ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645

Image Pattern Extraction and Compression using Pixel Neighborhood and Weighted PCA Algorithm

Pranab garg and Dinesh Kumar

Department of Computer Science, Guru Kashi University, Sardulgarh Road, Talwandi Sabo, Bathinda – 151302, Punjab, India; garg.pranab@gmail.com, Kdinesh.gku@gmail.com

Abstract

Background/Objective: An image can be compressed by compressing the patterns in itself. The patterns preserve the maximum entropy in an image. If patterns are extracted using the pattern extraction algorithm and compressed using their features, then rest of the image may be compressed to a high degree of decompression. The patterns are therefore principal components of the image and principal component analysis can be applied over that in order to achieve an efficient compression algorithm. The principal component analysis algorithm works fine when compressing the input image and outputs a reasonably good compression ratio. However, the compression ratio becomes dependent upon the number of eigen values/vectors chosen to get compressed image. Method: In the presented work, a selection criterion for selecting the Eigen values/vectors is suggested. The criteria is based on threshold selection that is computed by using different techniques and then taking the arithmetic mean of all thresholds from all techniques. Also, a weighted Eigen values are used for threshold computation and compared with the statistical threshold. The Eigen values/vector less than the threshold are taken into consideration for image compression and a well optimized compression ratio is obtained. Finding: The computation time may be improved by decomposing the input image using haar wavelet transform. The different frequency sub bands consist of low frequency (LL), high to low (HL), low to high (LH) and high (HH) frequency sub bands. As the LL sub band contains maximum information out of the four bands, the principal component algorithm is applied in this band only for compression.

Keywords: Compression, Eigen Value, Eigen Vector, Otsu Algorithm, Segmentation, Threshold

1. Introduction

Visuals of a scene are more elaborative rather than its textual description as a universal fact. At the same time, the visuals or images consumes more space than their respective textual content. Therefore, there is a trade off between the choice of preserving the information either in form of image or text on system hard disc. In order to reduce the images storing area on system/memory disc, the images need to be compressed at storage time and decompressed at the time of usage. A lot of compression-decompression algorithms are available for the said task. In the presented work, principal component analysis algorithm is presented for image compression and decompression. The algorithm works on the principal of

computation of eigen values/vectors for the input image and a threshold is computed for selecting the eigen values/vectors. The thresholded eigen values/vectors are used to store the image information. Inversely, the eigen values/vectors are used to get back the input matrix i.e. image as a decompression process.

The pattern extraction is an important procedural step in compressing the image using the patterns in itself. Patterns can be identified or extracted by first binarizing the image using a threshold value followed by segmentation process. The thresholding may be performed using the Otsu algorithm. The segmentation is performed using pixel neighbourhood algorithm using Euclidean distance criteria.

^{*}Author for correspondence

2. Related Works

A neural network approach was presented in this paper for image compression. A loss less image compression was achieved at reasonable compression ratio with improved psnr and mse. Neural network is a very efficient method for image compression. It suites to image compression on account of massively parallel and distributed architecture¹.

Two extended-PCA, one block by block and other block to row algorithm was discussed to manipulate the block information of the image. The results were tested and compared with existing algorithms².

Hybrid of PCA and DCT based approach was discussed here for loss less image compression³.

A memory assisted compression technique was presented here for image compression. Rapid growth of emerging medical applications such as health and tele-medicine requires loss less access to medical images and data over band limited channels⁴.

Principal Component Analysis (PCA) enables a fair independence and data compaction for Gaussian sources. Yet, non-linear generalizations of Principal component algorithm can provide better performance for more realistic non-Gaussian sources⁵.

Transform-based lossy compression had a huge potential for hyper spectral (HS) data reduction. Since hyper spectral images were highly coherent among each spectral band and in particular across neighboring frequency bands, the choice of a spectral decorrelation method that allowed to retain as much information content as possible was desirable⁶.

3. Image Decomposition using haar Wavelet

The colored input image in RGB format or jpeg format is converted into gray color format using the following color transformation:

$$Gray = 0.30310 * R + 0.5870*G + 0.1140*B$$

The R, G and B denotes the pixel intensities in red, green and blue color component of the input image. The gray level intensities ranges from 0 to 255, where 0 being the black and 255 as white and in between are black to white gray shades. The gray image is then decomposed using the haar wavelet transform (HWT) and is implemented by using the following matrix transformation:

$$HWT = \frac{1}{\sqrt{2}} \begin{bmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{bmatrix}$$
 (1)

If the input image size is NxN, then the sub band images are of $(N/2 \times N/2)$ size. N should be multiple of power of 2 for proper HWT decomposition of gray image. The decomposed sub band images are ordered as follows:

$$HWT = \begin{bmatrix} LL & HL \\ LH & HH \end{bmatrix}$$
 (2)

4. Eigen Values/Eigen Vectors Computation

By virtue of maximum information entropy in LL sub band image, the LL sub band is centred by computing the mean of the image intensity and then subtracting each pixel gray value from mean gray value.

$$Meanl = \frac{1}{row \ x \ col} \sum_{r=1}^{row} \sum_{c=1}^{col} I(r, c)$$

Centered Image
$$CI(\mathbf{r}, \mathbf{c}) = \sum_{r=1}^{row} \sum_{c=1}^{col} (MeanI - I(r, c))$$

In the next step, image covariance is computed as follows:

$$CovImg = CI_{(r,c)} * CI_{(r,c)}^{T}$$

Where $CI_{(r,c)}^{\ \ T}$ is the transpose of the matrix $CI_{(r,c)}$.

Next, the eigen values/vectors are calculated as given below:

$$A.V = \lambda.V$$

Where, A is the Gray Image matrix of size m x m

V is m x 1 non-zero vector (Eigen Vector) and Λ are scalar (Eigen Values)

The solution for λ are the Eigen value of A and the respective vector V as Eigen vector of A.

$$A.V = \lambda.V$$

$$A.V - \lambda.I.V = 0$$

$$(A - \lambda.I).V = 0$$

A threshold is computed for eigen value selection. The thresholded eigen values are called principal components of the image. Using the principal components, the image is transformed into its Eigen vectors as given below:

Image =
$$\begin{bmatrix} Eigen\ Vector - 1 \\ Eigen\ Vector - 2 \\ Eigen\ Vector - 3 \\ ... \end{bmatrix}$$

Highest Eigen vector images are used as principal image components and that is the compressed image.

5. Threshold Computation

The Optimum threshold for optimum compression ratio is computed by taking arithmetic mean of thresholds computed by using Otsu algorithm, mean of all Eigen values, median Eigen Value and mode of Eigen value.

$$Threshold_1 = \frac{T_{Otsu} + T_{Mean} + T_{Median} + T_{Mode}}{4}$$

 $T_{
m Otsu}$ is computed by using the minimum within class variance and is given by:

$$\sigma_{\omega}^{2}(t) = \omega_{1}(t)\sigma_{1}^{2}(t) + z_{2}(t)\sigma_{2}^{2}(t)$$

Where ω_i are the probabilities of the two classes separated by a threshold 't' and are variances of these classes.

 $T_{\rm Mean},\,T_{\rm Median}$ and $T_{\rm Mode}$ are mean, median and mode of eigen values array. The histogram of Eigen values may be generated by converting the Eigen values to nearest whole integer values.

The selected Eigen vectors are now arranged to get the compressed LL sub-band image. The inverse wavelet transform is applied to get back the final compressed image.

6. Weighted Threshold Algorithm

In case of weighted threshold, the Eigen values are arranged in descending order. Based on no. of Eigen values, a weight vector is generated in following way:

Say N is no. of Eigen values $(\square_1, \square_2, \square_3, \dots \square_N)$, then, the weight vector (W) is given by:

$$(W]_1, W_2, W_3, ... W_N$$

Also,
$$W_1 + W_2 + W_3 + ... + W_N = 1$$

Weight W. is given by:

$$W_i = \frac{N - (i - 1)}{\sum_{i=1}^{N} N_i}$$

The weighted threshold is computed by the followings:

$$Threshold_2 = \frac{1}{N} \sum_{i=1}^{N} W_i \mathbf{A}_i$$

7. Pattern Extraction and Segmentation

For pattern extraction, the input image is first converted into gray scale format. Otsu algorithm is applied over the gray image in order to compute the threshold value. Based on the computed threshold value, the gray image is binarized. The binary image is then exposed to segmentation algorithm. The segmentation algorithm articulates the different patterns into different frames for feature vector generation. The different patterns may be exposed to features vector extraction algorithm so that the patterns are stored using their features and not their image value. This gives the image compression a great value as the compressing is in form of numerical features and not as pixel's attributes. Otsu algorithm works on the principal of minimum within class variance comparison. Below figure shows the results of above mentioned steps as original, gray and binary images.

8. Quality Metrics for Performance Evaluation

The image compression algorithm can be best evaluated based on certain performance indices as listed below:

• Peak-Signal-to-Noise Ratio (PSNR):

$$\frac{1}{MxN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [(f(i,j) - f'(i,j)]^2)]$$

Where, $M \times N$, f(i,j) and f'(i,j) represent the size, gray-level pixel values of the original and reconstructed images, respectively.

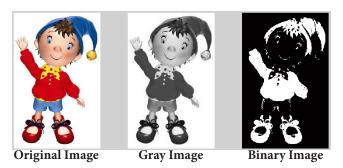


Figure 1. Original, Gray and Binary Images.

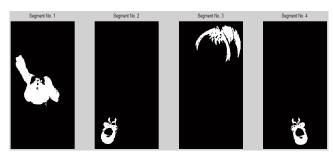


Figure 2. Segmented Images using Pixel Neighborhood Algorithm.

Table 1. MSE and PSNR with Different Technique

Algorithm	MSE	PSNR
Fig. 3	0.48	51.35
Fig. 4	0.45	51.44
Fig. 5	0.50	51.18
Fig. 6	0.44	51.73

Table 2. MSE and PSNR with Proposed Techniques for different figures

•		
Algorithm	MSE	PSNR
DCT	0.67	49.92
DWT	12.66	35.46
JPEG	7.13	39.59
PCA	2.12	44.91
Proposed (Threshold ₁)	0.48	51.35
Proposed (Threshold ₂)	0.47	51.44









Figures 3 to 6. Original Standard Test Gray Images.

• Standard Deviation (SD) →

$$\sigma^2 = \frac{1}{\mathbf{M}\mathbf{x}\mathbf{N}} \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{M}} \square \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{N}} \square \left(\mathbf{x}(\mathbf{i},\mathbf{j}) - \mu\right)^2$$

• Entropy (E)

$$H = -\sum_{i=0}^{L-1} P_i \log P_i$$

Where L and P_i are the grey level intensity and the ratio between the number of pixels of i_{th} grey level intensity and the total pixel number contained in an image respectively.

9. Results

The proposed algorithm has been tested over a no. of images including the standard images. The results for the standard images are compiled in the table.

The PSNR and MSE are at par at threshold-1 and 2 and are improved as compare with other algorithms as tabulated. The table- 1 and table-2 show the SD and entropy for the four standard figures 3 to 6, using the proposed algorithm:

10. Conclusion

The presented algorithm for image compression using discrete wavelet transform and followed by weighted principal component algorithm proves to be an efficient algorithm in terms of the PSNR and MSE with respect to compression ratio. The algorithm is loss less algorithm for image compression with fast implementation as the algorithm primarily targets on decomposed image that is already one quarter in size to that of the original image. The future work may include the algorithm efficiency on colored image compression keeping target of reducing the time of operation.

11. References

- Fang Tao, Lu Jiangang, Wang Zhi, and Sun Youxian. An Image Compressing Algorithm based on PCA/SOFMHybrid Neural Network. IEEE. 2003; p. 2103-7, 0-7803-7906-3/03.
- Lim ST, Yap DFW, Manap NA. Medical Image Compression Using Block based PCA Algorithm. Malaysia: IEEE 2014, ICCCCT. 2014; p. 171–76.

- 3. Huber-Lerner Merav, Hadar Ofer, Stanley R. Rotman and Huber-Shalem Revital. Compression of Hyperspectral Images Containing a Sub pixel Target. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2014 June; 7(6).
- Hesabi Zhinoos Razavi, Sardari Mohsen, Beirami Ahmad, Fekri Faramarz, Deriche Mohamed, and Navarro Antonio. A memory-assisted lossless compression algorithm for medical images. IEEE. 2014; p. 2030-38, 978-1-4799-2893-4/14.
- 5. Amrani N, Laparra V, Camps-Valls G, Serra-Sagrista J and Malo J. Loss less coding of hyperspectral images with principal polynomial analysis. IEEE. 2014; p. 4023-30, 978-1-4799-5751-4/14/.
- Licciardi GA, Chanussot J, Piscini A. Spectral compression of hyperspectral images by means of principal component analysis decorrelation. IEEE. 2014; p. 5092-98, 978-1-4799-5751-4/14/\$31.00.
- Saira Banu J, Babu Rajasekhara, Pandey Reeta. Parallel Implementation of Singular Value Decomposition (SVD) in Image Compression using Open Mp and Sparse Matrix Representation. Indian Journal of Science and Technology. 2015 July; 8(13). DOI: 10.17485/ijst/2015/v8i13/59410.
- 8. Rajeswari Joe J, Rama N. Neural Network based Image Compression for Memory Consumption in Cloud Environment. Indian Journal of Science and Technology. 2015 July; 8(15). DOI: 10.17485/ijst/2015/v8i15/73855.