International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-8,November 2015
ISSN: 2395-3470
www.ijseas.com

CRYPTOGRAPHY AND STEGANOGRAPHY IMPLEMENTATION ON IMAGE COMPRESSION

Khushboo[1],Parveen[2]
[1]NCCE,Panipat,India
[2]NCCE,Panipat,India

ABSTRACT

The fractal image compression is advanced technique to compress the image effectively by finding the similar portion in the image. The image compression can be extended to hide the data, it may confuse the unintended user to secure the data. Moreover, the data can be encoded by using the cryptography before hiding it inside the image. This paper defines a technique to hide the data the encoded data inside the compressed image. The simulation shows the effectiveness of the technique.

Keywords: Cryptography, FIC, APCC, Steganography

1. INTRODUCTION

Image compression is acquiring more concentration regularly like more rate compaction and excellent quality of picture are in more order[1]. A benefit of a picture compaction is to eliminate the time which is acquired for transmittance of a picture. An exemplification is that a picture has 512 columns and 512 rows. A deprived of compaction, entirely 512 x 512 x 8 = 2,097,152 bits information required to be saved. Every pixel is denoted by 8-bit figures structure. At present to compact or can say to eliminate the various bits required to save that bits deprived of losing the quality of a picture. The complete compaction-decompaction process is just shown in Fig 1.1 Image compaction is a difficulty of eliminating the quantity of information needed to depict a digital image. It is a procedure proposed to defer a compressed depiction of a figure, hereby eliminating the image storage space/communication necessity. The decrease in image dimension permits many images to be saved in specified quantity of memory space. Moreover, it also decreases the time needed for image to be send on the net or download through the net pages. It is as well as more applicable in communication satellite to minimize the transference time. compaction can be attained by the elimination of single or more of the three simple data abundances [2].

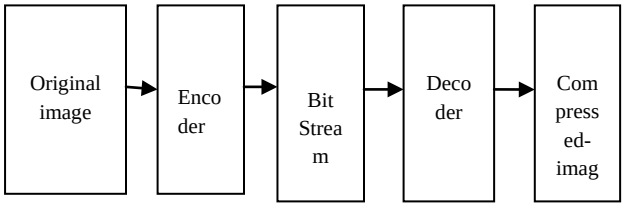

Fig. 1: The Basic Flow Chart of Image Compression Coding [2]

The primary goal of image compaction is to determine an image depiction in that pixels are not so much correlated. The two basic rules utilized in image compaction are unnecessariness and irrelevancy. Different types of redundancy: **Coding redundancy** Coding redundancy is available whenever fewer than optimum code words are utilized. A code is a structure of symbols and it is used to signify a body of information or group of actions. Every part of events or information is allocated a series of code symbols, known as code word. The Length of code words is determined by the number of symbols. **Inter pixel redundancy** Inter pixel abundance outcomes from interrelationship among pixel of a picture. Considering that the importance of some precondition pixel can be reasonable forecasted through the value of its neighbors, the information approved by separate single pixels is quite little. **Psycho visual redundancy** Psycho visual redundancy arises as human visual system ignores the data. By removing, this type of data human eye is not able to identify the segregation of a unique image data. To eliminate psycho visual redundancy they can utilize quantize. Because the elimination of psycho visually redundant data gives a loss of quantitative information. The compression technique reduces the size of data, which in turns requires less bandwidth and less transmission time and related cost. There are many algorithms developed for

the data compression using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) [2].

2 Fractal Image Compression

Fractal image compression is a relatively recent image compression method which exploits similarities in different parts of the image. Fractal Image compression (FIC) is one among the compression techniques in the spatial domain which exploits similarities in different parts of the image. One can see the self similar regions in the image above. Fractal compression stores this type of information to achieve compression. To do Fractal compression, the image is divided into sub blocks. Then for each block, the most similar block if found in a half size version of image and stored. Then during decompression, the opposite is done iteratively to recover the original image. It is a block based image compression, detecting and coding the existing similarities between different regions in the image. Time consuming is one the main drawback of Fractal image compression. In Fractal image compression encoding is take long time compared to decoding. Decoding is fast than encoding. In encoding the process is searching the appropriate domain for each range. In nowadays, Fractal image compression is very attractive [3]. Fractal image compression is developed by Hutchison and Barnsly. It is based on theory of IFS (Iterated Function System).In Image compression Jacquin and barnsly can first use of IFS.In ifs coding scheme the image will be partitioned into non overlapped range blocks. For each block, a similar domain block is found using IFS mapping. In IFS mapping coefficient will represent a data of block of the compressed image. Decoding proceed as follows. The transformation function used in fractal image compression is chosen in such a way that its unique fix point is a close approximation of the input image. Compression occurs because storing the details of the image transform (also known as encoding) takes up much less space than the original image. Decompression (or decoding) involves applying the transform repeatedly to an arbitrary starting image, to arrive at a (fractal) image that is either the original, or one very similar to it. To illustrate the concept of representing an image as a transform, consider one of the simplest fractals: the Sierpinski triangle (see Fig 2). This image is self-similar, in the sense that it consists of three smaller copies of itself, positioned at lower left, lower right and top middle. If ω1 is the transformation that maps the whole image onto the shrunken copy of itself in the top middle portion of the image, and ω2 and ω3 respectively map the whole image onto the copies at lower left and lower right, then the Sierpinski triangle is a fixed point of the transformation [4]

$$\omega = \omega 1 \cup \omega 2 \cup \omega 3 \quad (1)$$

because the effect of applying ω to the Sierpinski triangle is to leave it unchanged.

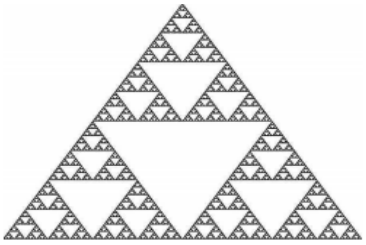

Fig. 2: The Sierpinski triangle [1]

According to the Contractive Mapping Fixed-Point Theorem, applied to a suitable space of images, the Sierpinksi triangle is the unique fix point of ω: the transformation ω is contractive because all of its transformations ωi are contractive, which means that when applied to any two points it brings them closer together. The theorem states that if ω is contractive there is a unique fixpoint (attractor) limit of the sequence

$$x, \omega x, \omega^2 x, \ldots \quad (2)$$

where x is any starting image, and ω^i denotes the composition of ω with itself i times. So the Sierpinski triangle can be generated to any required degree of accuracy by repeatedly applying ω to any starting image, as illustrated in Fig 1.4, and in this sense, the transformation ω represents the Sierpinski triangle. Hudak also illustrates this in [Hu00] by using a Haskell implementation of ω to draw the Sierpinski triangle

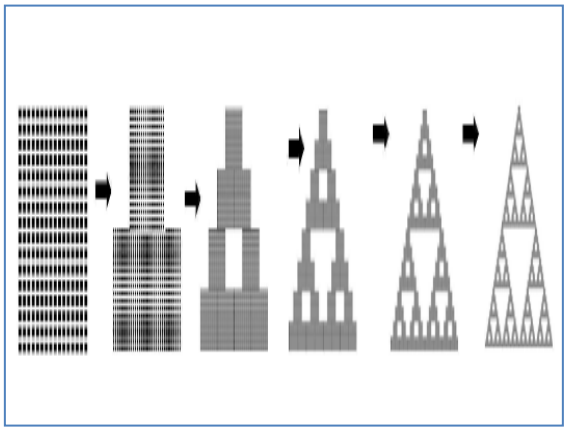

Fig. 3: Iterations of the Transformation [1]

The representation of an image by a union of contractive transforms, like (1), is known as an Iterated Function System (IFS). Although such systems are useful for compressing fractal images like the Sierpinski triangle, most images do not have such obvious self-similarity properties. However many images (particularly photographs) do have areas that are almost self-similar, as illustrated in Fig 1.5. The similarity of parts of an image provides the inspiration for the concept of a Partitioned Iterated Function System (PIFS), which is similar to an IFS, except that the contractive transforms have restricted domains. In practice, this means that an image to be compressed is partitioned into small regions (ranges), each of which is matched as closely as possible to one of several larger regions (domains) of the image. The best match for each range yields a contractive transformation that maps part of the original image onto that range block. As with an IFS, the union of the contractive transforms encodes the original image [5]. To illustrate, an easy way to partition an image for a PIFS is to divide the image into a grid of small squares for the range regions (blocks), and use larger square regions within the image for the domain blocks. Each range block is then compared with all the domain blocks to find the one that matches it most closely. As well as being reduced in size, each domain block can be rotated or reflected in order to best match the range block. It can also have its contrast and brightness adjusted. For example, Figure 3 illustrates a small square range block, 8-by-8 pixels, along with a larger 16-by-16 pixels domain block. This domain block, in the unrotated and unreflected orientation, is a good match for the range block, needing only a very small contrast and brightness adjustment, as the colors already match closely.The reproduction of the original image from the PIFS transform is illustrated in Fig 4. The decoding process involves repeatedly applying the transform until it converges to an image, which closely approximates the original:

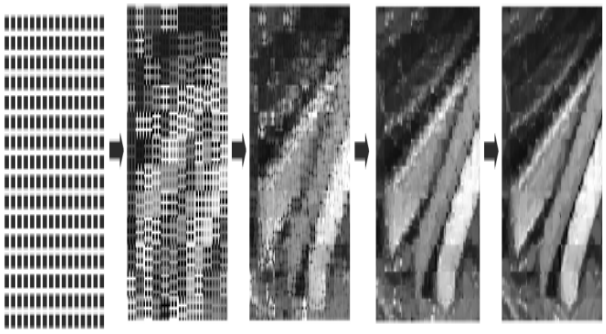

Fig. 4: Decoding of an Image [1]

3 General Procedure for Fractal Image Compression

Fractal encoding relies on the fact that all natural, and most artificial, objects contain redundant information in the form of similar, repeating patterns called fractals. Following are general procedure of fractal image compression [5].

- **Encoding:**
  Step 1: Initially we get a image and divide it into small and non-overlapping, square blocks. This is called as a “parent blocks”.
  Step 2: Divide each parent block into 4 each blocks, or “child blocks.”
  Step 3: Compare each child block against a subset of all possible overlapping blocks of parent block size.
  Step 4: Need to reduce the size of the parent to allow the comparison to work.
  Step 5: Determine which larger block has the lowest difference, according to some measure, between it and the child block.
  Step 6: Calculate a grayscale transform to match intensity levels between large block and child block precisely.
  Typically an affine transform is used (w*x = a*x + b) to match grayscale levels.
- **Decoding**
  Step 1: Read in child block and transform block position, transform, and size information.
  Step 2: Use any blank starting image of same size as original image
  Step 3: For each child block apply stored transforms against specified transform block

Step 4: Overwrite child block pixel values with transform block pixel values
Step 5: Repeat until acceptable image quality is reached.

4 Cryptography

Cryptography is a method of storing and transmitting data in particular form so that only those for whom it is intended can read and process it.The term is most often associated with scrambling plaintext (ordinary text, sometimes referred to as clear text) into cipher text (encryption), then back again (decryption).

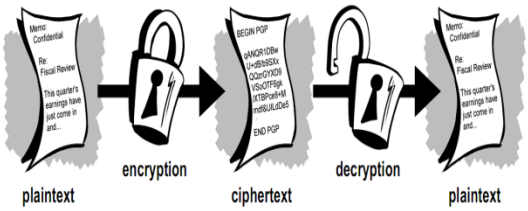

Fig 5: Cryptography process

Data that can be read and understood without any special measures is call plaintext or clear text. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable gibberish called cipher text. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption[29].The algorithm used for cryptography are as follow:

Encryption Algorithm:
Step 1: Convert the Data to ASCII by
datat=int(data)
Step 2: Input the key say K
Step 3: Convert Key to ASCII by
AK=int(K)
Step 4: For i=1:length(datat)
Step 5: ECtext(i)=datat(i)+AK(i)
Step 6: End
Step 7: Convert ECtext into char to analyze the output
ECtext=char(Ectext)
Decryption Algorithm:
Step 1: Convert the Cdata to ASCII by
datat=int(Cdata)
Step 2: Input the key say K
Step 3: Convert Key to ASCII by
AK=int(K)
Step 4: For i=1:length(datat)
Step 5: DCtext(i)=datat(i)-AK(i)
Step 6: End
Step 7: Convert DCtext into char to analyze the output
DCtext=char(DCtext)

5 Proposed System

APCC based FIC (Fractal Image Compression) technique which is based on the local similarity of image structure. It is widely used in many fields such as image retrieval, image denoising, image authentication, and encryption. FIC, however, suffers from the high computational complexity in encoding. First, all blocks in the range and domain pools are chosen and classified using an APCC-based block classification method to increase the matching probability. Second, by sorting the domain blocks with respect to APCCs between these domain blocks and a preset block in each class, the matching domain block for a range block can be searched in the selected domain set in which these APCCs are closer to APCC between the range block and the preset block. So for security purpose the APCC based FIC technique is not efficient. For the efficient working, the cryptography and Steganography techniques have been implemented on FIC scheme. Cryptography is a method of storing and transmitting data in particular form so that only those whom for it intended can read and process it. Steganography is a technique of concealing file, message, image or video into another file, message, image or video. This will results in efficient performance of the existing APCC based FIC scheme .So that more data can be transferred from one place to another without in the compressed form using APCC based classification. The graphical comparison confirms the better performance of the proposed protocol is better than the existing technique.This technique will also show that with more data to hide PSNR improves. Hence with the proposed technique PSNR as well as MSE improves.

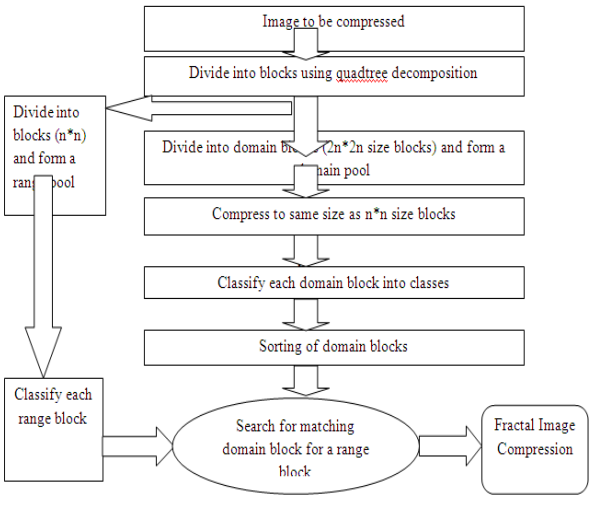

Fig. 6: Flow chart of Fractal Image Compression

Basic Steps of flow chart:
Step1: Take an image to be compressed.

Step2: Divide the image into blocks using quadtree decomposition.
Step3: Divide the blocks into Domain blocks (2n*2n) and pool obtained is known as domain pool and range blocks (n*n) and pool obtained is known as range pool.
Step 4: Compress each domain block into domain pool in n*n size (same as range block).
Step 5: Classify each domain block and range block into different classes.
Step 6: Searching of matching domain block for a range block and finally Fractal image compression has been achieved.

Flow Chart of Proposed Work

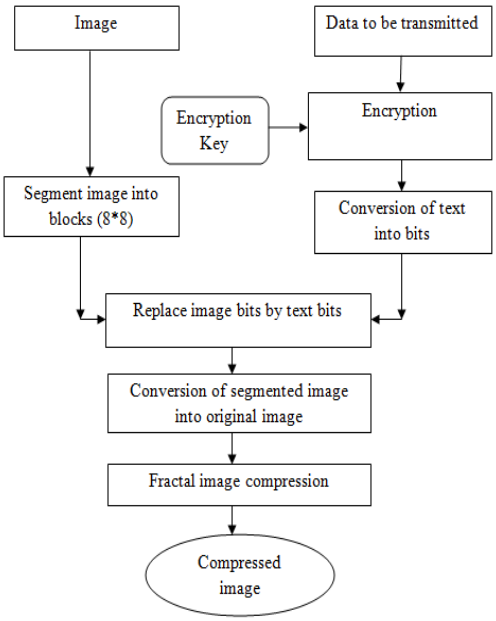

Fig. 7: Flow chart of Cryptography and Steganography implementation on Image compression scheme

Basic Steps of flow chart:

Step1: Take an image to be compressed and text to be hide and perform the encryption.
Step2: Segment the image into 8*8 size blocks and convert the text into bits.
Step3: Replace the image bits by text bits.
Step 4: Now convert the segmented image obtained in the above steps into the original image in which text is hidden.
Step 5: Perform the fractal image compression on the above process so we get the compressed image.
Step 6: Hence we get the compressed image with text stored in it via Steganography technique and security provided by encryption process.

6 RESULTS AND DISCUSSION

The comparison of image compression on various images is done by using the various parameters i.e. compression score and the PSNR and MSE. The compression score describe that how much the image is compressed. The target is to get the greater compression with good image quality. The MSE and PSNR are described below:

- **Mean Square Error (MSE)**
  Mean square error is a criterion for an estimator: the choice is the one that minimizes the sum of squared errors due to bias and due to variance. The average of the square of the difference between the desired response and the actual system output as a loss function, MSE is called squared error loss.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|I(i,j) - K(i,j)\|^2 \quad (3)$$

  Where m x n is the image size and I(i,j) is the input image and K(i,j) is the retrieved image.

- **Peak Signal to Noise Ratio (PSNR)**
  It is the the ratio between the maximum possible power of a signal and the power of corrupting noise .
  The PSNR is defined as :

$$PSNR = 10.\log_{10}\left(\frac{Max_I^2}{MSE}\right) = 20.\log_{10}\left(\frac{Max_I}{\sqrt{MSE}}\right) \quad (4)$$

  Here, MAXi is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

Fig. 8 represents sample images that have been used in the experiments. They are fiver images Lena, Car, Cameraman, Heart Khushboo; Fig. 9 shows the compression time, Fig .10 shows compression ratio, Fig. 11 shows the MSE, Fig. 12 shows PSNR according to the above given table.

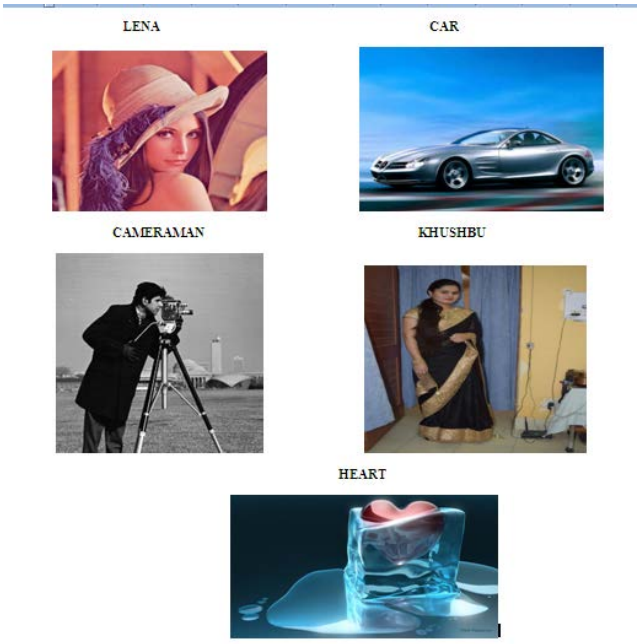

Fig. 8: Reference Images

Fig. 9 shows the comparison of compression or encoding time in seconds for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) in between existing APCC technique, previous encoding technique and proposed technique.

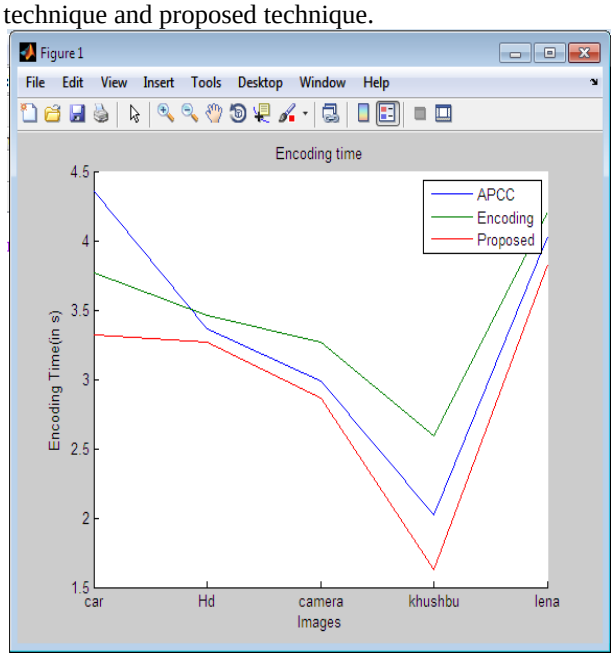

Fig. 9: Compression Time for different images

Fig. 10 shows the comparison of compression ratio for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) in between existing APCC technique, previous encoding technique and proposed technique.

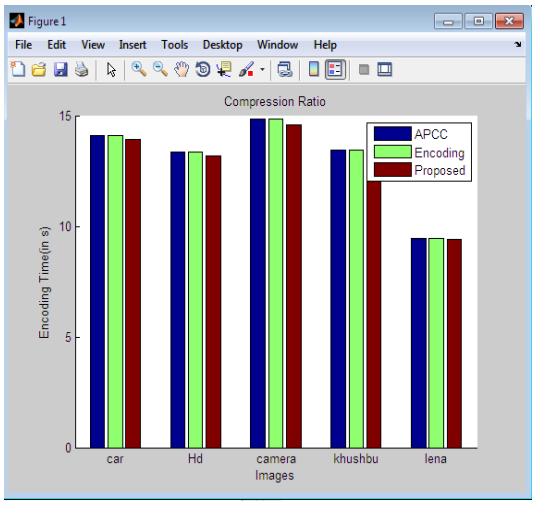

Fig. 10: Compression Ratio for different images

Fig. 11 shows the comparison of MSE for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) in between existing APCC technique, previous encoding technique and proposed technique.

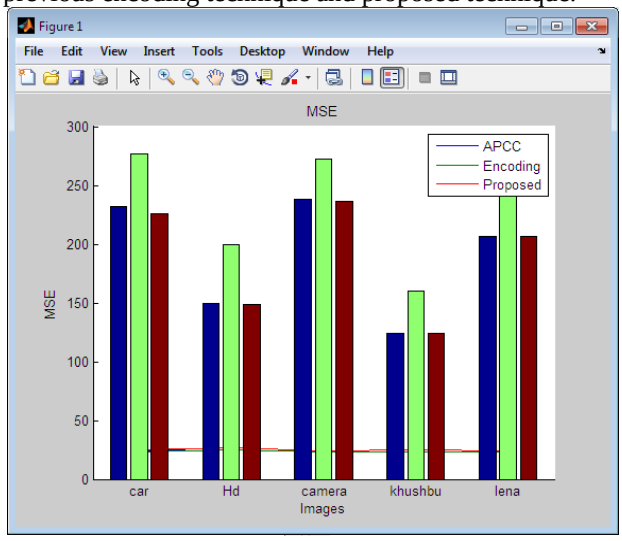

Fig 11: MSE for different images

Fig. 12 shows the comparison of PSNR for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) in between existing APCC technique, previous encoding technique and proposed technique.

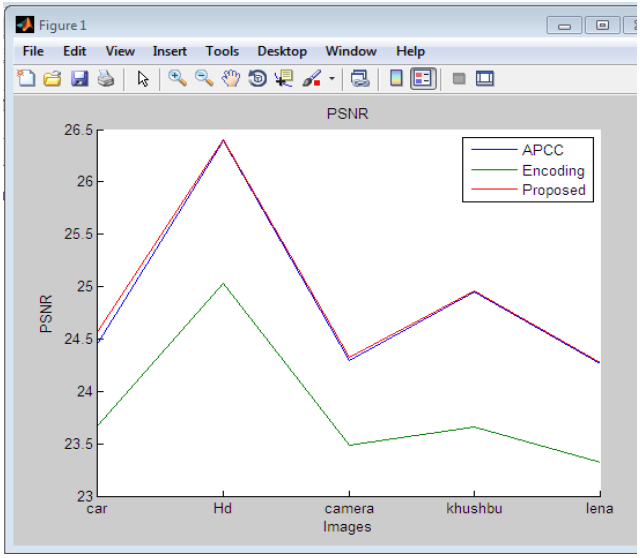

Fig. 12: PSNR for different images

Fig. 13 shows the MSE for different images (Lena, Cameraman, Heart (Hd), Car and Khushboo) at varying character length in proposed technique.

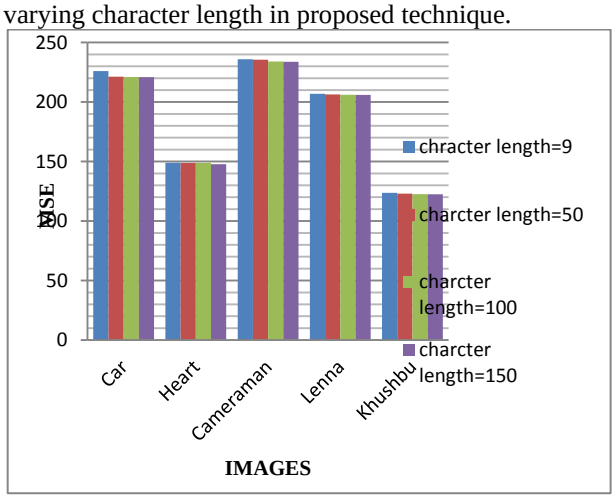

Fig. 13: MSE of different images at varying character length

Fig. 14 shows the PSNR for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) at varying character length in proposed technique.

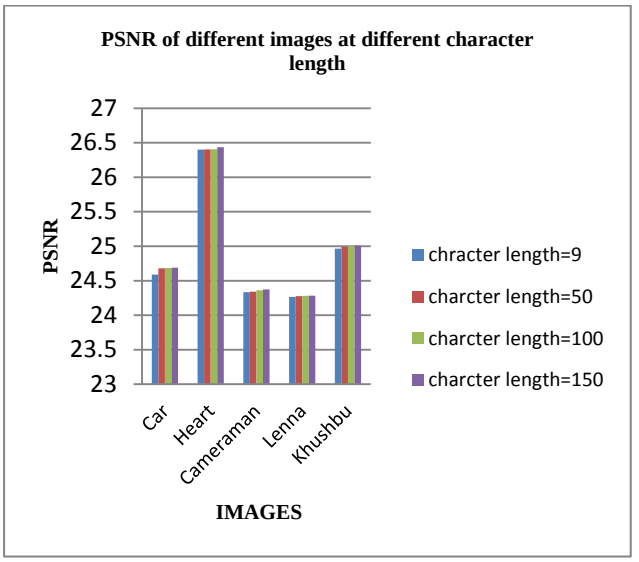

Fig. 14: PSNR of different images at varying character length

Fig. 15 shows the MSE for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) at varying domain step length with character length=150 in proposed technique.

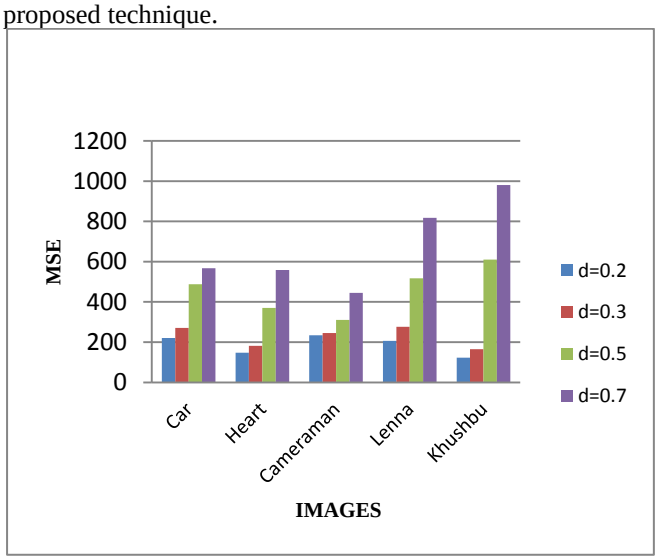

Fig. 15: MSE of different images at varying Domain step length

Fig. 16 shows the PSNR for different images (Lena, Cameraman, Heart (Hd), Car and Khushbu) at varying domain step length with character length=150 in proposed technique.

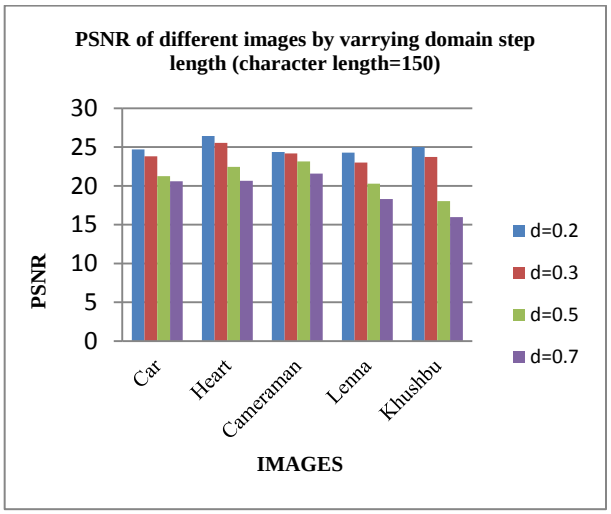

Fig. 16: PSNR of different images at varying Domain step length

CONCLUSION AND FUTURE SCOPE

Cascading the cryptography and steganography techniques are used so that more data can be transferred from one side to another in the compressed form using APCC based block classification scheme in FIC. This technique also shows that with more data to hide the PSNR improves. Hence in the given data the MSE (Mean square error) has been reduced. When there is more data to hide there is improvement in PSNR. Encoding time has also been reduced in the given technique. The given technique is highly secured. Application of Fractal image compression is also extended to the field of mobile communications. The simplicity and regularity of the method makes it suitable to be implemented on programmable logic devices, such as FPGA's. This type of compression can be applied in Medical Imaging, where doctors need to focus on image details, and in Surveillance Systems, when trying to get a clear picture of the intruder. Fractal image compression could be used to compress topographic images and it may be useful in PACS and telemedicine.

REFERENCES

[1] Curtis and Martin, "Functional fractal image compression" *Department of Computing, Oxford Brookes University, UK*, pp 383-398.

[2] M. B. Bhammar and K. A. Mehta.," A Survey of various image compression techniques" *International Journal of Darshan Institute on Engineering Research and Emerging Technology* ,pp 85-90, September 2012.

[3] Dr. K. Kuppusamy, R.Ilackiya, "Fractal Image Compression & Algorithmic Techniques" *International Journal of Computer & Organization Trends* –Volume3 Issue4, pp-141-145, May 2013.

[4] J. Wang, & N. Zheng," A Novel Fractal Image Compression Scheme with Block Classification and Sorting Based on Pearson's Correlation Coefficient" *IEEE Transactions on Image Processing*, pp- 3690-3702, September 2013.

[5] S.Michael Vanith & K...Kuppusamy,"Survey on Fractal Image Compression" *International Journal of Computer Trends and Technology (IJCTT)*, volume4 Issue 5, pp-1462-1464,May 2013.

[6] H. Hartenstein, & D. Saupe, "Lossless acceleration of fractal image encoding via the fast Fourier transform" *Signal Processing: Image Communication*, *16*(4), pp-383-394, July 2000.

[7] Y. G. Wu, M. Z. Huang, & Y. L. Wen, "Fractal image compression with variance and mean" *IEEE International Conference on Multimedia and Expo. (ICME)*, Vol. 1, pp-353-356, July 2003.

[8] J. Wang, & N. Zhen" A Novel Fractal Image Compression Scheme with Block Classification and Sorting Based on Pearson's Correlation Coefficient*".IEEE transactions on image processing*, VOL. 22, 2013. Pp 3690-3702, September 2013.