

Fractal Image Compression for Classification of PD Sources

E. M. Lalitha and L. Satish

Dept. of High Voltage Engg., Indian Institute of Science, Bangalore, India.

ABSTRACT

The fractal image compression technique has an unique feature due to which physical position of blocks/regions in the input image can be extracted directly from the compressed data. Applying this technique, ϕ - q - n partial discharge (PD) patterns (treated as an image) are compressed and stored as affine transformations. These transformations then are used directly to extract the embedded pattern features, which are classified by a neural network. The novel route to PD pattern classification described in this paper thus addresses both the tasks of compression and feature extraction in a single step. The task of compression is essential to store and handle large quantities of pattern data acquired, especially during on-line monitoring of PD in power apparatus. Results presented illustrate that this approach can address satisfactorily the tasks of compression and classification of PD patterns.

1 INTRODUCTION

MEASUREMENTS of partial discharge (PD) are performed on HV power apparatus to assess design adequacies and overall integrity of the insulation system. These measurements have been accepted the world over as a quality control check after manufacture, and also as a non-destructive tool for purposes of monitoring and diagnostics [1]. In spite of great advances in manufacturing techniques, tiny defects are inevitable and can be detected during PD measurements, with background knowledge of the system and operator expertise. In the early days, the continuously changing elliptical patterns formed the basis for such interpretations [2]. Since the past few years, digital PD measurement has gained popularity, as it enables the measured PD quantities to be archived and post-processed, thereby increasing the possibilities of building systems for PD interpretation.

A digital PD system is basically an extension of the analog PD system. It is designed and built to capture every PD event (to the extent permissible by hardware limitations) in a given interval of time, and quantifies this by the apparent charge magnitude q , and the phase angle ϕ , at which it occurs. The final result is popularly referred to as the ϕ - q - n pattern. It is well known that analyzing certain characteristics of these patterns, the source or cause of PD can be ascertained. Additionally, this mode of data gathering is widely accepted and happens to be an option provided in many of the commercially available digital PD systems. For this reason, the ϕ - q - n pattern has been chosen as the input in this work, and a typical pattern is shown in Figure 1, as an image using 64 grayscale levels.

Various factors influence the internal PD process, like the random nature of the discharge, the applied voltage magnitude and its duration, insulation material, defects if any, state of aging, etc. [1, 2]. Influence of these factors on the ϕ - q - n pattern are seen as changes at a macro-

scopic level, since it is an integrated plot, that can be described as dilation, contraction, shift, or any combination of these, on the PD pulse-clusters. In this context, it must be mentioned, that to characterize long-term aging behavior, explaining mechanisms and physics of discharges, etc., the pulse-sequence analysis [3] is particularly better suited. Thus, subsequent ϕ - q - n patterns will not be identical, but will possess only certain salient characteristics that are representative of a particular defect. The ϕ - q - n patterns are, in general very complex in shape, changing constantly, and hence its classification is indeed a challenging task.

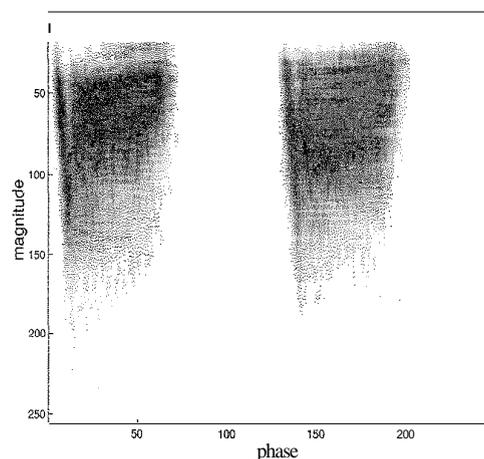


Figure 1. An example ϕ - q - n pattern (cavity discharge)

The topic of developing systems for automatic interpretation of PD patterns has been an area of active research, since the past few years. Different alternatives, like the use of expert systems [4], neural networks [5], statistical approaches [6], fuzzy logic [7], fractal geometry [8], hid-

den Markov models [9], and Fourier transforms [10] have been examined with varying degrees of success. In many of these systems, the ϕ - q - n pattern was used as the input. As these patterns contain large amounts of data (typically a 256×256 pattern with 0 to 255 gray scales will have 128KB of data when represented as integers), it is not convenient to be processed directly. A practical PD classification system would require numerous examples from each class to increase its robustness, which calls for a large storage space, apart from computational time and effort. Such a situation is often encountered during on-line monitoring, where a large number of PD patterns are collected routinely during the service-life of power apparatus, and these in turn need to be analyzed for trends and changes. To overcome this difficulty, data reduction methods have been employed. A simple, yet effective procedure involves averaging of the neighboring elements, thereby reducing the size of the input pattern, which in turn results in a coarser pattern. The finer aspects of the input are lost due to the averaging process, which is its limitation. Also, this data reduction operation is irreversible, because the original pattern cannot be retrieved from the averaged data. Using a slightly different route, other methods directly extract a few characteristic features, like statistical parameters [6], fractal quantities [8], and so on. Classification ability of such systems depend on the ability of the defined features to represent the salient aspects of the pattern. Here again, the feature extraction operation is irreversible and the original pattern cannot be retrieved from the features.

A need for retrieval of the original pattern can be visualized in the event of a misclassification, when an alternative analysis may be performed using the original pattern data, if available. Such situations are often encountered during practical PD testing. In addition, when a complex ϕ - q - n pattern is encountered, there often is a need for consulting a human expert, at which time there is a need to display such patterns, one after another. This is obvious, since human beings are definitely more comfortable interpreting such original patterns/objects, than feature vectors. Thus, the need for efficient data storage and retrieval is clear. Since the final goal is to classify PD patterns, efficient storage of data must invariably be followed by suitable data processing, which often is computationally intensive in such pattern recognition tasks, and needs minimization.

With the emerging multimedia systems, handling large data is not a bottleneck, but it would be very desirable and convenient, if the ϕ - q - n patterns could be stored individually in a compressed format. In addition, with graphical user interface facilities and increasing computing power afforded by these machines, they can display these ϕ - q - n patterns in a variety of ways (say as an image, 3-d color plots, display a series of patterns captured at regular intervals as a movie, and so on), in real-time, and thus reveal hidden dependencies, trends, temporal changes in patterns, etc. Such tools definitely would place the human expert in a better position to interpret PD patterns, when faced with difficult situations.

With these as the motivating factors, and after a thorough survey of the available literature, it emerges that the task of data compression is essential, followed by feature extraction, ultimately leading to classification of PD patterns. However, it would be very beneficial if the compressed data themselves could represent the embedded features; which would result in appreciable reduction in computational effort. To the

best knowledge of the authors, such a route to PD pattern classification has not been explored earlier, and hence is the subject matter of this paper. Although different features have been defined and extracted, as yet, there exists no consensus amongst researchers, regarding which of them is best suited on the whole, to address the task of PD pattern recognition. Hence, newer alternatives have to be explored.

This work combines the tasks of compression of PD patterns and feature extraction. Using the fractal image compression (FIC) technique, it is shown that these two tasks can be implemented in a single step. Although, many popular compression schemes are available, like DCT, JPEG, neural networks, and more recently wavelets, the authors prefer to employ FIC method, for the primary reason that it readily affords combining the tasks of pattern compression and feature extraction. This arises from the fact that the FIC technique has the unique property of storing spatial coordinates of portions/blocks of image, directly in the affine transformations used to compress data. Using this property along with a defined threshold, it is easy to extract feature information pertaining to the PD pulse-clusters in the ϕ - q - n pattern. Additionally, it has been reported recently, that the fractal based compression approach provides comparable compression ratios, high fidelity, and is easier to decode, when compared to the alternative compression schemes [11]. Due to these advantages, it was considered worthwhile to investigate its potential for the task of PD classification.

An introduction to the FIC technique and a suggested modification to the existing quadtree partitioning FIC (QPFIC) algorithm for storing transformations pertaining to the PD-pulse clusters alone (and thereby obtain higher compression ratios) is given in Section 2. Section 3 describes the simulations performed, results and discussions. Conclusions of this work are listed in Section 4.

2 BRIEF OVERVIEW OF FRACTALS

Mandelbrot is considered to be one of the first to propound ideas and principles of fractal geometry [12]. He coined the word 'fractal', and its concepts were shown to possess properties suitable for modeling and accurately describing natural shapes and phenomena, for which task conventional Euclidean geometry was recognized to be inadequate. The underlying characteristic of a fractal is that it is generated by an iterative process, starting from an initial tile. A formal definition of the fractal is based on this property, and it is any shape where the parts, when magnified, reveal as much detail as the whole. This quality is called self-similarity, and the fractal object can be regarded to be composed of many copies of itself, each of which is scaled-down with possible translations and rotations. Most of the mathematically generated fractals (Serpenski's triangle, Koch curve, space filling curve, etc.) possess this self-similar property, whereas those abundantly occurring in real world (clouds, coastlines, mountains, landscapes, etc.) are only statistically self-similar *i.e.*, these are only statistically scale-invariant when magnified. Apart from self-similarity fractals are associated with a number called the fractal dimension, which appeals intuitively to its roughness. Mathematically, a fractal can be represented by the set of equations given below

$$W \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (1)$$

Starting with an initial value of x and y , the above equation when recursively iterated generates a fractal. In the above Equation, e and f perform translation and a, b, c, d , the rotation and skewing operations. The set $\{W_j\}$ is called an affine transformation. These affine transformations are such that they bring subsequent iterates topologically closer and hence are called contractive affine transformations. A fractal can be represented by a set of such contractive affine transformations and is given by

$$W_n, p_n \quad n = 1, 2, \dots, N \quad (2)$$

where W_n are the affine transformations and p_n denotes their individual importance. This representation is called the iterated function system (IFS) code. Every IFS code has a unique geometrical object or fractal associated with it, usually called the attractor of the IFS. The mathematical basis for this representation is given by the Collage theorem [12]. The inverse operation of the above principle, *i.e.* representation of an object by a few transformations led to the idea and concepts of compression theory. The fact, that only a few transformations could entirely represent an object was the underlying principle. These transformations when recursively iterated generate the object. This led to the general idea of the FIC using affine transformations.

2.1 FRACTAL IMAGE COMPRESSION

Barnsley [13] was first to suggest that storing images as a collection of transformations could lead to image compression. For an exactly self-similar image, the degree of compression obtained is very high and reconstruction possible without loss of any information. Applicability of this principle was extended to compress any image (which can be considered as statistically self-similar), by Jacquin [14]. Due to this development, any input image could be compressed. In this approach, a region/portion of the image is searched with the rest of the image in an effort to find a suitable portion, which is similar in a statistical sense [14, 15]. This technique is basically a search process and consists of the following steps: partitioning the image into sub-images or blocks, search for parts of the images or blocks which are self-similar in a statistical sense, and once a match is located, compute the transformations.

In the following Sections, a brief account of the partitioning scheme employed (quadtree partitioning), procedure involved in computing the transformations for a given image (encoding algorithm), and the steps to reconstruct the image from the transformations (decoding algorithm) are presented [14, 15].

2.2 QUADTREE PARTITIONING

There are many ways of partitioning an image [15]. Some of the popular schemes are quadtree partitioning (QP), horizontal-vertical partitioning, triangular partitioning, polygonal partitioning, *etc.*. In this work, the QP scheme was chosen as it is a typical fractal procedure, inherently suited for square images, and also comparatively easier to code and implement when compared to the other schemes. The horizontal-vertical and polygonal partitioning schemes are particularly well suited when images to be compressed are non-square [15]. However, as the goal was to highlight the potential of the FIC, only results from quadtree partitioning scheme have been presented for purposes of illustration.

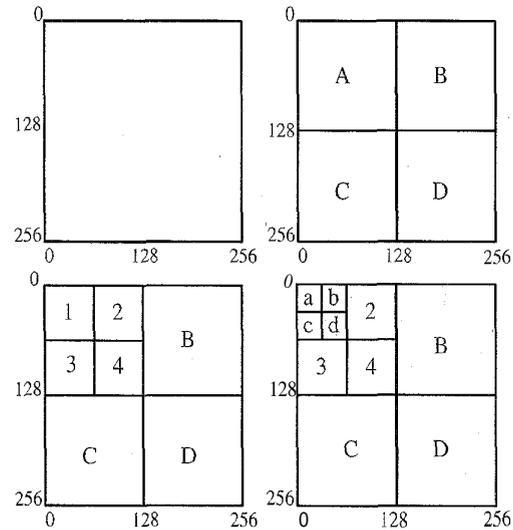


Figure 2. Recursive quadtree partitioning. (a) no QP, (b) after 1 step of QP, (c) after 2 steps of QP, (d) after 3 steps of QP.

In QP, at every partition step, the square in the image (largest contained square, if the input is non-square) is divided into 4 sub-images as illustrated in Figure 2. The sub-images are named as 'ranges' and 'domains'. The sub-images labeled 1, 2, 3, and 4 in Figure 2(c) represent the range blocks and A, B, C, D in Figure 2(b) are the domains. The size of domains is twice that of the ranges. Quadtreeing is a recursive process, and depending on the depth to which quadtreeing is performed, size of range and domain blocks are determined. Presence of similarity in the image is judged by comparing the range and domain blocks. Usually, a minimum and maximum depth of partition are defined. Dependence of the quality of the reconstructed image and the resulting compression ratios are discussed as a function of the depth of partition, later in Section 3.

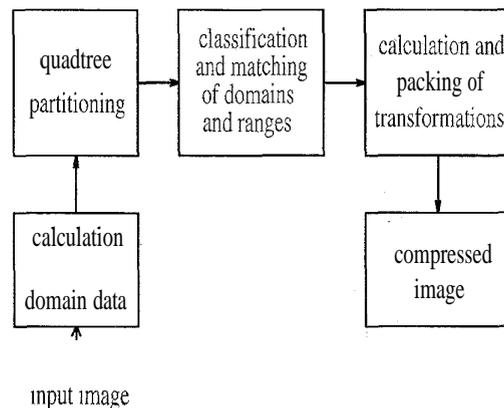


Figure 3. Encoding QPFIC algorithm.

2.3 ENCODING ALGORITHM

The forward process of compression (*i.e.* obtaining transformations of an input image) is realized by this step. The block diagram of the QPFIC encoding algorithm is given in Figure 3. First, the domain data consisting of domain size, location, number of domains and the classes of do-

Table 1. Transformations of the example image from the QPFIC algorithm. r_x, r_y :left top corner coordinates of range block, r_{rx}, r_{ry} :right bottom corner coordinates of range block, o : offset, the brightness setting of the range block. All the values for dx, dy , the left top corner coordinates of domain block matched with range block, the symmetry operation performed for domain-range match and the scale contrast setting of the range block are zero.

	r_y	r_{rx}	r_{ry}	dx	dy	ym.op	scale	o
	0	64	64	0	0	0	0	0
	0	128	64	0	0	0	0	0
	64	64	128	0	0	0	0	0
	64	128	128	0	0	0	0	76.3
	0	192	64	0	0	0	0	0
	0	256	64	0	0	0	0	0
	64	192	128	0	0	0	0	76.3
	64	256	128	0	0	0	0	2
9	0	128	64	192	0	0	0	0
10	64	128	128	192	0	0	0	76.3
11	0	192	64	256	0	0	0	0
12	64	192	128	256	0	0	0	0
13	128	128	192	192	0	0	0	76.3
14	192	128	256	192	0	0	0	2
15	128	192	192	256	0	0	0	2
16	192	192	256	256	0	0	0	0

mains are computed. Classification of domains is done using a scheme which takes into account the brightness values of the domains (for details, please see [15]). The image is then quadtree partitioned to get the range blocks. Range blocks are classified using the same classification scheme used for the domains, and then a search is carried out to determine the best domain-range match. The match operation is restricted to only those domains and ranges belonging to the same class, and the error between them (tolerance) is computed. If this error value is less than the specified tolerance, the transformation is calculated, else, the image is further partitioned. This process is repeated for the whole image. Thus, classification helps in reducing the search time and computational overhead involved during the matching operation. Lastly, the calculated transformations are packed using a pack routine, before being stored [15].

The mechanism of directly extracting the relevant feature information from the transformations will be explained using an example image, shown in Figure 4(a). This binary image consists of a background (pixel value = 0), and a centrally elevated flat portion shown in black (pixel value = 77). Figure 4(b) depicts two levels of quadtree partition. The numbers shown in Figure 4(b) will be useful in identifying the transformations associated with the blocks they occupy. The transformations for this example image are obtained as explained above, and listed in Table 1. It works out that for this image 16 transformations are required and each one of them consist of the following values

1. Second and third column (r_x, r_y) are the coordinates of left top-corner of range block.
2. Fourth and fifth column (r_{rx}, r_{ry}) are the coordinates of right bottom-corner of range block.
3. dx, dy , all are zero, are the left top-corner coordinates of the domain block with which the range block is matched.
4. Eight column is the symmetry operator to match domain and range blocks.
5. Ninth column (scale) is the contrast setting of the range blocks.

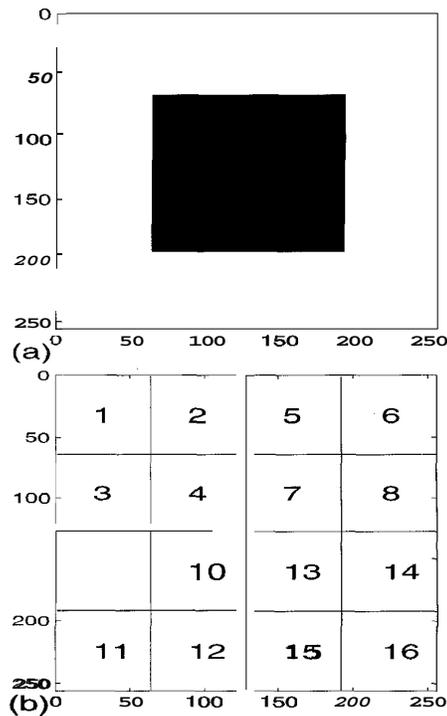


Figure 4. An image and its range blocks. (a) An image, (b) range blocks after 2 steps of QP.

6. Tenth column (offset) gives the offset, which corresponds to the brightness value of the encoded block, and was found to be very nearly the same as the average pixel value.

The transformations thus show that the spatial coordinates of the image are retained in the encoded information. Because of this attribute, the phase and magnitude spread of each of the PD pulse-clusters present in a $\phi-q-n$ pattern (in case of PD images) can be obtained directly from these transformations, which form important features. While applying this algorithm to PD images, a modification is necessary because PD images tend to contain relevant information as pulse clusters, as can be seen in Figure 1 (2 pulse-clusters are present). Only this foreground pulse-cluster information is of interest and the other background information need not explicitly be stored. Hence, the encoding algorithm is suitably modified to store only those transformations pertaining to the PD pulse clusters, by specifying a suitable threshold. Only those transformations whose offset value is higher than the threshold are stored. This threshold, apart from neglecting the background information, also eliminates some spurious, infrequently occurring pulse clusters and helps in obtaining only the relevant information from the $\phi-q-n$ pattern. The choice of threshold depends on two aspects, viz. the noise level in the image, and the quality of the reconstructed image sought for further processing. Thus, for example, to extract information about the elevated portion in the sample image (Figure 4(a)), only those transformations whose offset value is higher than the threshold (background) are necessary to be stored. It results, that for the image considered, transformations that need to be stored are 4, 7, 10, and 13, thus avoiding storage/processing of all the 16 transformations. It is thus

clear, that information pertaining to the phase and magnitude spreads, location on the ϕ - q plane, and the average pixel value of the individual pulse clusters can be extracted directly from the transformations alone. This happens to be the unique advantage of this approach. Elaborate computations involved in the feature extraction operation are thus avoided. Results explicitly showing the effect of thresholding on the reconstructed image and compression ratios achievable are discussed in Section 3.

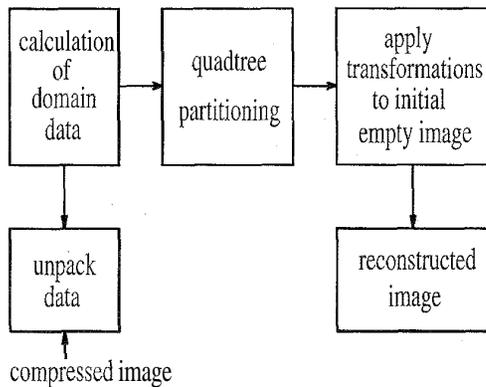


Figure 5. Decoding QPFIC algorithm.

2.4 DECODING ALGORITHM

A block diagram of the decoding algorithm to reconstruct images from the transformations is shown in Figure 5. Basic data, such as maximum and minimum partitions, domain size, etc. are unpacked from the compressed data and domain data such as number of domains are computed. Next, the image is partitioned in a similar way as described in the encoding algorithm. This is a recursive procedure whose recursion tree follows that of the encoding algorithm. Starting with an initial empty image, the transformations are applied, using 8 different cases [15], which perform operations like rotation, skewing and translation to reconstruct the image. The decoding algorithm is suitably modified to reconstruct the image encoded from the modified encoding algorithm. Regions in the image for which the transformations are not stored (due to thresholding) are replaced by zeros in the reconstructed image. This will be seen clearly as patches in the reconstructed image. One such case will be shown in the next Section, where the threshold has been so chosen on purpose to highlight this effect. The basic encoding and decoding algorithms are available at the web site <http://inls3.ucsd.edu/y/Fractals>.

3 SIMULATION

The proposed scheme for classification of PD ϕ - q - n patterns used in this paper is shown in Figure 6. PD image is input to the modified QPFIC algorithm and the transformations are computed. Based on the method of feature extraction, two modes of processing are possible. The first method (mode 1) involves decoding and reconstruction, and thereafter extracting features from the reconstructed image. But, this obviously is a computationally intensive and time consuming process, and seeking alternatives to which, is the goal of this work. The same information can be extracted readily via mode 2 from the transformations themselves, as explained in Section 2. In this work, the mode 2 approach is adopted,

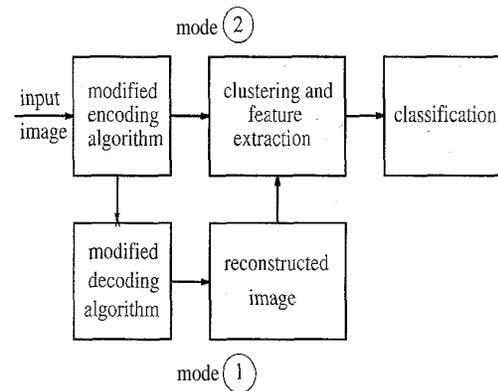


Figure 6. Scheme adopted for PD classification.

and the classification is performed based on features extracted directly from the transformations (*i.e.* compressed data). The PD sources sought for classification are single point corona in air, cavity discharges, surface discharges in air and multiple corona in air. The pattern data used in this work has been drawn from earlier work [16].

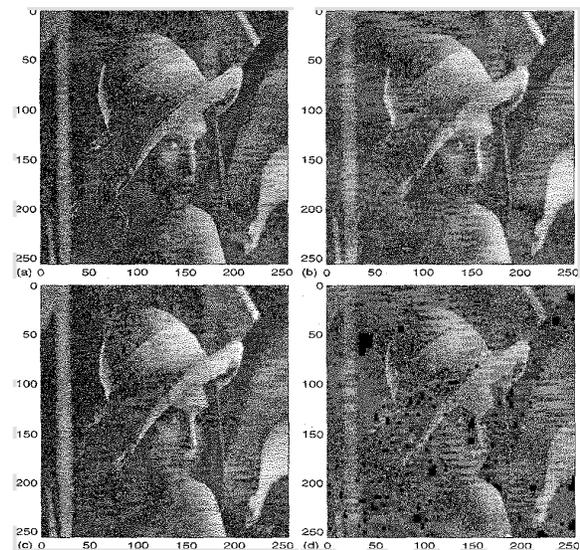


Figure 7. Original and reconstructed Lena images using QPFIC algorithm. (a) Original image, (b) compression = 8.5, error = 8.7, (c) compression = 9.5, error = 9.0, (d) compression = 10.6, error = 28.5.

To verify the proposed modification done to the QPFIC algorithm (*i.e.* thresholding), it was first implemented on the standard Lena image. Figure 7(a) shows the original image. Figure 7(b) is the reconstructed Lena image, where the maximum and minimum number of partitions is six and five respectively and a low tolerance value has been used. By reducing the maximum and minimum number of partitions to five and four respectively, the reconstructed image becomes further coarser and is shown in Figure 7(c). When lower partition depths are used, the resulting ranges/domains are larger in size, and this effectively manifests as 'jaggedness' in the reconstructed image, which is beginning to appear in Figure 7(c). A close comparison of Figures 7(b) and (c) will reveal this aspect. The image in Figure 7(c) is coarser when compared to that in

Figure 7(b), but offers a higher compression ratio. A compromise between the resulting distortion in the reconstructed image and the compression ratios must be made, and this depends very much on the application. Finally, the effect of thresholding is illustrated. Figure 7(d) is the reconstructed image using the modified algorithm, with a high threshold value of fifteen. The compression ratio obtained has increased, but the reconstructed image is visibly distorted. Regions of dark patches in Figure 7(d) indicate that in those regions the threshold was not exceeded and the transformations were not stored. The increase in the compression ratio is a direct consequence of storing transformations only above a specific threshold. This clearly indicates that the value of threshold must be low, and cautiously chosen. Lastly, the proposed modification is only applicable to images which have a well defined foreground and background information, as is the case in PD images. Despite this fact, the proposed modified algorithm was tested on the Lena image, merely for purposes of illustrating the effect of thresholding.

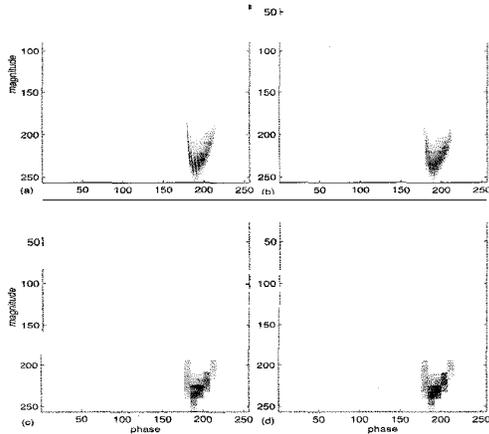
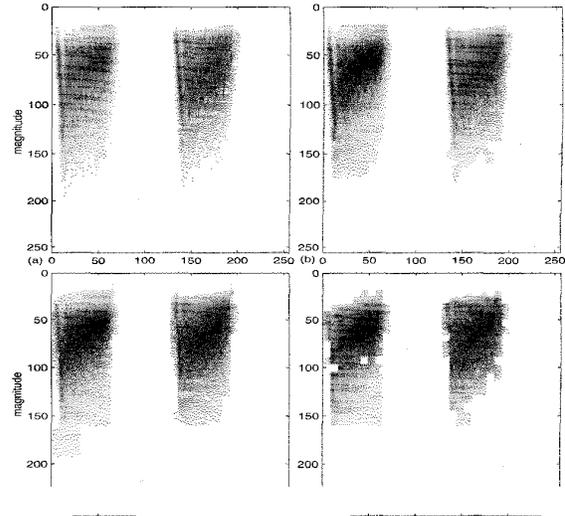


Figure 8. Original and reconstructed images of a simple PD pattern using QPFIC algorithm. (a) Original image. (b) compression= 34, error = 3.8, (c) compression = 133, error = 5.6, (d)compression = 381, error = 5.7.

The different cases discussed for the Lena image are next applied to PD images and the results obtained are shown in Figure 8 for a simple PD image, and in Figure 9 for a considerably complex PD image. Comparing Figures 8(b) and (c) it is seen, that due to lower depths of partition used, the reconstructed figure appears more jagged, and also, finer features of the original are not adequately reconstructed. This not-so-accurate reconstruction is however accompanied by an increase in the compression ratio. Finally, a constant threshold value of five was selected and only transformations with offset values above this threshold were stored. The reconstructed image resulting from this operation is shown in Figure 8(d). The corresponding results for another image are shown in Figures 9(a) to (d). It must be noted that the difference between images shown in parts (c) and (d) arises primarily due to thresholding, and the rest of the parameters remain unchanged. Therefore, if the threshold is carefully chosen, and kept at a low value, increased compression ratio can be achieved, and the reconstructed image will have minimum distortions as well. Hence, a compromise between obtainable compression ratios and quality of the reconstructed images is necessary. This is evident from the images shown in Figures 8 and 9. From the point

Images	ϕqn storage	QPFIC case 1		QPFIC case2		Modified QPFIC	
		ratio	error	ratio	error	ratio	error
Source 1							
256x256	10.4	34	3.8	133	5.6	381	5.7
256x256	10.2	34	5.1	133	7.1	392	7.2
292x48	17.2	22	6.2	23	6.2	53	5.3
292x48	21.0	22	8.6	22	8.6	43	8.9
Source 2							
256x256	1.1	24	4.0	60	7.5	76	10.5
290x42	1.6	6	1.9	7	2.3	18	2.9
290x42	1.3	6	2.2	12	2.7	21	3.9
290x42	1.4	6	2.5	6	2.8	17	2.8
290x42	1.2	6	1.6	6.5	1.7	17	2.6
256x256	1.8	45	2.5	56	2.8	72	4.8
256x256	1.3	26	2.3	56	2.8	72	4.3
290x42	1.1	6	2.3	6	2.8	12	10.9
290x42	1.2	6	3.3	11	3.7	20	4.4
Source 4							
256x256	4.8	72	3.3	80	3.4	135	4.4
256x256	5.6	32	3.0	78	3.0	130	3.3
290x42	4.0	6	4.7	6	5.7	13	5.8
292x54	4.0	7	3.5	12	3.6	17	6.8

in Table 2. Source 1 is single point corona in air, source 2 is cavity discharges, source 3 is surface discharges in air and source 4 is multiple-point corona in air. The compression ratios are evaluated with respect to storing the entire ϕ - q - n pattern (say, a 256×256 matrix) in its raw form. The second column gives the compression ratio, when only the non-zero phase, charge and pulse-count values in the 4-4-72 pattern are stored. Column 3 gives the ratios for case (b) and column 5 gives ratios for case (c) in Figures 8, 9. Here, the effect of decreasing the number of maximum and minimum quadtree partitions on compression ratio is illustrated. Column 7 gives the ratios when using the modified QPFIC algorithm with a threshold value of five. It is observed, that for a simple image like that of source 1, the compression ratio is as high as 392 with the modified QPFIC algorithm. As the complexity of the image to be compressed increases, the compression ratios tend to decrease, as seen from Table 2. A study of Table 2 reveals, that on an average, using modified QPFIC algorithm compression ratios of ~ 60 for a square image, and ~ 15 for a rectangular image can be obtained. Additionally, the modified QPFIC algorithm enhances the compression ratio by a factor of 1.5 to 2 (compare values in columns 5 and 7). It is also observed, that compression ratios seem to be lower for non-square images when compared to square images. The reason for this artifact is that the partitioning scheme adopted is inherently designed for square images. By adopting a different partitioning scheme, comparable ratios can be achieved [15]. The procedure used for feature extraction and its classification abilities will be discussed in the following Sections.

3.1 FEATURE EXTRACTION

Success and robustness of any pattern recognition task is primarily dependent on the defined features. They are supposed to capture effectively all the salient characteristics present in the input. Although, different features have been suggested for the task of PD pattern recognition in recent times, as yet there seems to be no agreement, regarding which of them is best suited. In this work, an effort is made to define and extract physical features from the pattern, which are what perhaps a human-expert may use while interpreting ϕ - q - n patterns. Features like, the physical location of the pulse clusters, its phase and magnitude spreads, average pulse-counts *etc.* are considered. Thus, authors intend to make the pattern recognition task performed by the computer and the human-expert, to some extent, similar.

In this work, the ϕ - q - n patterns are first compressed using the modified QPFIC algorithm. As already explained in Section 2, features are directly extracted from the transformations. To begin with, those transformations pertaining to the individual PD pulse clusters are to be identified and separated. This is achieved by employing the K-means clustering algorithm [17]. The input to the clustering algorithm are the transformations. After the process of clustering is complete, the average pulse-count value and the phase and magnitude spreads are extracted. It was verified, that the offset value available in the transformation does indeed closely match with the average pulse-count value of the individual pulse-clusters, and hence can be taken to represent the average pulse-count. All these values are normalized taking into account the maximum value of each one of them, and are put together to form the feature vector. The steps for feature extraction are summarized as follows.

1. Modified QPFIC algorithm is applied to the PD image and transformations are obtained.
2. Transformations are clustered using the K-means clustering algorithm, yielding information about each cluster and the corresponding transformations.
3. Transformations belonging to individual clusters are used to extract the following features: a_1, a_2 average pulse-count value of first and second cluster respectively; a_3, a_4 initial and final phase angle of first cluster; a_5, a_6 initial and final charge magnitude of first cluster; a_7, a_8 initial and final phase angle of second cluster; and a_9, a_{10} initial and final charge magnitude of second cluster

The values a_1 to a_{10} form the feature vector.

3.2 CLASSIFICATION OF PD SOURCES

The next task after feature extraction is to test the features for its ability and robustness to correctly classify different input patterns. Since the feature vector was 10-dimensional, it was not possible to use graphical means to demonstrate this capability. Although various classifiers are available, the authors preferred to choose the neural network (NN) classifier, for the simple reason that NN has been shown to possess abilities to learn any complex input-output mapping information, as required by the classification task, from mere examples. It is a well known fact that NN has been employed very successfully in a wide variety of practical applications, despite the existence of a number of theoretical issues questioning its very use. In addition, the authors are aware of a publication [18] where three NN architectures were compared and performance of back-propagation NN was found to be the best. It is also mentioned in [18] that NN might misclassify those patterns for which they are not trained. Therefore, NN was used for purposes of illustration alone, in particular to verify classification abilities possessed by the features.

The PD sources sought for classification are those in Table 2. Among the many learning algorithms, the back-propagation is popularly used to train the NN. The MATLAB NN toolbox was used for this purpose. No specific rules govern the choice of NN topology and architecture, which is known to depend on complexity of the input-output mapping to learn. The architecture used for classification was arrived at by trial and error, and it was found that $\sim 70\%$ of the trainings resulted in convergence. The input layer had ten neurons, the output had four neurons and the hidden layer had 10 neurons.

To generate additional training and testing patterns, noise was added to the existing PD images. This was done by adding a uniformly distributed random number (between 0 and 50) to those (ϕ, q) values for which $n > 0$. It is well established that no two PD patterns are identical, though they arise from the same source. To simulate these realistic variations in the PD patterns, arising due to various factors mentioned in Section 1, a random shift of $\pm 10^\circ$ in phase and ± 10 pC in magnitude was also introduced. The sum squared error during training was fixed at 0.001. During the testing phase, an output value ≥ 0.7 was considered as a correct classification. Twenty two testing patterns (not including the 12 training patterns), were used to ascertain the generalization ability of the trained NN. Among the converged networks, 80% were found to achieve a classification accuracy of 75% with the testing patterns, and the remaining could yield $\sim 100\%$ classification. This shows

that the architecture of NN chosen for this task is stable, and more importantly illustrates that the extracted features possess good classification abilities.

During simulations, input patterns were modified randomly to account for certain realistic changes. In addition to these, during practical PD measurements, it is often seen that more than one PD source is usually active, and the pattern data gathered is a cumulative effect of the individual contributions. Under such circumstances, an expert may use 'gating or windowing principles' (as in elliptic patterns), if there exists a phase or magnitude separability in the ϕ - q - n patterns. If such a separability exists, then with a background knowledge of the system, identification of individual PD sources can be achieved, though with some difficulty. The ability of the proposed method was gauged bearing this aspect in mind, however only to a limited extent. For this, two patterns belonging to different PD sources were merged. Features extracted from the resulting pattern was tested and found to possess the potential to address such tasks as well. It must be noted, that this aspect was examined at a primitive level, and much work needs to be done before one can comment on its true abilities. Since multiple PD source recognition is an important problem, mentioning this capability was considered appropriate.

4 CONCLUSIONS

THE importance of PD pattern compression was highlighted, and using the fractal image compression technique, it was shown how PD pattern compression could be combined with feature extraction, in a single step. This was possible, since the compressed data are stored as affine transformations, and from this relevant physical features could be extracted directly. Thereby, a separate and elaborate feature extraction operation was avoided. A modification of the QPFIC algorithm was implemented by defining a threshold, using which, only those transformations pertaining to PD pulse-clusters in ϕ - q - n patterns could be stored. This yielded an increase in compression ratio, almost by a factor of 1.5 to 2. The extracted features were verified for their pattern classification abilities, using a feedforward NN. The results obtained demonstrate the suitability of fractal image compression technique, not only for PD pattern compression but also for classification of PD sources.

ACKNOWLEDGMENT

The authors sincerely thank Prof. Gulski, of TU Delft, Netherlands, for providing them the PD data used in this work. Details of the exper-

imental setup used has been described in an earlier paper [16].

REFERENCES

- [1] F. H. Kreuger, *Partial Discharge Detection in High Voltage Equipment*, Butterworths 1989.
- [2] CIGRE WG 21-03, "Recognition of Discharges", *Electra*, Vol. 11, pp. 61-98, 1969.
- [3] R. Patsch and M. Hoof, "Pulse-sequence-analysis: A way to get a better insight into the physics of discharges", pp. 49-51, *IEE Int. Conf. on PD*, Canterbury, UK, 28-30 Sept. 1993.
- [4] L. Satish and B. I. Gururaj, "Performance of Three AI Paradigms for Recognition of Actual PD Patterns", *Japan, paper No. 62.11*, pp. 161-164, 8th ISH Conf., Aug. 23-27, 1993.
- [5] E. Gulski and A. Krivda, "NN as a Tool for Recognition of PD", *IEEE Transactions on Electrical Insulation*, Vol. 28, No. 6, pp. 984-1001, Dec. 1993.
- [6] E. Gulski and F. H. Kreuger, "Computer-aided Recognition of Discharge Sources", *IEEE Transactions on Electrical Insulation*, Vol. 27, No. 1, pp. 82-92, Feb. 1992.
- [7] A. Groz and H. G. Kranz, "Possibilities of Diagnostic Methods with Different Neural Networks and Fuzzy Sets for PD-Measurements" pp. 165-168, 8th ISH Conf., Aug. 23-27, Japan, 1993.
- [8] L. Satish and W. S. Zaengl, "Can Fractal Features be used for Recognizing 3-d Partial Discharge Patterns?", *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, No. 3, pp. 352-359, June 1995.
- [9] L. Satish and B. I. Gururaj, "Use of Hidden Markov Models for Partial Discharge Pattern Classification", *IEEE Transactions on Electrical Insulation*, Vol. 28, pp. 172-182, Apr. 1993.
- [10] T. Hucker and H. G. Kranz, "Requirements of Automated PD Diagnosis Systems for Fault Identification in Noisy Conditions", *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, No. 3, pp. 544-556, Aug. 1995.
- [11] Y. Fisher, D. Rogovin and T. P. Shen, "A Comparison of Fractal Methods with DCT and Wavelets", available at <http://www.inl3.ucsd.edu/y/Fractals/#papers.html>
- [12] H. O. Peitgen and D. Saupe (Ed), *The Science of Fractal Images*, Springer-Verlag, 1988.
- [13] M. F. Bamsley and A. D. Sloan, "A Better Way to Compress Images", *Byte Magazine*, pp. 215-223, January 1988.
- [14] A. E. Jacquin, "Image Coding Based on a Fractal theory of Iterated Contractive Image Transformations", *IEEE Transactions on Image Processing*, Vol. 1, No. 1, pp. 18-30, January 1992.
- [15] Yuval Fisher (Editor), *Fractal Image Compression - Theory and Applications*, Springer-Verlag NY Inc., 1995.
- [16] A. Krivda, E. Gulski, L. Satish and W. S. Zaengl, "The Use of Fractal Features for Recognition of 3-D Discharge Patterns", *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, No. 5, pp. 889-892, October 1995.
- [17] R. O. Duda and E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1973.
- [18] A. Krivda, "Automated Recognition of Partial Discharges", *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, No. 5, pp. 796-821, October 1995.

Manuscript was received on 24 June 1997, in final form 18 May 1998.