et al. [11] proposed dualFuzzy algorithm that classifies each input image block into either smooth block or edge block using histogram based fuzzy edge classifier. At the encoder, smooth image blocks are encoded to its mean value and the edge block are compressed using fuzzy transform. At the decoder, the reconstructed image obtained using the mean value and the decompression using inverse fuzzy transform, contains artifact that are reduced using Gaussian block based enhancement scheme, whereas this paper classifies each input image block into either LV, MV and HV using competitive fuzzy edge detector, compresses and decompresses each classified block using improved fuzzy transform and uses fuzzy switched median filter for reducing blocking artifacts.

### 3 Proposed method

The proposed pairFuzzy method follow three steps: (1) preprocessing based on competitive fuzzy edge detection, (2) compression and decompression using improved fuzzy transform, and (3) postprocessing using fuzzy switched median filter based artifact reduction. Figure 1 shows the block diagram for proposed pairFuzzy method and the subsections below describe each step in detail:

#### 3.1 Preprocessing using competitive fuzzy edge detection (CFED)

In this step, CFED assigns each input pixel to one of the six classes depending on its neighborhood. These classes are then classified into LV, MV and HV blocks depending on the number of edge pixels contained in them.

CFED can speedily detect edge pixels effectively because it uses different fuzzy membership functions based on neighborhood situation for different directions and

hence is used for segmenting the blocks. CFED accepts input either in the form of feature vectors based on fuzzy set membership functions or vectors belonging to various classes based upon fuzzy memberships. Each input vector is assigned a class depending upon the max/min fuzzy membership function. The competitive fuzzy edge detector proposed by Liang and Looney [16] first detect the edge

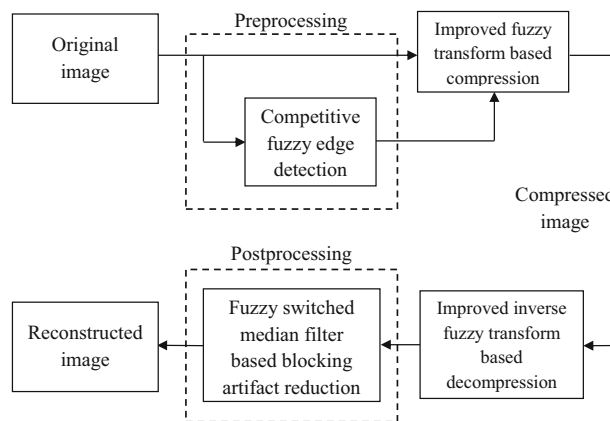


Fig. 1 Proposed method

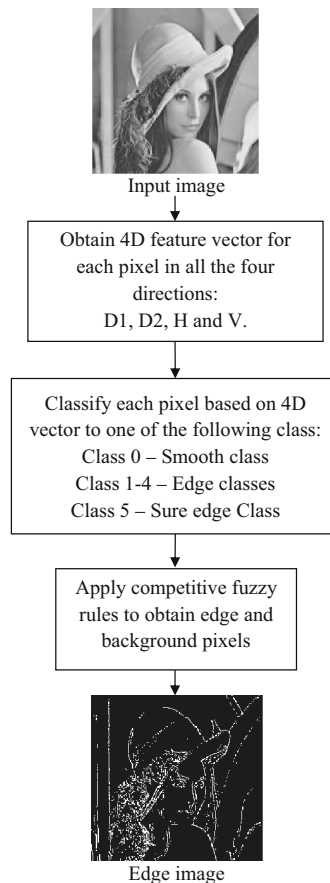


Fig. 2 Competitive fuzzy edge detection (CFED)

pixels and then apply competitive rules for thinning the edges. Figure 2 shows the flow diagram for competitive fuzzy edge detection.

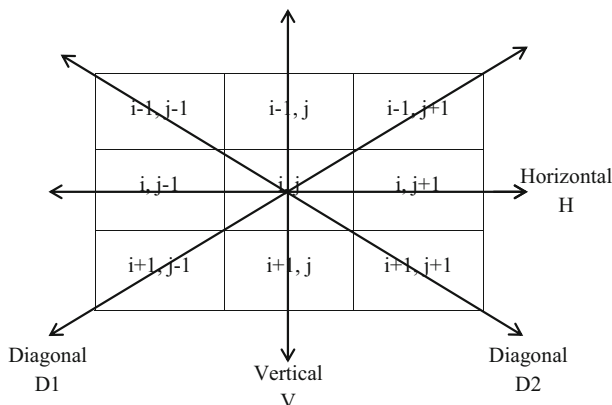
### 3.1.1 4D Feature vector

A four-dimensional (4D) feature vector defined such as  $V = \{v_1, v_2, v_3, v_4\}$  and is computed for each pixel in an input image, by summing the gray level magnitude difference of the pixels in the four directions (shown in Fig. 3) on  $3 \times 3$  neighborhood about the center pixel. The bidirectional sum magnitude differences ( $v_1, v_2, v_3, v_4$ ) along the four directions (diagonal  $D_1$ , diagonal  $D_2$ , horizontal  $H$  and vertical  $V$ ) for a pixel  $x(i, j)$  are obtained respectively as:

$$\begin{aligned} v_1 &= |x(i, j) - x(i + 1, j - 1)| + |x(i, j) - x(i - 1, j + 1)| \\ v_2 &= |x(i, j) - x(i - 1, j - 1)| + |x(i, j) - x(i + 1, j + 1)| \\ v_3 &= |x(i, j) - x(i, j - 1)| + |x(i, j) - x(i, j + 1)| \\ v_4 &= |x(i, j) - x(i - 1, j)| + |x(i, j) - x(i + 1, j)| \end{aligned} \quad (1)$$

### 3.1.2 Edge classification

The competitive fuzzy classifier distinguishes pixels into six classes: class 0 (smooth class), class 1–4 (edge classes) and class 5 (sure edge class) depending upon its neighborhood situations. The smooth class is assigned to a pixel that has low magnitude intensity differences along all the four directions in its neighborhood. A sure edge class is assigned to a pixel that has high magnitude of pixel intensity differences along all the directions in its neighborhood. Class 1 is assigned to a pixel that has low magnitude intensity difference along diagonal  $D_2$  and high magnitude differences along remaining directions. Class 2 is assigned to a pixel that has low magnitude intensity difference along



**Fig. 3** Edge directions for a pixel at  $(i, j)$

vertical  $V$  and high magnitude differences along remaining directions. Class 3 is assigned to a pixel that has low magnitude intensity difference along diagonal  $D_1$  and high magnitude differences along remaining directions. Class 4 is assigned to a pixel that has low magnitude intensity difference along horizontal  $H$  and high magnitude differences along remaining directions. Pixels with feature vector not satisfying any of the above six conditions are assigned to one of the edge class for further processing. Once a feature vector for each pixel has been determined six-4D prototype vectors  $C_i$ ,  $i = 0$  to 5, are constructed to represent the respective center of six classes i.e. a smooth class, four-edge classes and a sure edge class. These centers are assigned a value ‘low’ to represent low sum magnitude difference and a value ‘high’ to represent high sum magnitude difference in the given direction.

Competitive rules are then applied to each edge pixel according to class assigned to it. Only the pixels classified as edge pixels and sure edge pixels are assigned the value high (representing white pixels) in the output image and all the other pixels are assigned low in the output image (representing black pixels). This result in white edges on black background. The fuzzy set membership functions for six-classes based on 4D feature vector ( $V$ ) and 4D prototype vector ( $C$ ) are obtained using extended Epanechnikov function as:

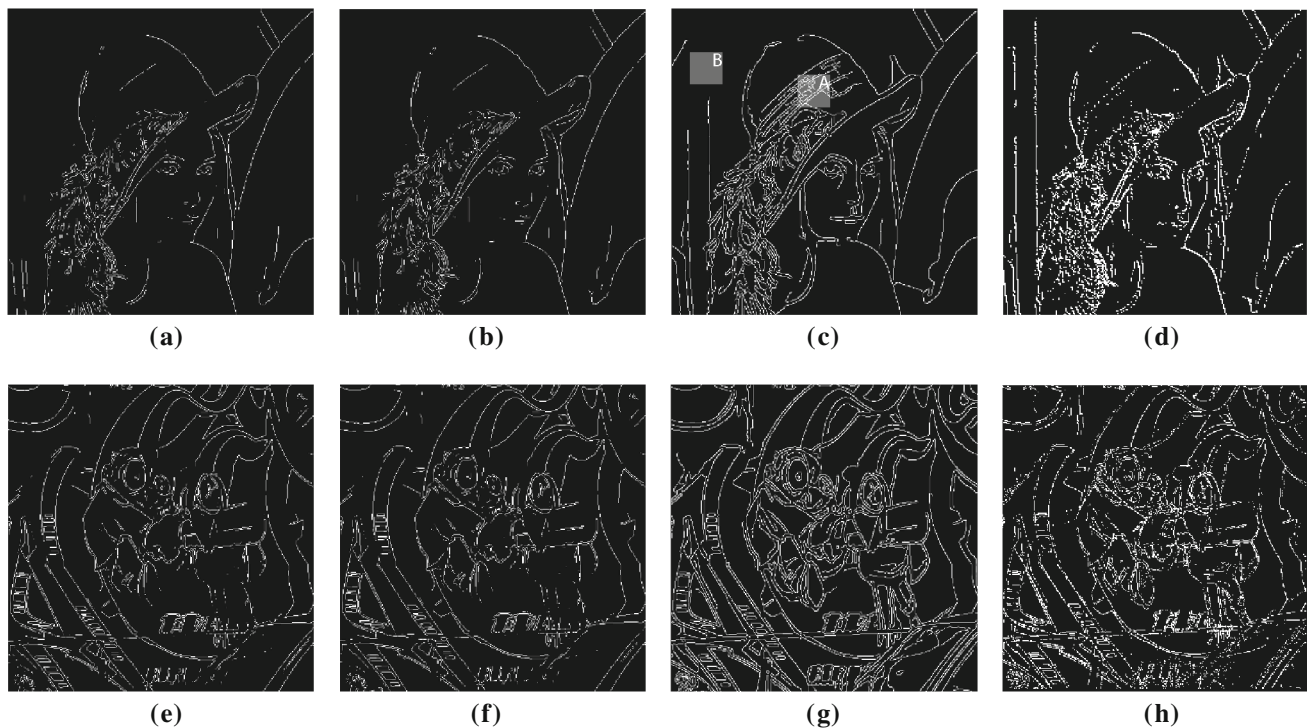
$$\mu_i(v) = \text{Max} \left\{ 0, \left( 1 - \frac{|v - c_i|^2}{W^2} \right) \right\} \quad (2)$$

where  $i = 0$  represents the smooth class,  $i = 1, 2, 3, 4$  represents edge class,  $i = 5$  represent the sure edge class, and  $W$  defines the width of these fuzzy set membership functions. Membership functions based on uniform, triangular, Epanechnikov kernels can be used, however, Epanechnikov function is optimal in mean square error sense and hence is preferred [20].

### 3.1.3 Fuzzy rule firing and defuzzification

Before classifying an edge pixel to either white or black, it is compared to neighborhood edge pixels. After classifying edge pixel to an edge class, it is compared with the pixel on either side of it and the pixel with the largest absolute difference is stored as white (edge) pixel and the others are saved as black (smooth) pixel thereby thinning the edges. The rules for this competition are given below:

1. Rule 1: If  $x$  belongs to smooth class (class 0), then change pixel to black.
2. Rule 2: If  $x$  belongs to edge class 1, compare  $v_1$  with its neighboring pixels along the diagonal  $D_2$ . If  $v_1$  is large then change pixel value to white else black.



**Fig. 4** Results of various edge detection algorithms for Lenna and Graffiti. **a** Sobel edge, **b** Prewitt edge, **c** Canny edge, **d** CFED, **e** Sobel edge, **f** Prewitt edge, **g** Canny edge, **h** CFED

3. Rule 3: If  $x$  belongs to edge class 2, compare  $v_3$  with its neighboring pixels along the horizontal  $H$ . If it is large then change pixel to white else black.
4. Rule 4: If  $x$  belongs to edge class 3, compare  $v_2$  with its neighboring pixels along the diagonal  $D_1$ . If it is large then change pixel to white else black.
5. Rule 5: If  $x$  belongs to edge class 4, compare  $v_4$  with its neighboring pixels along the vertical  $V$ . If it is large then change pixel to white else black.
6. Rule 6: If  $x$  belong to sure edge class (class 5), then change pixel to white.

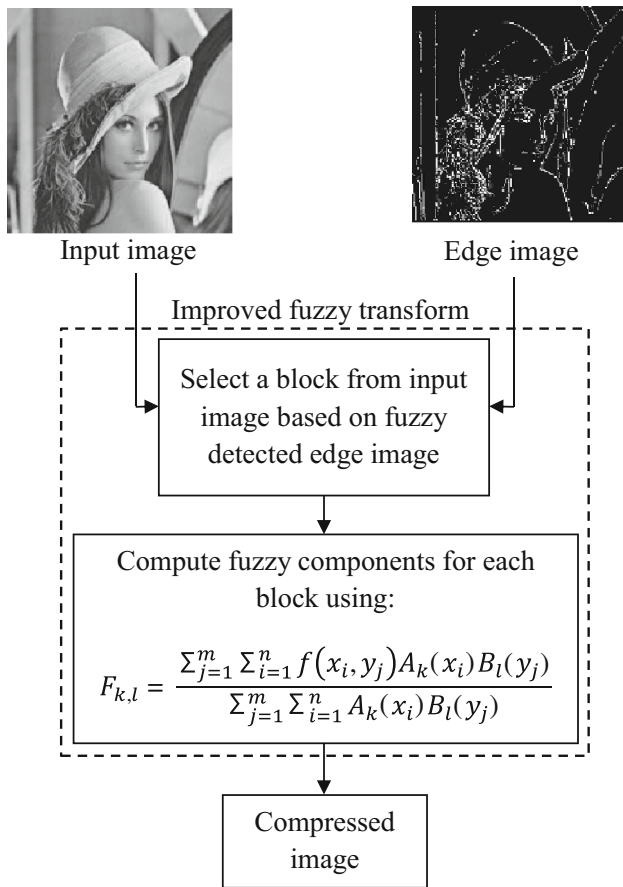
The competitive fuzzy edge detection method classifies the pixel either as a edge class or as a smooth class using the competitive rules and yielded white lines on black background which are neither thick nor noisy as yielded by threshold based Prewitt or Sobel operator. The method yielded thin black lines even for the diffused edge as well as even when the required parameters are adjusted roughly by the user (magnitude intensity difference nearly equal to 3–10 is assigned low and between 30 to 40 is assigned high [16]). The use of six fuzzy set membership functions help in the speedy detection of edges in all the directions. The parameters of competitive fuzzy edge detector for producing desirable results could be selected in very less number of trials as compared to Canny edge detector [4] that required many number of trials with different combination of parameters before arriving at final set of

parameters. Also Canny edge detector sometimes connect lines into closed contour thereby misinterpreting the edges whereas competitive fuzzy edge detector is free from this drawback. The results of various edge detection algorithms for Lenna and Graffiti are shown in Fig. 4.

The competitive fuzzy edge detector can efficiently detect edges in the image block. These image blocks are then classified into LV, MV and HV block depending upon number of edge pixels contained in them. On extensively experimenting CFED on different standard images and real time images, it is concluded that treating blocks with less than 20 % of edge pixels as LV block, between 20 and 70 % of edge pixels as MV block and rest as HV block yields an optimum quality of compressed image at low bit rate.

### 3.2 Compression and decompression using improved fuzzy transform

Fuzzy transform based image compression and decompression have been proposed in [7, 8] and compared with fuzzy relation equations (FEQ) and JPEG. Fuzzy transform gives better results than FEQ and but results obtained using Fuzzy transform are similar or slightly less than JPEG based compression. Thus an attempt is made to provide an algorithm that performs better than JPEG and fuzzy transform. In the proposed algorithm, the improved fuzzy



**Fig. 5** Proposed fuzzy edge detection based improved fuzzy transform compression

transform treats each of the LV, MV and HV blocks differently for compression. These blocks are compressed to different size blocks thereby maintaining an average value of compression rate and performing better than JPEG at similar compression rates. Figure 5 shows proposed fuzzy edge detection based improved fuzzy transform compression.

### 3.2.1 Fuzzy transform

A fuzzy transform converts a continuous function on the closed interval  $[a, b]$  into a set of  $N$ -dimensional vector. The inverse fuzzy transform converts this  $N$ -dimensional vector into a continuous function which approximates the original function up to a small quantity  $\epsilon$ . Fuzzy transform has an advantage of producing a simple and unique representation of an original function which if used in place of original function makes complex computations easier.

Fuzzy partition of the universe: Consider a closed interval  $[a, b]$  as a universe consisting of  $n$  fixed nodes ( $n \geq 2$ )  $x_1 \leq x_2 \leq x_3 \dots \leq x_n$ , such that  $x_1 = a$  and  $x_n = b$ . The fuzzy sets  $[A_1, A_2, \dots, A_n]$  identified with their membership functions  $[A_1(x), A_2(x), \dots, A_n(x)]$  defined on  $[a, b]$

forms the fuzzy partition of the universe, if the following conditions are satisfied for  $k = 1, 2, 3, \dots, n$ .

1.  $A_k : [a, b] \rightarrow [0, 1]$ , and  $A_k(x_k) = 1$ .
2.  $A_k(x) = 0$  if  $x \notin (x_{k-1}, x_{k+1})$ .
3.  $A_k(x)$  is a continuous function on the domain  $[a, b]$ .
4.  $A_k(x)$  monotonically increases on  $[x_{k-1}, x_k]$  and monotonically decreases on  $[x_k, x_{k+1}]$ .
5. For all  $x$ ,  $\sum_{k=1}^n A_k(x) = 1$ . If the set of points  $x_1, x_2, x_3, \dots, x_n$  for  $n \geq 3$  are equidistant, i.e.  $x_k = x_1 + \Delta(k - 1)$ , where  $\Delta = \frac{(x_n - x_1)}{(n - 1)}$  for  $n \geq 2$  then  $[A_1, A_2, \dots, A_n]$  forms a uniform fuzzy partition, if the following two additional conditions are satisfied.
  - (a)  $A_k(x_k - x) = A_k(x_k + x)$ , for all  $x$  and  $k = 2, \dots, n - 1$  ( $n \geq 2$ ).
  - (b)  $A_{k+1}(x) = A_k(x - \Delta)$ , for all  $x$  and  $k = 2, \dots, n - 1$  ( $n \geq 2$ ).

For a uniform fuzzy partition,  $\Delta$  is the length of support of  $A_1(x)$  and  $A_n(x)$  whereas,  $2\Delta$  is the support of other fuzzy functions,  $A_k(x)$  for  $k = 2, \dots, n - 1$ .

### 3.2.2 Discrete fuzzy transform and its inverse for one variable

Let the original function  $f(x)$ , be known only at some nodes,  $x_1, x_2, x_3, \dots, x_N \in [a, b]$  which are sufficiently dense with respect to fixed partition i.e. for every  $i = 1, 2, 3, \dots, N$ , there exist an index  $j \in \{1, 2, \dots, n\}$  such that  $A_i(x_j) \geq 0$ . Then the  $n$ -tuple  $F = [F_1, F_2, F_3, \dots, F_N]$  is defined as the discrete fuzzy transform of  $f(x)$  with respect to  $[A_1, A_2, A_3, \dots, A_N]$ , if

$$F_i = \frac{\sum_{j=1}^n f(x_j) A_i(x_j)}{\sum_{j=1}^n A_i(x_j)} \tag{3}$$

for  $i = 1, 2, 3, \dots, N$ .

It is clear from Eq. 3 that the components of the discrete fuzzy transform i.e.  $F_i$  are the weighted mean values of the given function, where the weights are given by membership functions.

The inverse discrete fuzzy transform  $f_{FN}(x_j)$  of  $F$  with respect to  $\{A_1, A_2, A_3, \dots, A_N\}$  such that it approximately reconstructs the original function  $f(x)$  and is defined as:

$$f_{FN}(x_j) = \sum_{i=1}^N F_i A_i(x_j) \tag{4}$$

The inverse discrete fuzzy transform  $f_{FN}(x)$  can approximate the original function,  $f(x)$  over a set of points  $x = \{x_1, x_2, x_3, \dots, x_N\} \in [a, b]$  if,

$$|f(x_j) - f_{FN}(x_j)| < \epsilon \tag{5}$$

for  $j = 1, 2, 3 \dots N$ , where  $\epsilon$  is very small positive quantity and  $0 < \epsilon < 1$ .

### 3.2.3 Discrete fuzzy transform and its inverse for two variables

Consider a two dimensional (2D) function,  $f(x, y)$  on closed interval  $[a, b] \times [c, d]$  as a universe consisting of  $(N + M)$  fixed nodes (where  $N, M \geq 2$ ),  $x_1, x_2, x_3, \dots, x_N$ , such that  $x_1 = a$  and  $x_N = b$ , be the  $N$  number of fixed nodes on  $[a, b]$  and  $y_1, y_2, y_3, \dots, y_M$  such that  $y_1 = c$  and  $y_M = d$  be the  $M$  number of fixed nodes on  $[c, d]$ . Let  $[A_1, A_2, A_3, \dots, A_N]$  and  $[B_1, B_2, B_3, \dots, B_M]$  be the fuzzy partition of  $[a, b]$  and  $[c, d]$  respectively and  $[A_1(x), A_2(x), \dots, A_N(x)]$  and  $[B_1(y), B_2(y), \dots, B_M(y)]$  be the membership functions of the related fuzzy partition such that  $A_i(x) > 0$  for  $[i = 1, 2, 3, \dots, N]$  and  $B_j(y) > 0$  for  $[j = 1, 2, 3, \dots, M]$ . Moreover these nodes are sufficiently dense with respect to the chosen partitions i.e for  $i = 1, 2, 3 \dots N$  there exists an index  $k = 1, 2, 3 \dots n$  such that  $A_i(x_k) > 0$  and for each  $j = 1, 2, 3 \dots M$  there exists an index  $l = 1, 2, 3 \dots m$  such that  $B_j(y_l) > 0$ . The 2D discrete fuzzy transform of the function  $f(x, y)$ , with respect to  $[A_1, A_2, \dots, A_N]$  and  $[B_1, B_2, \dots, B_M]$  is defined as:

$$F_{k,l} = \frac{\sum_{j=1}^m \sum_{i=1}^n f(x_i, y_j) A_k(x_i) B_l(y_j)}{\sum_{j=1}^m \sum_{i=1}^n A_k(x_i) B_l(y_j)} \tag{6}$$

for  $k = 1, 2, 3, \dots, N$  and  $l = 1, 2, 3, \dots, M$ . And the discrete inverse discrete fuzzy transform of  $F$  with respect to  $\{A_1, A_2, \dots, A_N\}$  and  $\{B_1, B_2, \dots, B_M\}$  is defined as:

$$f_{FN}(x_i, y_j) = \sum_{k=1}^N \sum_{l=1}^M F_{k,l} A_k(x_i) B_l(y_j) \tag{7}$$

for  $i = 1, 2, 3 \dots n$  and  $j = 1, 2, 3 \dots m$ .

The inverse discrete fuzzy transform  $f_{FN}(x, y)$  can be approximates the original function,  $f(x, y)$  over the set of points  $x = \{x_1, x_2, x_3 \dots x_N\}$  and  $y = \{y_1, y_2, y_3, \dots, y_M\}$  which are used to be made sufficiently dense with respect to  $\{A_1, A_2, A_3, \dots, A_N\}$  and  $\{B_1, B_2, B_3, \dots, B_M\}$ , if

$$|f(x_i, y_j) - f_{FN}(x_i, y_j)| < \epsilon \tag{8}$$

for  $i = 1, 2, 3, \dots, n$  and  $j = 1, 2, 3, \dots, m$ . Perfilieva and Martino [7, 27] proposes, a method of lossy image compression and its reconstruction based on discrete fuzzy transform.

### 3.3 Postprocessing [artifact reduction with fuzzy switched median filter (ARFSM)]

The reconstructed image obtained by applying improved inverse fuzzy transform on compressed image may contain

artifacts at the boundary of the blocks. Liu et al. [18] proposed a DCT-domain algorithm for blind measurement of blocking artifacts and reduced them by using a DCT based post filtering method. Kim et al. [14] proposed signal adaptive weighted sum filter for reducing blocking artifacts, these were reduced by assigning weights higher to the pixels that are nearer to the boundary than those that are away from the boundary. These weights are assigned according to the correlation and activity in the neighborhood. Jagroop et al. [31] reduced the artifacts in block based DCT compressed images using signal adaptive filter. Researches have considered pixels at the boundary of the block as the noisy pixels and reduces them using filtering action. Thus this paper proposes fuzzy switched median filter for reducing the blocking artifacts. Since fuzzy switched median filter is more robust than the mean filter and also it does not create unrealistic pixels value when it processes an edge. For this reason, median filter is preferred for reducing the artifact and preserving edges.

Artifacts can be successfully reduced by using a square filtering window  $W_{(2M+1) \times (2M+1)}$  ( $M$ : an even integer) with  $(2M + 1)$  rows and  $(2M + 1)$  columns centered at pixel  $x(i, j)$ , positioned at  $i, j$ . The filtering operation is required only at the boundary of the adjacent blocks and the pixel under artifact is replaced by a corrected pixel value obtained using fuzzy switched median filter [32, 33]. When the window is placed on the right (left) boundary of the image, then the neighboring pixels on its left (right) side are assumed to be free from any artifact. The value of the corrected pixel is calculated based on the median value of pixels under the square filtering window ( $W$ ).

$$Med(i, j) = Median\{x(i + k, j + l)\} \tag{9}$$

for  $k, l = -M$  to  $+M$ .

After computing the median value, the fuzzy input variable  $D(i, j)$  that provides local information in  $(2M + 1) \times (2M + 1)$  window is calculated as maximum of absolute difference between intensities of pixel  $i, j$  with its neighboring pixels.

$$D(i, j) = Max\{|x(i + k, j + l) - x(i, j)|\} \tag{10}$$

where  $x_{i+k,j+l} \neq x_{i,j}$ .

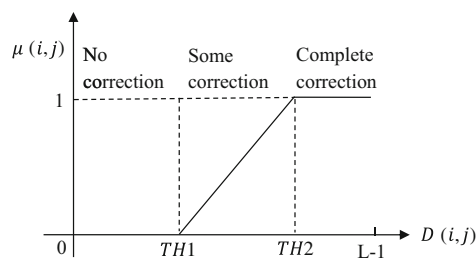


Fig. 6 Fuzzy membership function

The switched median filter uses the fuzzy variable  $D(i,j)$  for reducing the blocking artifacts from the decompressed image, by defining the fuzzy membership function  $\mu(i,j)$  (shown in Fig. 6) as:

1. Pixels with value of  $D(i,j)$  between 0 and  $TH1$  are most likely to be non-edge pixels and hence treated as noise free pixels with zero membership value and requires no correction for reducing artifacts.
2. Pixels with value of  $D(i,j)$  between  $TH1$  and  $TH2$  are most likely to be edge pixels and hence treated as noisy pixels with membership value between 0 and 1 and requires some correction on the basis of their membership for reducing artifacts.
3. Pixels with value of  $D(i,j)$  greater than  $TH2$  are sure edge pixels and hence has membership value equal to 1 and needs to be corrected for reducing artifact.

$$\mu(i,j) = \begin{cases} 0 & \text{for } D(i,j) \leq TH1 \\ mD(i,j) + c & TH1 < D(i,j) < TH2 \\ 1 & D(i,j) \geq TH2 \end{cases} \quad (11)$$

Where  $m$  is

$$m = \frac{1}{(TH2 - TH1)}$$

and  $c$  is

$$c = \frac{(-TH1)}{(TH2 - TH1)}$$

Finally, artifact reduced resultant image is obtained by replacing the current pixel  $x(i,j)$  with a corrected pixel  $y(i,j)$ , defined as:

$$y(i,j) = [(1 - \mu(i,j))]x(i,j) + \mu(i,j)Med(i,j) \quad (12)$$

The corrected pixel is a linear fuzzy weighted combination of the original pixel  $x(i,j)$  and the median value  $Med(i,j)$  for the block. Extensive experiments show that ARFSM is successful in producing a high quality of compressed image with very negligible artifact if the two thresholds  $TH1$  and  $TH2$  are in the range of  $5 < TH1 < 15$  and  $25 < TH2 < 35$  for a gray scale image having number of levels  $L = 256$  and these value of thresholds agrees with those stated in [32]. Figure 7 shows flow diagram for the proposed fuzzy switched median filter based artifact reduction technique.

The proposed work is summarized as: The pairFuzzy algorithm preprocesses an image using CFED and then compresses it using improved fuzzy transform. The

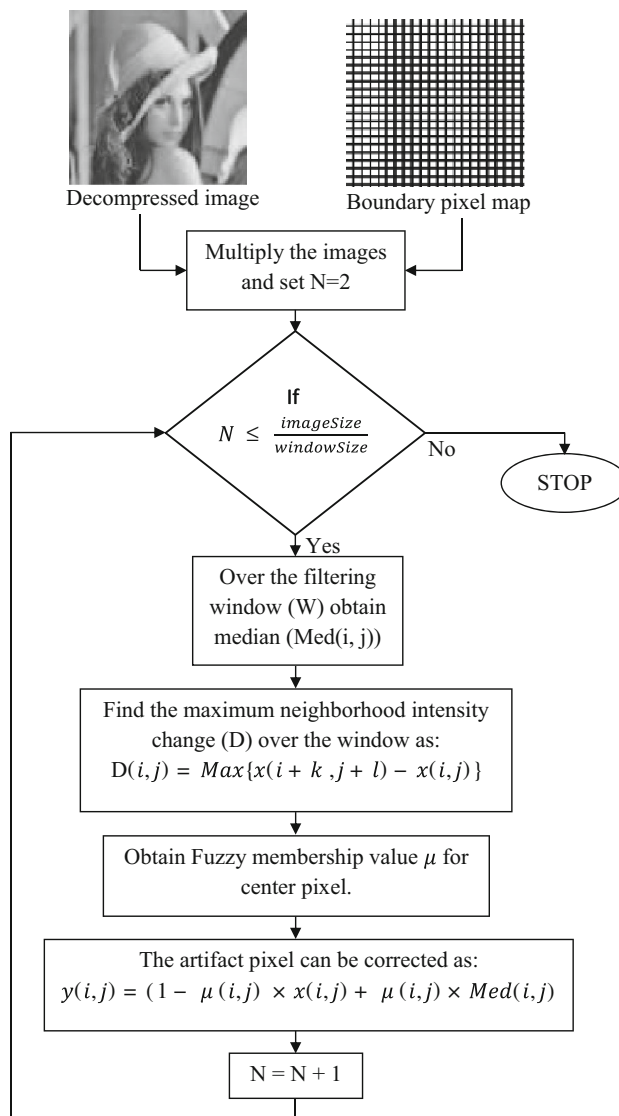


Fig. 7 Proposed fuzzy switched median filter based artifact reduction algorithm

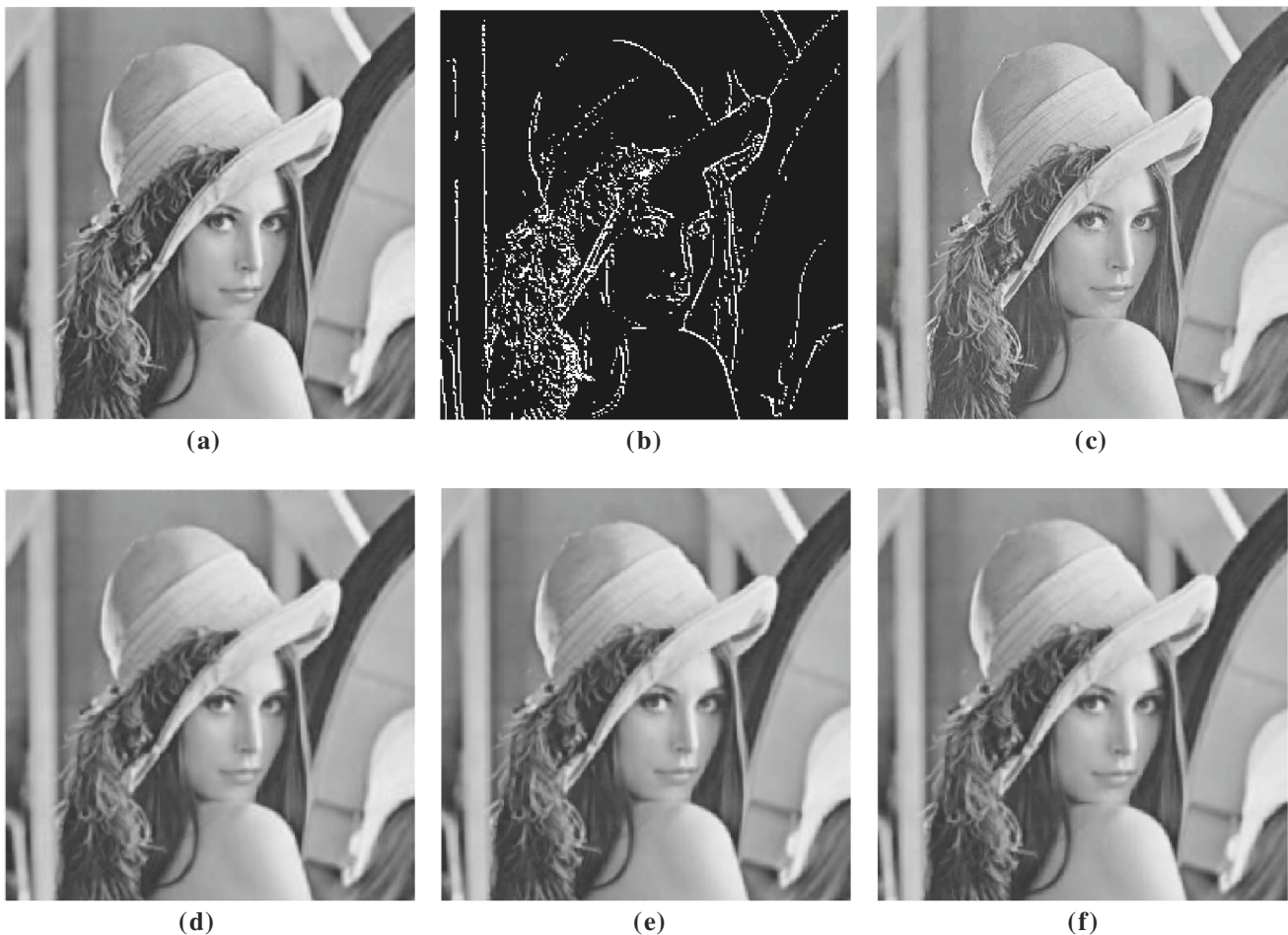
sensitivity of the proposed algorithm lies in the edge detection algorithm. If too many edge pixels are detected then the images are compressed less by the proposed algorithm and achieves low compression ratio whereas, too less detected edge pixels results in heavy compression of image, achieving high compression ratio but loss of information. Canny edge detector with its best sensitivity [16] for Lena image is achieved with Threshold ( $T$ ) = 0.2 and standard deviation ( $\sigma$ ) = 0.5 whereas, competitive fuzzy edge detector with low = 0, high = 20 and width ( $W$ ) = 256.



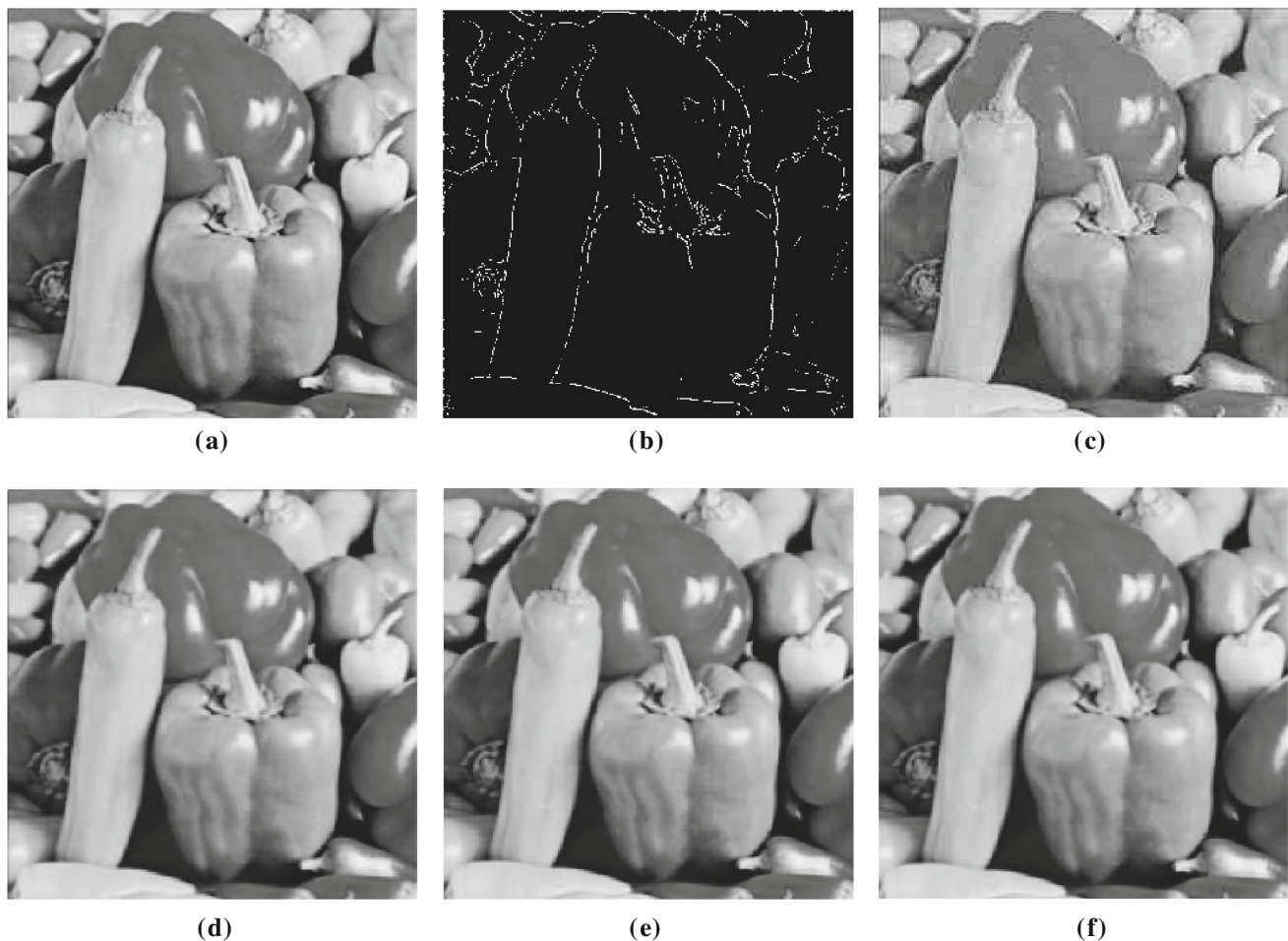
**Table 1** No. of LV, MV and HV blocks for different images

Test images	256 × 256			512 × 512		
	LV	MV	HV	LV	MV	HV
Lenna	994	27	3	4072	14	10
Goldhill	963	54	7	3943	141	12
Peppers	1006	17	1	4017	64	15
Camerman	962	50	12	3838	242	16
House	1001	18	5	4075	20	1
Jetplane	945	70	9	3751	273	72
Woman blonde	993	25	6	3762	261	73
Living room	931	89	4	3672	363	61
Graffiti	857	153	14	3921	173	2
Wall	938	75	11	3414	669	13
Leuven	957	64	3	3801	261	34
Boat	694	265	65	3335	684	77
Trees	671	293	60	3028	1040	28
Bikes	987	34	3	4004	89	3
Bark	1006	15	3	4060	33	3

However Canny edge detector detects some spurious edges (shown in Fig. 4c, region A) and even misses some of the edges (shown in Fig. 4c, region B) that affect the performance of the proposed algorithm in terms of image quality and compression ratio. The fuzzy transform based image compression proposed by Perfilieva and Martino [7, 27] compressed all blocks of an image into same dimensions, whereas the authors improve fuzzy transform by reducing all blocks of an image into different dimensions depending upon the number of edges contained in them. The proposed pairFuzzy uses the idea such that each block of the image is to be compressed is classified to either low variation (LV) block or medium variation (MV) block or high variation (HV) block based upon competitive fuzzy edge detection. A block with large number of edge pixels is termed as HV, with very small number of edge pixels is termed as LV and remaining as MV. These blocks are the compressed using improved fuzzy transform. Since HV block contains more edge information hence it is compressed less as compared to LV block. For example: a  $8 \times 8$  LV block is compressed



**Fig. 8** Lenna test image. **a** Lenna original, **b** edge image, **c** JPEG, **d** fuzzy transform, **e** fuzzy transform with ARFSM, **f** pairFuzzy



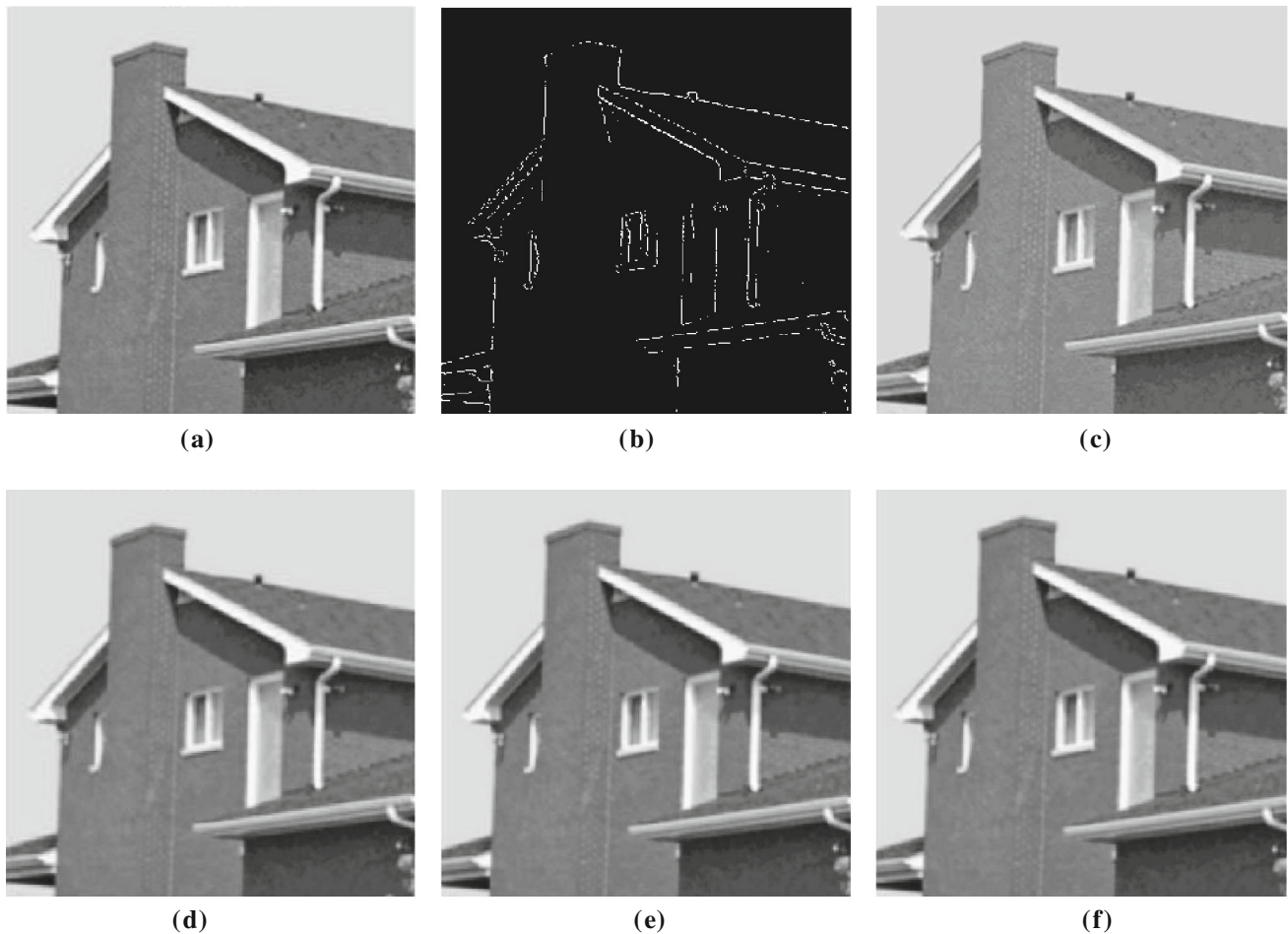
**Fig. 9** Peppers test image. **a** Peppers original, **b** edge image, **c** JPEG, **d** fuzzy transform, **e** fuzzy transform with ARFSM, **f** pairFuzzy

to  $3 \times 3$  block,  $8 \times 8$  MV block is compressed to  $5 \times 5$  block and  $8 \times 8$  HV block is compressed to  $7 \times 7$  block. The improved inverse fuzzy transform of these compressed blocks then results in the compressed image. The compressed image contain blocking artifacts. To reduce these artifacts, the proposed pairFuzzy algorithm uses fuzzy switched median filter scheme. This scheme uses the property that pixels at the block boundary are uncorrelated with pixels around the center of the blocks, hence treated as the noisy pixels and improvement of these pixels is achieved using linear combination of fuzzy weighted pixel value and its median value.

#### 4 Results and discussions

To evaluate the performance of the proposed pairFuzzy algorithm, it has been applied to a variety of test images of size  $256 \times 256$  as well as to images of size  $512 \times 512$ . Test

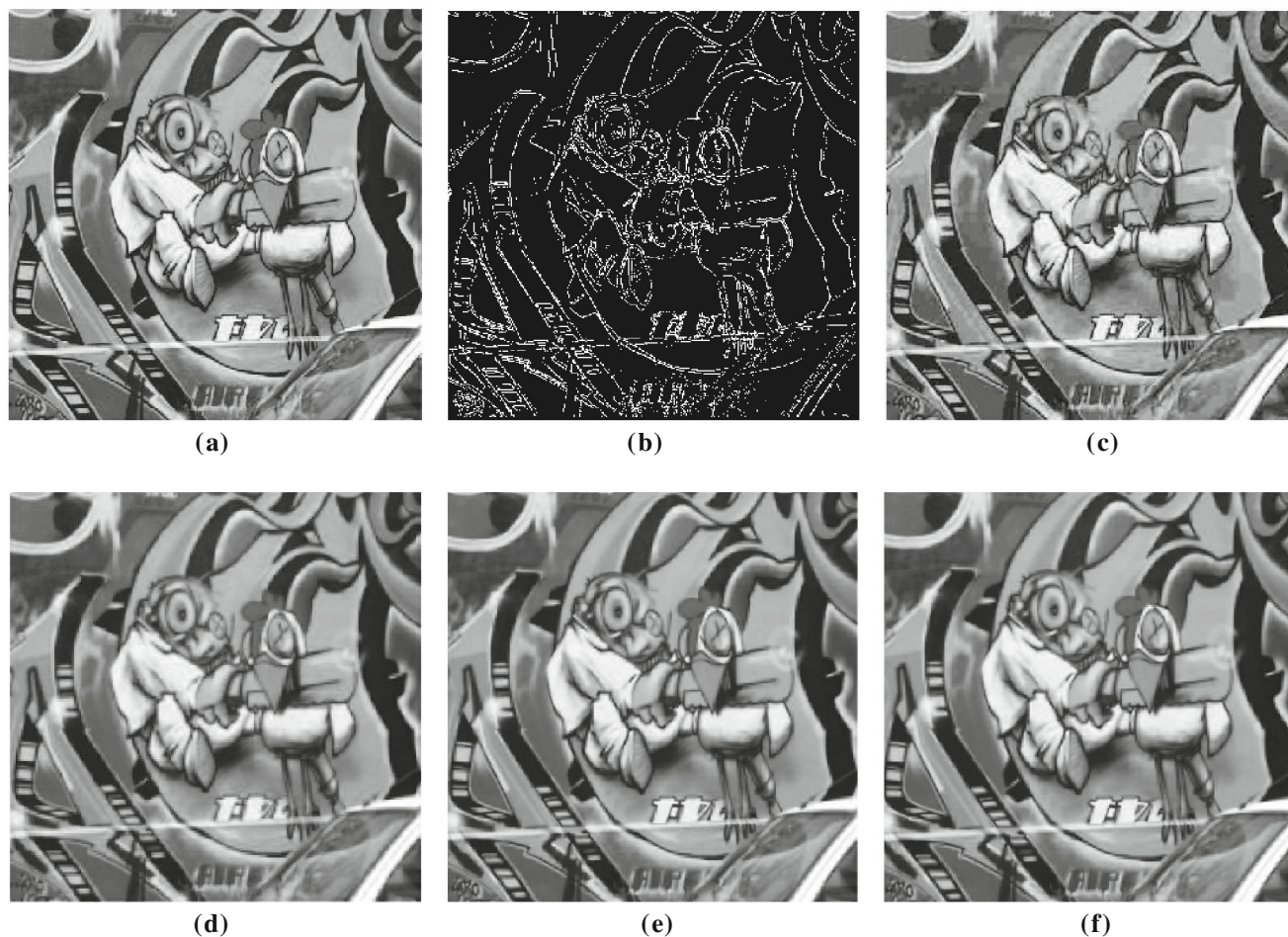
images on which experiment is performed include general test images and images taken from Oxford University, UK [12]. For compression, the images are initially divided into blocks, the total number of  $8 \times 8$  size blocks for  $256 \times 256$  size image is  $\frac{(256 \times 256)}{(8 \times 8)} = 1024$  and for  $512 \times 512$  size image is  $\frac{(512 \times 512)}{(8 \times 8)} = 4096$ . Table 1 shows the number of low variation (LV) blocks, medium variation (MV) blocks and high variation (HV) blocks for both size of images. Experimentally it has been found that block containing less than 20 % of edge pixels should be treated as LV block, between 20 and 70 % of edge pixels as MV block and rest as HV block. Fuzzy switched median filter for artifact reduction has also been applied to reduce the artifact present in compressed images obtained using fuzzy transform based image compression. The proposed algorithm is also analyzed using both objective and subjective measures and the results obtained are compared with those obtained using JPEG, fuzzy transform and fuzzy transform with



**Fig. 10** House test image. **a** House original, **b** Edge image, **c** JPEG, **d** Fuzzy transform, **e** Fuzzy transform with ARFSM, **f** pairFuzzy

ARFSM. Figures 8, 9, 10, and 11 shows the subjective comparison of the above said algorithms on the various test images. These figures show the original image (Figs. 8, 9, 10, 11a), edge image obtained using competitive fuzzy edge detector (Figs. 8, 9, 10, 11b), compressed images obtained using JPEG (Figs. 8, 9, 10, 11c), fuzzy transform (Figs. 8, 9, 10, 11 d), fuzzy transform with ARFSM (Figs. 8, 9, 10, 11e) and pairFuzzy (Figs. 8, 9, 10, 11f) image compression methods. It is observed that the proposed algorithm provides better visual quality than the other methods. JPEG compressed images suffers from huge blocking artifacts that degrades the visual image quality. Fuzzy transform compressed images also exhibit artifacts but smaller than those contained in the JPEG compressed images. ARFSM helps in detection and reduction of these

artifacts hence improving the performance of fuzzy transform with ARFSM. Pairfuzzy produces most pleasing results that contains significantly reduced artifacts while simultaneously preserving small details and sharp discontinuities. Figure 12 shows the zoomed upper left corner visual results of JPEG, fuzzy transform and pairFuzzy compression methods for Lenna and Grafiti images. From these results it is observed that compressed image obtained using pairFuzzy method has proper contrast, smooth edges and negligible artifacts as compared to JPEG and Fuzzy transform. The presented image results is at compression of  $\sim 0.144$  bpp (bits per pixel). For objective analysis, widely used statistical parameters such as: peak signal to noise ratio (PSNR), mean square error (MSE), mean structural similarity index measure (MSSIM) [37] and feature



**Fig. 11** Graffiti test image from the database of Oxford University, UK [12]. **a** Graffiti original, **b** edge image, **c** JPEG, **d** fuzzy transform, **e** fuzzy transform with ARFSM, **f** pairFuzzy

similarity index measure (FSIM) [40] are calculated. Peak signal to noise ratio (PSNR) is defined as

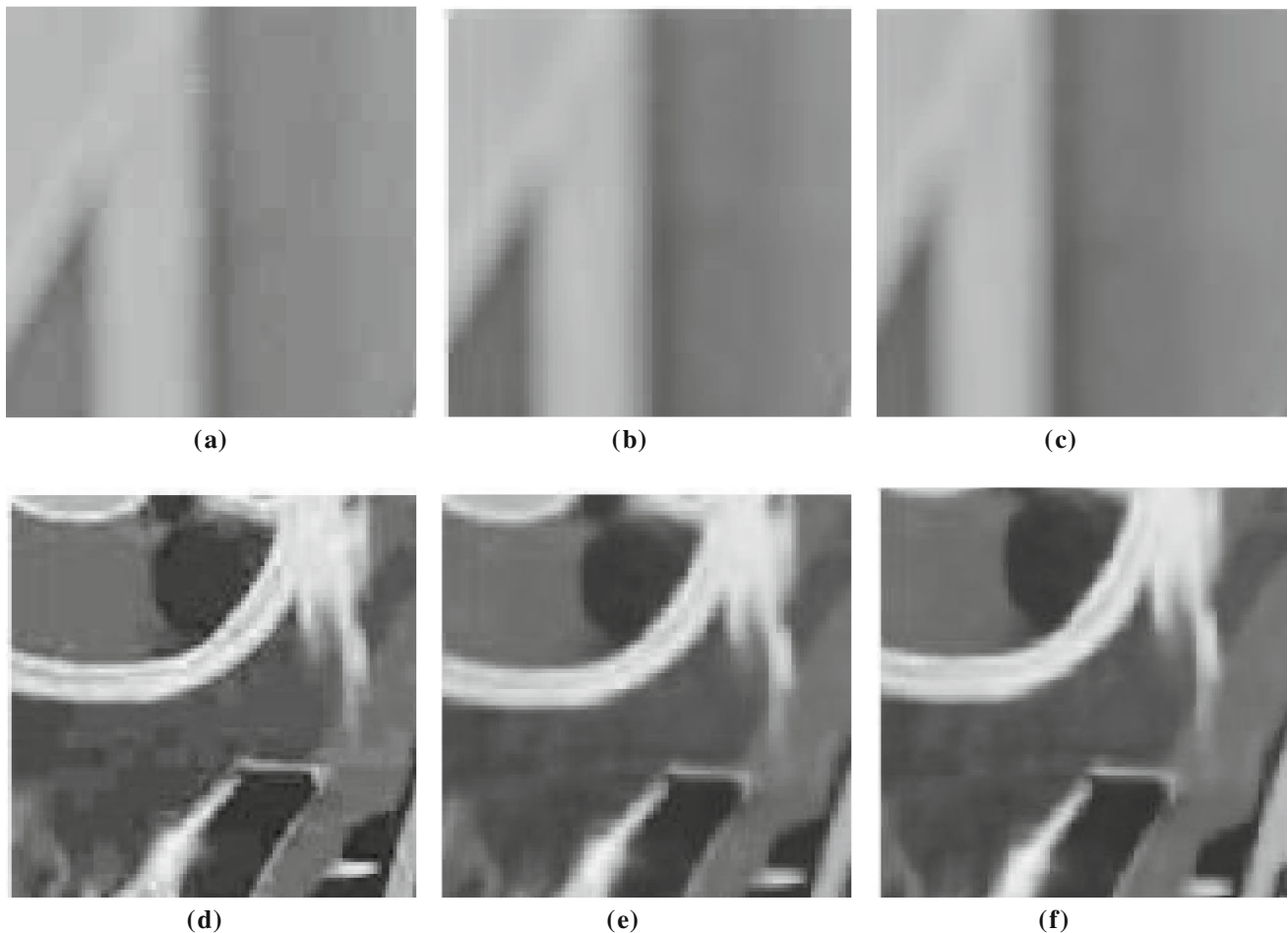
$$PSNR(X, Y) = 10 \log_{10} \left( \frac{L^2}{MSE} \right) \quad (13)$$

where  $L$  is the maximum possible value of intensity (for 8 bit image,  $L = 255$ ) and mean square error (MSE) [36] is defined as

$$MSE(X, Y) = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M [X(i, j) - Y(i, j)]^2}{M \times N}} \quad (14)$$

where  $M \times N$  is total number of pixels in an image and  $X(i, j)$  and  $Y(i, j)$  are the gray level pixel values of original and reconstructed image respectively.

In general, a compression technique resulting in high PSNR and low MSE is generally considered as a better technique. Tables 2 and 3 indicates that the proposed algorithm outperforms all above said algorithms in terms of PSNR and MSE. Though PSNR and MSE are most commonly used measures as they are easy to calculate and have clear physical meaning but they are not well matched to visual image quality. These measures are based on pixel level calculation and do not take into account the inter-dependencies between pixels and hence not sufficient for evaluating the performance of an algorithm. Since human visual system understands an image mainly according to its low-level features and feature similarity index measure (FSIM) brings image assessment from pixel level to feature level, hence preferred in this paper. In



**Fig. 12** Zoomed images for comparison. **a** Zoom Lenna JPEG, **b** zoom Lenna fuzzy transform, **c** zoom Lenna pairFuzzy, **d** zoom Graffiti JPEG, **e** zoom Graffiti fuzzy transform, **f** zoom Graffiti pairFuzzy

FSIM, phase congruency (PC) measures the significance of local structures and gradient magnitude (GM) takes into account the visual contrast. FSIM however does not consider various visual masking effects. Structural similarity index measure (SSIM) also avoids pixel wise calculation and considers loss of structure in an image, hence provides good approximation of perceived image quality. FSIM and MSSIM are good measures for assessing the quality of image.

Structural similarity index measure (SSIM) measures the structural similarity between the original image and the reconstructed image and is calculated as

$$SSIM(X, Y) = \frac{(2\mu_x\mu_y + (K_1L)^2)(2\sigma_{xy} + (K_2L)^2)}{(\mu_x^2 + \mu_y^2 + (K_1L)^2)(\sigma_x^2 + \sigma_y^2 + (K_2L)^2)} \tag{15}$$

where  $\mu_x, \mu_y$  and  $\sigma_x, \sigma_y$  are mean intensities and standard deviations for X and Y respectively,  $K_x$  and  $K_y$  are

constants such that  $0 < K_x, K_y < 1$ . For image quality assessment, SSIM index is applied locally rather than globally. The local statistics and the SSIM are computed within a square window of size  $m \times m$ , which moves from one pixel to another over the whole image. To compute the overall measure of the complete image, a mean structural similarity index measure (MSSIM) is computed as

$$MSSIM(X, Y) = \frac{1}{N} \sum_{i=1}^N SSIM(x_i, y_i) \tag{16}$$

where X and Y are the original and reconstructed images respectively,  $x_i$  and  $y_i$  are the contents of image at the  $i$ th local window and N is the number of local windows in the image. A large value of MSSIM shows its ability to retain the structural similarity of the original image. For two identical images, MSSIM is equal to 1.

**Table 2** Comparison results of different algorithms in terms of MSE

Test images	256 × 256				512 × 512			
	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy
Lenna	171.03	133.99	125.04	119.14	45.50	40.28	35.24	34.52
Goldhill	207.05	169.85	157.42	147.59	78.53	52.36	47.21	45.71
Peppers	180.33	146.58	143.57	120.24	43.16	36.06	34.68	30.91
Cameraman	232.31	211.87	190.58	160.35	100.48	70.97	54.96	52.49
House	117.78	91.00	88.93	82.43	27.61	25.01	23.99	23.17
Jetplane	219.32	204.68	164.46	155.26	100.25	97.07	66.54	63.84
Woman blonde	153.84	124.48	116.70	101.40	133.07	110.94	90.79	72.79
Living room	230.18	197.73	169.85	158.15	129.44	118.59	100.71	84.73
Graffiti	666.93	550.90	487.61	412.17	331.19	275.47	248.35	242.14
Wall	235.54	225.98	210.90	200.02	287.13	273.57	228.07	215.81
Leuven	360.64	294.49	255.90	228.07	246.08	208.48	178.27	167.52
Boat	490.99	411.22	394.53	370.74	533.43	487.61	415.03	355.69
Trees	489.86	440.63	382.01	341.25	444.71	417.90	294.49	280.59
Bikes	200.48	175.42	159.25	143.57	141.60	123.90	115.36	109.92
Bark	126.78	115.63	113.78	101.64	67.46	62.96	61.38	57.82

**Table 3** Comparison results of different algorithms in terms of PSNR (dB)

Test images	256 × 256				512 × 512			
	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy
Lenna	25.80	26.86	27.16	27.37	31.55	32.08	32.66	32.75
Goldhill	24.97	25.83	26.16	26.44	29.18	30.94	31.39	31.53
Peppers	25.57	26.47	26.56	27.33	31.78	32.56	32.73	33.23
Cameraman	24.47	24.87	25.33	26.08	28.11	29.62	30.73	30.93
House	27.42	28.54	28.64	28.97	33.72	34.15	34.33	34.48
Jetplane	24.72	25.02	25.97	26.22	28.12	28.26	29.90	30.08
Woman blonde	26.26	27.18	27.46	28.07	26.89	27.68	28.55	29.51
Living room	24.57	25.17	25.83	26.14	27.01	27.39	28.10	28.85
Graffiti	19.89	20.72	21.25	21.98	22.93	23.73	24.18	24.29
Wall	24.41	24.59	24.89	25.12	23.55	23.76	24.55	24.79
Leuven	22.56	23.44	24.05	24.55	24.22	24.94	25.62	25.89
Boat	21.22	21.99	22.17	22.44	20.86	21.25	21.95	22.62
Trees	21.23	21.69	23.31	22.80	21.65	21.92	23.44	23.65
Bikes	25.11	25.69	26.11	26.56	26.62	27.20	27.51	27.72
Bark	27.10	27.50	27.57	28.06	29.84	30.14	30.25	30.51

The feature similarity measure (FSIM) evaluates the similarity between two images by computing locally the combination of the phase congruency (PC) and gradient magnitude (GM) information. PC and GM are complementary to each other and they reflect different aspects of

human visual system in assessing the local image quality. PC provides information about local structures in an image whereas GM provides contrast information.

The overall similarity index for grayscale images is defined as

**Table 4** Comparison results of different algorithms in terms of MSSIM

Test images	256 × 256				512 × 512			
	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy
Lenna	0.736	0.807	0.824	0.833	0.868	0.950	0.960	0.970
Goldhill	0.597	0.625	0.672	0.681	0.861	0.874	0.904	0.937
Peppers	0.824	0.834	0.860	0.862	0.892	0.956	0.968	0.979
Cameraman	0.738	0.808	0.842	0.845	0.902	0.930	0.946	0.948
House	0.756	0.851	0.864	0.871	0.861	0.957	0.968	0.968
Jetplane	0.764	0.808	0.836	0.842	0.917	0.933	0.943	0.948
Woman blonde	0.725	0.802	0.823	0.826	0.816	0.912	0.921	0.931
Living room	0.657	0.676	0.726	0.730	0.789	0.869	0.875	0.900
Graffiti	0.654	0.699	0.753	0.756	0.872	0.915	0.919	0.923
Wall	0.439	0.510	0.529	0.538	0.721	0.801	0.808	0.835
Leuven	0.711	0.739	0.756	0.757	0.835	0.901	0.900	0.911
Boat	0.560	0.626	0.724	0.727	0.772	0.857	0.878	0.892
Trees	0.520	0.532	0.636	0.643	0.747	0.815	0.849	0.863
Bikes	0.632	0.728	0.737	0.740	0.815	0.889	0.888	0.902
Bark	0.643	0.702	0.703	0.704	0.827	0.893	0.898	0.908

**Table 5** Comparison results of different algorithms in terms of FSIM

Test images	256 × 256				512 × 512			
	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy	JPEG	Fuzzy transform	Fuzzy transform with ARFSM	pairFuzzy
Lenna	0.858	0.862	0.873	0.879	0.866	0.957	0.964	0.970
Goldhill	0.788	0.793	0.823	0.828	0.835	0.927	0.947	0.955
Peppers	0.776	0.859	0.877	0.882	0.945	0.955	0.965	0.973
Cameraman	0.780	0.812	0.846	0.849	0.874	0.929	0.942	0.946
House	0.847	0.851	0.866	0.867	0.857	0.954	0.961	0.962
Jetplane	0.786	0.825	0.845	0.850	0.836	0.934	0.945	0.950
Woman Blonde	0.777	0.857	0.873	0.878	0.894	0.943	0.956	0.957
Living Room	0.724	0.773	0.806	0.808	0.894	0.909	0.923	0.933
Graffiti	0.711	0.776	0.806	0.812	0.833	0.925	0.925	0.929
Wall	0.644	0.724	0.742	0.742	0.810	0.906	0.913	0.922
Leuven	0.739	0.807	0.819	0.820	0.924	0.928	0.929	0.935
Boat	0.734	0.751	0.812	0.815	0.806	0.902	0.917	0.922
Trees	0.682	0.734	0.805	0.805	0.835	0.911	0.930	0.934
Bikes	0.787	0.821	0.827	0.829	0.890	0.930	0.934	0.939
Bark	0.779	0.806	0.808	0.810	0.913	0.931	0.938	0.944

**Table 6** PSNR comparison for artifact reduction methods

Test images	256 × 256		
	Zou [41]	Luo [22]	Kim [14]
Lenna	26.31	26.60	26.75
Goldhill	25.40	25.41	26.05
Peppers	26.01	26.10	26.37
Cameraman	24.67	25.27	25.05
House	27.95	28.15	28.66
Jetplane	24.86	24.08	25.25
Woman blonde	26.71	27.20	26.95
Living room	24.86	25.47	25.08
Graffiti	20.30	20.33	20.89
Wall	24.50	25.03	24.93
Leuven	23.00	22.60	23.30
Boat	21.59	22.29	22.33
Trees	21.45	20.96	21.61
Bikes	25.40	25.37	25.56
Bark	27.29	26.53	27.49

$$FSIM(X, Y) = \frac{\sum_i \sum_j S(i, j) \max\{PC_x(i, j), PC_y(i, j)\}}{\sum_i \sum_j \max\{PC_x(i, j) PC_y(i, j)\}} \quad (17)$$

where  $PC_x$  and  $PC_y$  are the local phase congruency values determined for the original image and the reconstructed image respectively, and  $S(i, j)$  is the local similarity value determined as

$$S(i, j) = \left\{ \frac{2 \cdot PC_x(i, j) \cdot PC_y(i, j) + K_{PC}}{PC_x^2(i, j) + PC_y^2(i, j) + K_{PC}} \right\} \times \left\{ \frac{2 \cdot GM_x(i, j) \cdot GM_y(i, j) + K_{GM}}{GM_x^2(i, j) + GM_y^2(i, j) + K_{GM}} \right\} \quad (18)$$

where  $K_{PC}$  and  $K_{GM}$  are small stabilizing constants. The value of  $K_{PC}$  depends on the dynamic range of  $PC$  values. The value of  $GM$  can be determined using gradient operators such as the Prewitt operator, the Sobel operator or the Scharr operator. The detailed analysis of the phase congruency measurement is presented in the paper [19]. Tables 4 and 5 show the values of MSSIM and FSIM

indices respectively obtained using different algorithms for different images and it is observed that there is significant improvement in these parameters using the proposed algorithm as compared to above-said algorithms. Table 6 shows comparison of PSNR values obtained for various images of size  $256 \times 256$  using different artifact reduction methods presented in [14, 22, 41] when applied to improved fuzzy transform.

Figure 13 shows the performance comparison of various algorithms to proposed method in terms of percentage reduction in MSE and percentage gain in PSNR for different test images. It is concluded that the performance of pair-Fuzzy algorithm has significantly improved over JPEG while compressing both

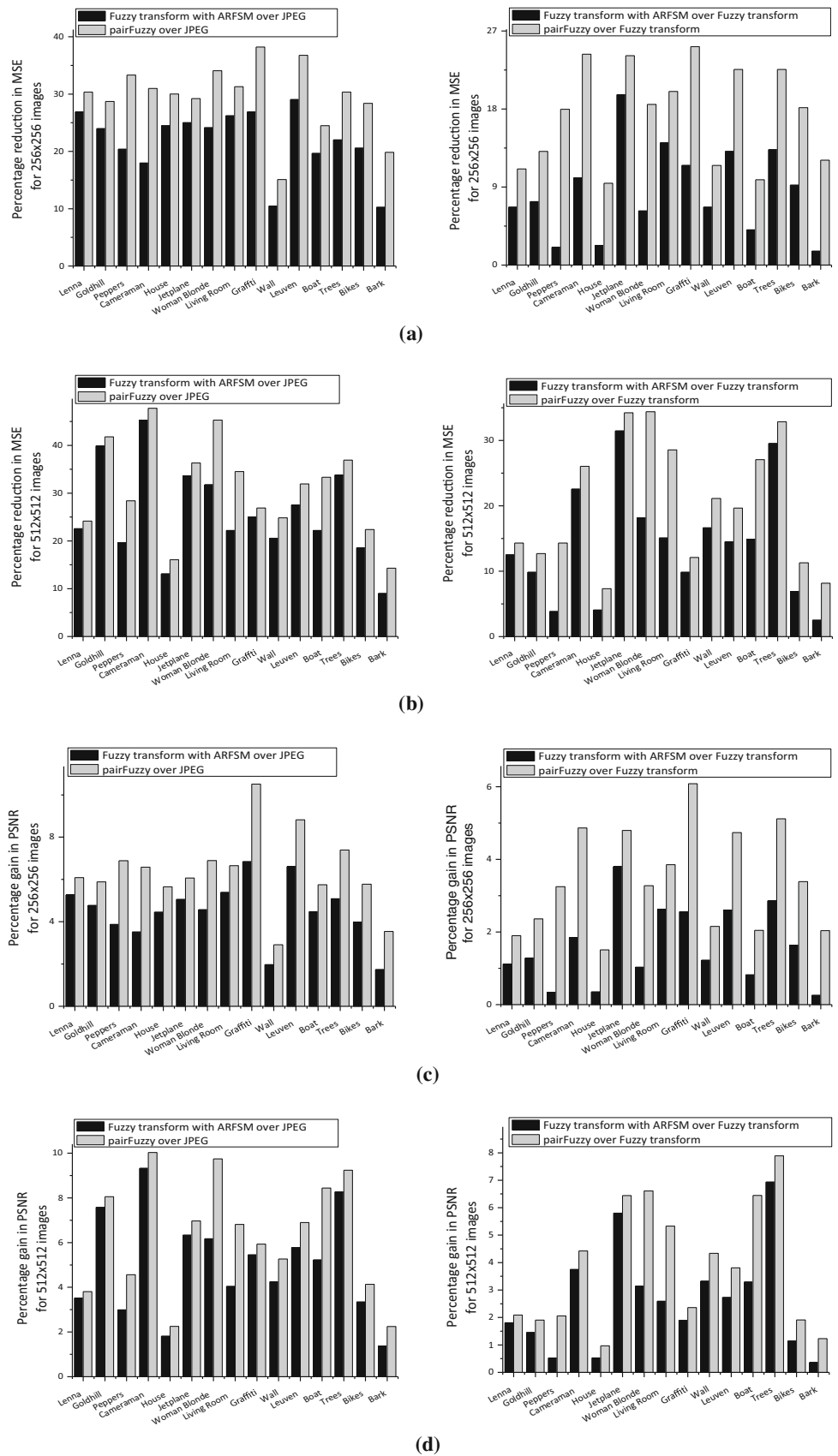
$256 \times 256$  size image as well as  $512 \times 512$  image. It is further concluded that there is significant improvement in the performance of fuzzy transform over JPEG, fuzzy transform with ARFSM over fuzzy transform and in turn pairFuzzy over fuzzy transform with ARFSM.

## 5 Conclusion

This paper proposes an image compression method named pairFuzzy. The proposed method preprocesses an image using competitive fuzzy edge detector algorithm that results in classification of each image block as either LV, MV and HV. These blocks are then compressed and decompressed using improved fuzzy transform. The reconstructed image may contain the blocking artifacts that are further reduced using fuzzy switched median filter after reconstructing the image. The proposed method is both subjectively and objectively analyzed as well as compared to other methods and found to be superior in terms of quality of the reconstructed image and artifact reduction as compared to JPEG and fuzzy transform algorithm. The proposed method presented in this paper achieves significant improvement in terms of quality of the image and reducing the artifacts but still it needs more improvement. In future, the advantage of neural network alongwith fuzzy transform can be used to develop image compression algorithm at low bit rate.



**Fig. 13** Performance comparison of various algorithms in terms of percentage reduction in MSE and percentage gain in PSNR. **a** Percentage reduction in MSE for various test images of sizes  $256 \times 256$ , **b** percentage reduction in MSE for various test images of sizes  $512 \times 512$ , **c** percentage gain in PSNR for various test images of sizes  $256 \times 256$ , **d** percentage gain in PSNR for various test images of sizes  $512 \times 512$



## References

- Amarunnishad TM, Govindan VK, Mathew Abraham T (2008) Improving BTC image compression using a fuzzy complement edge operator. *Signal Process* 88(12):2989–2997
- Avramovic A (2011) Lossless compression of medical images based on gradient edge detection. In: Proceedings of the 19th telecommunications forum (TELFOR), Belgrade, pp 1199–1202
- Biswas M, Kumar S, Nguyen TQ, Balram N (2007) Support vector machine (SVM) based compression artifact-reduction technique. *J Soc Inf Disp* 15(8):625–634
- Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 6:679–698
- De A, Guo C (2014) An image segmentation method based on the fusion of vector quantization and edge detection with applications to medical image processing. *Int J Mach Learn Cybernet* 5(4):543–551
- Desai U, Masaki I, Chandrakasan A, Horn BKP (1996) Edge and mean based image compression. In: Proceedings of IEEE acoustics, speech, and signal processing, ICASP 1996, p 49
- Di Martino F, Loia V, Perfilieva I, Sessa S (2008) An image coding/decoding method based on direct and inverse fuzzy transforms. *Int J Approx Reason* 48(1):110–131
- Di Martino F, Loia V, Sessa S (2003) A method for coding/decoding images by using fuzzy relation equations. In: Fuzzy sets and systems IFSA 2003, lecture notes in computer science. Springer, Berlin Heidelberg, pp 436–441
- Gambhir D, Rajpal N (2012) Fuzzy edge detector based adaptive quantization image coding: FuzzAQC. In: Proceedings of the recent advances in information technology (RAIT), pp 101–106
- Gambhir D, Rajpal N (2013) Fuzzy edge detector based blocking artifacts removal of DCT compressed images. In: Proceedings of the circuits, controls and communications (CCUBE), pp 1–6
- Gambhir D, Rajpal N (in Press) Image coding using fuzzy edge classifier and fuzzy F-transform: dualFuzzy. *Int J Fuzzy Comput Model*. ISSN online: 2052–3548
- <http://www.robots.ox.ac.uk/vgg/data/data-aff.html>. Accessed September, 2014
- Kathirvalavakumar T, Ponmalar E (2013) Self organizing map and wavelet based image compression. *Int J Mach Learn Cybernet* 4(4):319–326
- Kim J, Sim C-B (2011) Compression artifacts removal by signal adaptive weighted sum technique. *IEEE Trans Consum Electron* 57(4):1944–1952
- Lee S, Park SJ (2012) A new image quality assessment method to detect and measure strength of blocking artifacts. *Signal Process Image Commun* 27(1):31–38
- Liang LR, Looney CG (2003) Competitive fuzzy edge detection. *Appl soft comput* 3(2):123–137
- Liu L, Wang Y (2009) A mean-edge structural similarity for image quality assessment. In: Fuzzy systems and knowledge discovery, 2009. FSKD'09. Sixth international conference on, vol 5, IEEE, pp 311–315
- Liu S, Bovik AC (2002) Efficient DCT-domain blind measurement and reduction of blocking artifacts. *IEEE Trans Circuits Syst Video Technol* 12(12):1139–1149
- Liu Z, Laganière R (2007) Phase congruence measurement for image similarity assessment. *Pattern Recognit Lett* 28(1):166–172
- Looney CG, A fuzzy classifier network with ellipsoidal epanechnikovs. [www.cse.unr.edu/looney/cs674/unit2/fuzclassifier-total.pdf](http://www.cse.unr.edu/looney/cs674/unit2/fuzclassifier-total.pdf)
- Lossy image compression (2008) In: Encyclopedia of GIS. Springer, Berlin
- Luo Y, Ward RK (2003) Removing the blocking artifacts of block-based DCT compressed images. *IEEE Trans Image Process* 12(7):838–842
- Mainberger M, Bruhn A, Weickert J, Forchhammer S (2011) Edge-based compression of cartoon-like images with homogeneous diffusion. *Pattern Recogn* 44(9):1859–1873
- Mainberger M, Weickert J (2009) Edge-based image compression with homogeneous diffusion. In: Computer analysis of images and patterns, pp 476–483. Springer, Berlin
- Milanova M, Kountchev R, Todorov V, Kountcheva R (2007) Pre- and post- processing for enhancement of image compression based on spectrum pyramid. Innovative algorithms and techniques in automation, industrial electronics and telecommunications. Springer, Netherlands, pp 269–274
- Nie Y, Barner KE (2003) Optimized fuzzy transformation for image deblocking. In: Multimedia and expo, 2003. ICME '03. In: Proceedings. 2003 international conference on, volume 1, pp 541–544
- Perfilieva I (2006) Fuzzy transforms. *Fuzzy Sets Syst* 15(8):993–1023
- Perfilieva I, De Baets B (2010) Fuzzy transforms of monotone functions with application to image compression. *Inf Sci* 180(17):3304–3315
- Petrosino A, Ferone A (2009) Rough fuzzy set-based image compression. *Fuzzy Sets Syst* 160(10):1485–1506
- Saha S (2000) Image compression: from DCT to wavelets: a review. *ACM Crossroads* 6(3):12–21
- Singh J, Singh S, Singh D, Uddin M (2011) A signal adaptive filter for blocking effect reduction of JPEG compressed images. *AEU Int J Electron Commun* 65(10):827–839
- Toh KKV, Mahyuddin MNH (2008) Salt-and-pepper noise detection and reduction using fuzzy switching median filter. *IEEE Trans Consum Electron* 54(4):1956–1961
- Toh KKV, Isa NAM (2010) Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Process Lett* 17(3):281–284
- Tsushima Y, Komine K, Sawahata Y, Hiruma N (2014) Higher resolution stimulus facilitates depth perception: Mt+ plays a significant role in monocular depth perception. *Scientific reports: Nature*, 4(6687)
- Wang L, Jiao L, Wu J, Shi G, Gong Y (2010) Lossy-to-lossless image compression based on multiplier-less reversible integer time domain lapped transform. *Signal Process Image Commun* 25(8):622–632
- Wang Z, Bovik AC (2009) Mean squared error: love it or leave it? a new look at signal fidelity measures. *IEEE Signal Process Mag* 26(1):98–117
- Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
- Wu X, Zhang X, Wang X (2009) Low bit-rate image compression via adaptive down-sampling and constrained least squares up-conversion. *IEEE Trans Image Process* 18(3):552–561
- Xia Q, Li X, Zhuo L, Lam KM (2011) A novel low-bit-rate image compression algorithm. In: Advances in multimedia information processing-PCM 2010. Springer, Berlin, pp 100–110
- Zhang L, Zhang D, Mou X (2011) FSIM: a feature similarity index for image quality assessment. *IEEE Trans Image Process* 20(8):2378–2386
- Zou JJ, Yan H (2005) A deblocking method for BDCT compressed images based on adaptive projections. *IEEE Trans Circuits Syst Video Technol* 15(3):430–435