A Method for Digital Image Compression with IDP Based on Gaussian Radial Basis Function NNs

Noha A. Hikal[1], Roumen Kountchev[2], R.M El-Awady[3], A.A Shaalan[4]

Abstract – **In this paper a new algorithm for still image compression based on Gaussian Radial Basis Function NNs in correspondence with Inverse Difference Pyramid (IDP) decomposition is represented. The new algorithm is well suited to be used in Progressive Image Transmission (PIT). The method advantages relay on the learning process and on the adaptation capability of the NNs to reduce the matrices computation complexity, the total number of pyramid levels required for PIT, and to maximize the PSNR. In addition to, for image reconstruction no interpolation is needed for the improvement of the reconstructed image quality.**
Keywords – **IDP Decomposition, PIT. Neural Networks, Gaussian Radial Basis Function.**

I. INTRODUCTION

The image compression is an important tool to store and transmit visual information used for multiple applications. The image compression refers to a process in which the amount of data used to represent an image is reduced to meet a bit rate requirement (below or at most equal to the maximum available bit rate), while the quality of the reconstructed image satisfies a requirement for a certain application and the complexity of the computation involved is affordable for the application. The progressive Image Transmission (PIT) [1] concept is of particular importance in browsing large image files. Hence, it permits the initial reconstruction of the image approximation and gives the viewer an early impression of image contents, followed by gradual quality improvement of the reconstructed image. The observer may terminate the transmission of an image as soon as its contents are recognized. In order to send the image data progressively, it should be organized hierarchically in the order of importance, from the global image characteristics to its local details. There are two types of data structures for progressive transmission depending on the encoding method employed [2]: (i) Transform-based encoding, (ii) Spatial encoding. In the transform-based encoding, the image is first divided into a set of contiguous non-overlapping blocks, and then each block is transformed into a set of transform coefficients, (e.g, Discrete Cosine Transform (DCT). On the other hand the spatial approach, like the pyramid-level resolutions, the image is successively reduced in spatial resolution and size by sub-sampling or averaging. The image approximation is obtained using a single frame or a combination of frames representing the image. Therefore sending a set of image frames in pyramid form from top to bottom level naturally constitutes a progressive transmission.

[1]N. A. Hikal is with Special Studies Academy Cairo, Eypt, Eng_noha_hikal@yahoo.com
[2]Roumen Kountchev is with FCTT, Radiocomm. Dept. Technical University of Sofia, K1. Ohridsky 8, Sofia, Bulgaria, rkountch@tu-sofia.bg
[3]R. M El-Awady is with Faculty of Engineering, Comm. Dept., El-Mansoura University, Egypt
[4]A. A Shaalan is with Faculty of Engineering, Comm. Dept., El-Zagaaig University, Egypt, drshaalan2006@yahoo.com

II. PYRAMIDAL IMAGE REPRESENTATION

The first pyramidal data structure is the Gaussian-Laplacian pyramid (GP/LP) [3]. The GP can be viewed as a set of low pass filtered copies of the original image, while in the LP each level is a difference between two successive levels of the GP. Various pyramid data structures for PIT have been proposed like: Mean Pyramid, Reduced Sum Pyramid, Reduced Difference Pyramid, Hierarchy Embedded Differential Image, etc [4,5,6]. Many pyramidal decomposition techniques showed improvements over the JPEG standard [7]. The Inverse Pyramidal Decomposition, based on Discrete Cosine Transform (IDP/DCT), was proposed in [8]. The IDP decomposition differs in the way of obtaining the pyramid levels, and the word "inverse" refers to the requirement to compute the pyramid levels from pyramid top (level zero) to the bottom. The coding/decoding processes are simple and flexible. Therefore, it is suitable for real time processing and serves PIT requirements. NNs are interesting alternatives to classical image compression techniques due to their lower matrices computational complexity, adaptation and learning capability. In [6,9] two algorithms for image compression based on IDP using BPLA and 2D-SOM VQ NNs, respectively, were presented. The algorithm showed superiority over IDP/DCT in terms of PSNR, CR, and the total number of levels required for the image reconstructing at the receiver side. In this work, a new technique for image compression based on merging IDP and GRBF NNs is presented, combining the advantages of IDP and NNs methods. The Gaussian networks are a special RBF type, characterized by being highly nonlinear and providing good locality for incremental learning. This paper is organized as follows: Section (3) introduces a definition for the basic GRBF NNs. Section (4) explains the new developed algorithm for image compression based on IDP-GRBF NNs. Sections (5) and (6) represent the performance measurements criteria and the simulation results respectively. Finally Section (7) gives a conclusion on the results of the new algorithm.

III. BASIC GAUSSIAN RBF NNS MODEL

The Gaussian RBFNNs [10] is the most important RBFN class. The Basic model employs for its RBF an un-normalized form of Gaussian density function presented as:

$$\phi(r) = \exp\left(-\left(\frac{r}{\sigma}\right)^2\right) \qquad (1)$$

It has a peak at the center r=0 and decreases monotonically as the distance from the center increases. The basic Gaussian network, Fig.1, consists of a three stages network [11] with input, intermediate and output stage. The intermediate stage consists of an array of nodes c_i that contain center vectors. These nodes calculate the Euclidean distance between the centers and the network input vector and the results are passed through a Gaussian function. The output stage of the neuron is a linear combiner used to obtain the weighted summation of the intermediate outputs. Let X ($x \in R^n$) and Y ($y \in R^m$) be the n-input and the m-output of the network, respectively, and u ($u \in R^l$) be the l outputs of the l Gaussian units. A Gaussian radial basis function Φ_i is assumed to be the same for all units. The connections between the nodes in the hidden layer and the j^{th} output (y_j; here: j=1,2…m) through the linear combiner unit are shown in Figure 1 [11].

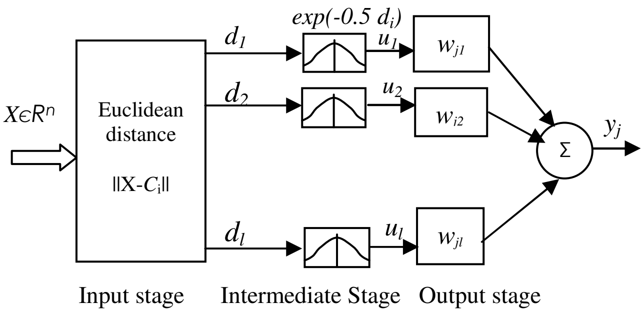


Fig. 1. Block diagram of a Gaussain RBF

Here the term d denotes the difference computed as:

$$d_i = \sum_{k=1}^{l} [(x_i - c_{ik}) / \sigma_{ik}^{2}] \tag{2}$$

and σ^2 is the variance controlling the width of the Gaussian function; u_i is the output of the i^{th} hidden Gaussian neuron, presented with the relation [11]:

$$u_i = \exp\left(-\frac{1}{2} \sum_{k=1}^{n} \left[\frac{x_k - c_{ik}}{\sigma_{ik}} \right]^2 \right), \quad 1 \le i \le l \tag{3}$$

Therefore, the input-output relationship of a GRBFNN with multiple outputs y_j ,$1 \le j \le m$,can be described mathematically by the following equation:

$$\begin{aligned} y_j &= \sum_{i=1}^{l} w_{ji} u_i \\ &= \sum_{i=1}^{l} w_{ij} \exp\left(-\frac{1}{2} \sum_{k=1}^{n} \left[\frac{x_k - c_{ik}}{\sigma_{ik}} \right]^2 \right) \end{aligned} \tag{4}$$

With the centers and variance parameters that are initialized to nearly optimum values, the learning task (which is described by the input/output data pairs $\{x, y_d\}$) will follow the gradient-descent method to form updating equation for the unknown weights. Then the instantaneous value of the cost function, which should be minimized ill be:

$$E = \frac{1}{2} \sum_{j=1}^{m} (y_{dj} - y_j)^2 = \frac{1}{2} \sum_{j=1}^{m} e_j^2 \tag{5}$$

Where:

$$\begin{aligned} e_j &= y_{dj} - y_j \\ &= y_{dj} - \sum_{i=1}^{l} w_{ij} \exp\left(-\frac{1}{2} \sum_{p=1}^{n} \left[\frac{x_k - c_{ip}}{\sigma_{ip}} \right]^2 \right) \end{aligned} \tag{6}$$

Using both the above definitions and the network equation (4), the weight updating equation used to minimize the cost function could be derived as follows [11]:

$$w_{ij}^{new} = w_{ij}^{old} + \eta u_i e_j \tag{7}$$

Where η is positive learning rate constant which controls the learning rate, convergence, and stability.

IV. Image Compression Algorithm Using GRBF NNs Based on IDP Decomposition

The basic steps for image compression using GRBF NNs based on IDP coding/decoding process are introduced in the following two subsections.

Coding process

Step 1: For the first pyramid level (P=0), the whole image B(i,j) of size 2^nx2^n, $1 \le i, j \le 2^n$, is divided into a number of l blocks, each of them of size (m x m) pixels.

Step 2: A GRBF network will be created with an input vector of length m^2x1; an intermediate stage of the Gaussian network consisting of an array of center vectors c_i (i=1,2,.....l; total number of blocks in the whole image), the RBF center vectors are set to the values of the pixels corresponding to each block. The total l outputs from the intermediate stages feed the Linear Combiner (LC) stage to be multiplied m times by the randomly initialized weight vectors $W_{j.} = [w_{j1}\ w_{j2} ....w_{jl}]^T$; $1 \le j \le m$. The output vector y of length mx1 will be compared with the desired output vector y_d (mx1), which is set to the average vector of each block. Fig. 2 shows a block diagram for a GRBF network for block compression.

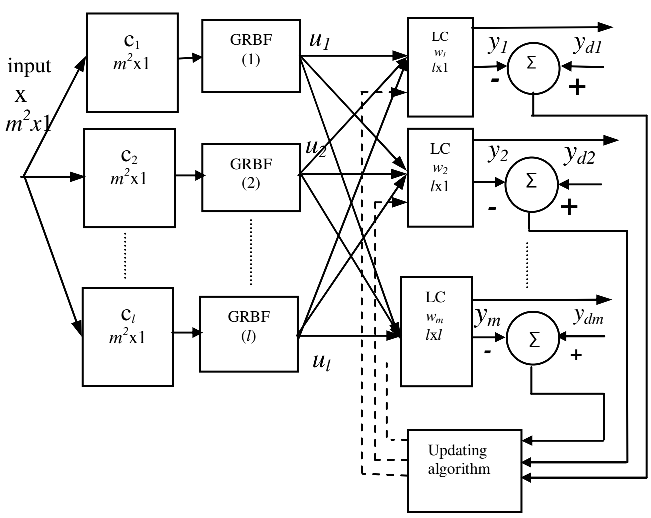


Fig. 2. GRBF network for block compression

Step 3: Applying the training input–output pairs $\{x, y_d\}$ (which are the image blocks themselves) to update the weights of the network. Once the training is complete, the image compression is demonstrated in the recall phase, using row scanning, pixels of each block are arranged as a vector of length (m^2) and these (m^2) components are considered to be the input vector of the GRBF-NN. The obtained weights matrix (lxm) forms the coefficients of the first pyramid level (p=0). The weights matrices will be quantized, encoded and transmitted later. So instead of sending the whole image of size 2^nx2^n, only the weights matrix of size (lxm) per block will be send. For example, if the original image of size 128x128 was divided into 4 blocks each of size 64x64, after compression using the first level of the GRBF-NN, only 4x64 coefficients per block will be sent, this is corresponding to compression ratio of (128x128)/(4x64x4)=16, for further compression encoding techniques may be applied. This form of reduction in the number of pixels is considered a spatial reduction, which is used in PIT.

Step 4: At the transmitter side, the reconstructed image will be recovered from the weights matrices (after inverse quantization and decoding process) in reverse arrangement using input vector of size (lmx1), (m) intermediate stages, output vector of length (m^2) in Fig. 3, the center vectors are set to the weight matrices corresponding to the original blocks and the desired output vector is representing the original pixels of the corresponding original input block. After training, the network reaches stability and a reconstructed image of same size, as the original will be obtained. Pixel by pixel difference is calculated between the original image and the reconstructed image, which results in a difference image $E_0(i, j)$, $1 \le i, j \le 2^n$, of same size as the original one 2^nx2^n.

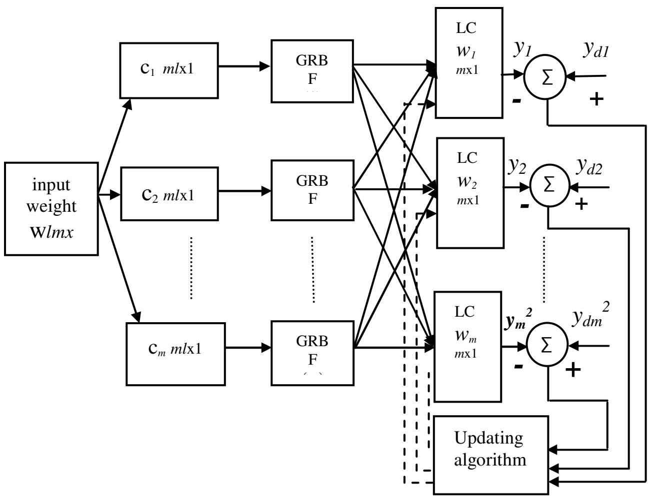


Fig. 3. GRBF NN for Block Reconstruction

Step 5: For the second pyramid level (P=1), starting with the difference image E_0, the whole difference image $E_0(i, j)$ of size 2^nx2^n is divided into a number of ($4l$) blocks, each of them of size ($m/2$x$m/2$) pixels. A GRBF network will be created with an input vector of length $(m/2)^2$x1, an intermediate stage consists of an array of centers c_i (i=1,2,...,4l), the RBF center vectors are set to the values of the pixels corresponding to each block. The output stage consists of $m/2$ weight vectors each weight vector $W_{i\cdot}=[w_{i1}\ w_{i2}.......w_{4il}]^T$. The output vector y of length $m/2$x1 will be compared with the desired output vector y_d ($m/2$x1), which is set to the average vector of each block.

Step 6: Applying the training input–output pairs, and following the same procedure of Step (3), considering the new vectors dimensions. The obtained weight matrices (4lx$m/2$) per block form the coefficients of the second pyramid level (p=1). The weights will be quantizated, encoded and transmitted.

Step 7: Following the same reconstruction procedure as Step (4), considering the new dimensions, a reconstructed image of the same size, as the original will be obtained. Pixel by pixel difference is calculated between the difference image E_0 and the reconstructed difference image E_0', results in a new difference image $E_1(i,j)$, $1 \le i, j \le 2^n$, of the same size as the original one 2^nx2^n.

Step 8: For the remaining pyramid levels, starting from Step (5) and following the same procedure to obtain the compressed coefficients, these coefficients are used at the transmitter side to reconstruct the difference image E_p' and to obtain new difference image E_p to be used for the next pyramid level. The stopping criterion here is when the minimum cost function is obtained in accordance with Eq.(6).

Decoding process

Step 1: For each level, the process of decoding the received weight matrices and de-quantization has to be done.

Step 2: The reconstructed difference image is obtained using the same arrangement of block reconstruction as the one, used at the transmitter side for each level.

Step 3: The elements $\hat{B}(i,j)$ of the restored image are calculated in accumulation way [6,8,9], which can be expressed mathematically as follows:

$$\hat{B}(i,j)=\sum_{p=0}^{P-1}\tilde{B}(i,j) \tag{8}$$

Where i, j = 1,2, . . , 2^n and $\tilde{B}(i,j)$ is the reconstructed image using the GRBFNN at each level of the receiver side.

V. PERFORMANCE EVALUATION OF THE IDP-GRBF

In order to evaluate the performance of the proposed BPNN-IDP algorithm, the commonly known measures will be used:

(i) The Peak Signal to Noise Ratio (PSNR) obtained for the reconstructed image at each level p of the pyramid as:

$$PSNR(p)=10\log_{10}\frac{B_{max}^2}{\bar{\varepsilon}^2(p)},\text{dB} \tag{9}$$

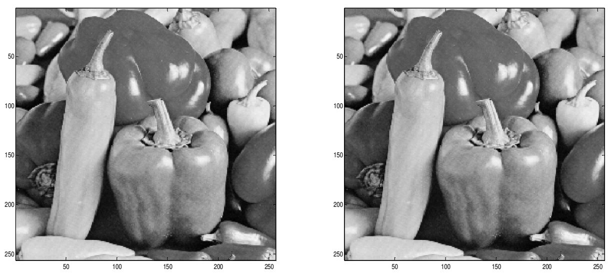
Where $\bar{\varepsilon}^2(p)$ is the mean square error (MSE) at level p:

$$\bar{\varepsilon}^2(p)=4^{-n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}[B(i,j)-\hat{B}_p(i,j)]^2 \tag{10}$$

(ii) The image compression is calculated as the ratio of the total number of bits transmitted to the total number of pixels in the original image.

VI. Simulation Results

A MATLAB Version 6.5, computer simulation program was designed to simulate the IDP-GRBF depending on the NNs toolbox. Experiments have been performed on grayscale and color images as well. For color images, the described algorithm can be performed applying it on the matrix of every primary color component: R, G, B. In order to obtain higher compression ratio, the R, G, B components of every pixel (i,j) were transformed in Y, Cr, Cb format. For a 512x512 pixel test images, the first level was of size 4x4 blocks (128x128 pixel), while for 256x256 pixels test images the first level was 2x2 blocks (128x128 pixels) and continued as described in section IV. The results for PSNR, CR, and number of levels for a number of test images were reported and compared. Fig. 5a shows two consequent pyramid levels for “Peppers”, Fig.5b represents the first level for “crosses” and “circles” test images. Table 1 shows a comparative result of the developed algorithm with the standard JPEG 2000 for the same compression ratio (results of JPEG are obtained with Lura Smart Compress software); the results obtained from the developed algorithm can be further compressed by applying encoding techniques.



CR=0.125 bpp, PSNR=32.4 dB   CR=1 bpp, PSNR=34.2 dB

CR=0.28 bpp, PSNR=28.9 dB   CR=0.28 bpp, PSNR=28.3 dB

Fig. 5. a) Two consecutive pyramid levels for reconstructed test image “Peppers”, 512x512 pixels; b) Test images “Circles” and “Crosses”, 256x256 pixels

VII. Conclusion

The novelty in this study was to develop a new pyramidal scheme based on IDP decomposition lies in modeling each pyramid level using GRBF NNs. This new algorithm can be compared with the most similar pyramid proposed in articles concerned with IDP. It can be underlined that, compared to IDP-DCT decomposition, IDP-GRBF reduces the number of levels, required to reconstruct the image at the receiver side and increases the PSNR, and CR. The coding and decoding of the hidden weights values are relatively simple.

TABLE 1. Comparison between JPEG 2000 standard and IDP-GRBFN for same compression ratio.

Test		CR	JPEG2000 PSNR (dB)	GRBF PSNR (dB)
Lena	7.1	1.13	38.4	33.2
Barbra	7.1	1.13	38.95	31.01
Peppers	7.1	1.13	40.04	34.2
Text	7.1	1.13	28.5	28.09
Circles	28.4	0.28	33.3	28.98
Crosses	28.4	0.28	30.54	28.33

The computation does not contain complexities compared with transformation methods and the process can be speeded-up by using the pre-defined parameters; also the quality of the reconstructed image obtained from the new method after a few number of levels used is considerable. The quality of the image obtained from the IDP-BPLA [10] and the IDP-GRBF is almost the same while the developed algorithm has the superiority in learning time, in addition to no probability to stick in local minima. The main drawback of the developed method is that to reconstruct the image at the receiver side, it requires the receiver to have a prior knowledge about the received image to start the supervised learning algorithm.

VIII. Acknowledgement

We would like to acknowledge the financial support of the NSF of Bulgaria (Contract ВУ-МИ-104), and the Special Studies Academy at Cairo.

References

[1] K.Tzou, "Progressive Image Transmission: A review and comparison of techniques" *Opt. Eng.,Vol 26, pp.581-589, July 1987.*

[2] A. Poularikas. "The Transforms and Applications Handbook". *CRC press 2000.*

[3] P. Burt. "The Laplacian Pyramid as a Compact Image Code". *IEEE Trans. Commun., Vol 31, No. 4, April 1983.*

[4] M. Goldberg. "Comparative Performance of Pyramid Data Structures for Progressive Image Transmission*", IEEE Trans. commun. Vol. 39, No.4, April 1991.*

[5] W. Kim, et al. "Hierarchy Embedded Differential Image for Progressive Transmission using Lossless Compression" *IEEE Trans. Circints and systems for Video Technology. Vol. 5, No.1, Feb. 1995.*

[6] N. Hikal, R. Kountchev, ”2D-SOFM Vector Quantization for Image Compression based on Inverse Difference Pyramidal Decomposition”, *Proceeding of TELSIKS, Sept. 2005.*

[7] W. Kou, ”Image compression Algorithms and Standards”. *Kluwer Academic Publishers, Boston, 1995.*

[8] R. Kountchev, et al. "Inverse Pyramidal Decomposition with Multiple DCT". *Signal Processing. Image communication 17 (2002), pp. 201-218.*

[9] N. A. Hikal, R. Kountchev. ”BPNN for image compression based on IDP Decomposition”. *Proceeding of ICEST, June 2005, pp. 15-18.*

[10] Y. H. Hu, J. N. Hwang. "Handbook of Neural Networks and Signal Processing”, *CRC Press 2002.*