SCIRES it
SCIentific RESearch and Information Technology
*Ricerca Scientifica e Tecnologie dell'Informazione*
Vol 13, Issue 1 (2023), 167-180
e-ISSN 2239-4303, DOI 10.2423/i22394303v13n1p167
Open access article licensed under CC-BY-NC-ND
CASPUR-CIBER Publishing, http://www.sciresit.it

# SIMILARITY BASED OPTIMIZATION TO FRACTAL IMAGE ENCODING BASED ON MULTITHREADING PARALLELIZATION

*Ranjita Asati*, Mukesh Raghuwanshi**, Kavita Singh**

*Department of Computer Technology, Yeshwantrao Chavan College of Engineering - Nagpur, India.
**Department of Computer Science and Engineering , Symbiosis Institute of Technology - Nagpur, India

**Abstract**

E Higher compression time is a major issue in adoption of fractal image coding even though it offers various advantages in terms of higher compression ratio, higher resolution, and lower decompression time. Many optimizations have been proposed earlier to reduce the computation time in terms of parallelism and encoding space reduction. This work proposes an integrated approach combining both multithreaded parallelism and similarity based encoding space reduction to diminish the time of compression in Fractal image coding. The compression time of the proposed integrated method is tested for images of different resolution and the proposed solution is able to reduce the compression time by almost 4.4 times compared to existing fractal image compression techniques.

**Keywords**

Fractal image compression, Fractal image coding, Data parallelism, Multithreading, SSIM, Entropy

## 1. *Introduction*

Compressing the image is important for efficient storage and transmission. Exponentially growing data volumes need effective data storage management also the rise of computer networks needs the effective bandwidth of networks by reducing the number of bits before transmission. The goal of image compression is to reduce irrelevance and redundancy of the image data to store or transmit data in an efficient form (Drakopoulos et al 2013). Image compression techniques can be briefly categorized to two types of: lossy and lossless schemes. Image quality is retained without information loss in case of lossless schemes. Lossy schemes exhibit a tolerable information loss. In proportion to loss, the compression ratio also increases. Fractal image coding (FIC) is one such lossy compression scheme (Jacquin et al 1989). It is built exploiting the redundancy in structure of the images. Fractal image compression achieves a higher compression ratio for images of self-similar characteristics.. A notable feature of FIC is that it is resolution independent. Due to this, there is not much difference in encoding for images and its scale up or scaled down versions. FIC encodes the image in terms of set of contractive affine transformations and parameters of these transformations. These transformations and parameters are saved to a binary file (Chen et al 2013). The effectiveness of FIC in achieving higher compression ratio is demonstrated in many works. FIC was able to give superior quality compressed image event at compression ratio as that of DCT based algorithms (Wohlberg et al 1999). For satellite images, FIC achieved a compression ratio of 170:1 (Woon et al 2000). For videos, a compression ratio of 244:1 was realized using FIC (Fisher et al 1995). Though these works achieved higher compression ratio, the compression time was very high. Many attempts have been made by various works to reduce the compression time in FIC. The existing strategies for reducing the compression time are based on two important optimizations: parallelism and reducing the encoding space. Search of matching blocks and finding the effective transformation for it is the time-consuming work in FIC (Fisher et al 1995). Parallelism speeds up this task by starting multiple instances of this task and maximizing the CPU utilization. Encoding space reduction algorithms reduce the number of instances of searching of matching blocks by exploiting the self-similarity in the images. In this work, an integrated solution is proposed combining parallelism and encoding space reduction. Encoding space reduction is done by splitting the images to grid and grouping similar grids based on two measures of entropy and structural similarity index metric (SSIM).

A representative grid is selected in the group of similar grids. Multi thread-based parallelism-based FIC is done on the representative grid. By this way the encryption time is reduced. The compression time can be reduced in inverse proportion to grid similarity thresholds used in entropy and SSIM. Remainder of this paper is structured as follows: Section 2 put forward related work on the image similarity metrics and measures. Section 3 briefly describes the proposed solution. Section 4 discourses about the result part. In the end, some conclusions are presented in Section 5.

## 2. *Related Work*

The core of the proposed solution lies in two concepts of encoding space reduction and parallelism. Our earlier work, Ranjita et al (2021) provided a detailed review of existing works on parallelism. In this work review was done on reducing encoding space. Encoding space reduction involves grouping similar image blocks and doing FIC only on the most representative block in the similar blocks. This necessitates computing the similarity between the blocks. The existing techniques to compute the similarity between image blocks are detailed in this section. The similarity metrics most used in existing works are: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Universal Quality of Image Index (UQI), Multi scale structural similarity index (MS-SIM), Error Relative Globale Adimensionnelle de Synthèse (ERGAS), Spatial Correlation coefficient (SCC), Spectral Angle Mapper (SAM), Visual information fidelity (VIF) and Entropy.

Mean Square Error (MSE) is the average squared difference between two image patches pixel by pixel. It is calculated as

$$d = \frac{1}{N} . \sum_{i=1}^{N} (x_i - y_i)^2 \qquad (1)$$

Where N is the number of pixels in image patch x and y. MSE just provides the average of distance between pixels. PSNR and RMSE are a variation of MSE. MSE, RMSE and PSNR does not provide any information about structure. A study by Hore et al (2010) and Lu et al (2019) have found that these metrics could not discriminate structural contents in the image and various types of degradations applied to the same image can yield same values for MSE and PSNR. Thus RMSE, MSE and PNSR are used only for measuring the effectiveness of image de-noising algorithms and not in image similarity applications. SSIM proposed in Wang and Bovik (2002) and Wang et al. (2004) is a composite measure involving three factors: mean ($\mu$) of luminance, standard deviation ($\sigma$) of contrast and correlation coefficient ($\rho$). It is calculated as

$$SSIM = \frac{2.\mu_x \mu_y}{{\mu_x}^2 + {\mu_y}^2} . \frac{2.\sigma_x \sigma_y}{{\sigma_x}^2 + {\sigma_y}^2} . \rho \qquad (2)$$

Palubinskas et al (2019) conducted various experiments with various types of distortions and found SSIM was able to discriminate structure contents and very sensitive to degradations compared to MSE. Wang et al (2002) proposed a metric called universal objective image quality index. It is based on three factors of: correlation loss, luminance distortion and contrast distortion. But the mathematical formulation used in this index does not adequately capture the estimate of correlation between images. Due to this, its sensitivity to distortion is more than MSE but falls short of SSIM. Wang et al (2003) proposed a multi scale structural similarity index (MS-SIM). Though it is more flexible than single scale methods in incorporating variations of viewing conditions, the computation complexity is high, and the scale selection is dependent on images. Wald et al (2000) proposed a metric called ERGAS specific for comparing similarity between synthetic images. The proposed metric is based on spectral correlation analysis and the computational complexity is higher in this method. Similar its ERGAS, an image similarity metric in spatial domain called Spatial Correlation Coefficient (SCC) was proposed by Vallejos et al (2016). The metric is based on analysis of hidden spatial association between two images. It does this by adding a co-dispersion coefficient to SSIM. Addition of co-dispersion coefficient to SSIM increases the computation complexity. Girouard et al (2004) proposed a metric called spectral angle mapper. This metric extracts spectra from individual or groups of pixels and compute statistics for regions of similar composition. This metric can also say if there are any similar regions in two images. Sheikh et al (2006) proposed visual information fidelity metric based on image information and visual quality. The image information quantifies the information present in image and how much of this can be extracted from another image. The metric involves too many matrix multiplications, and it is computationally intensive. Aljanabi et al (2018) proposed an entropy measure for measuring image similarity, combining information theory and joint histogram. This measure has higher performance and accuracy compared to structural similarity index measure (SSIM), feature similarity index measure (FSIM) and feature-based structural measure (FSM). From the survey, Entropy measure proposed by Aljanabi et al (2018) and SSIM metric are found to more suitable for FIC as both have higher performance in computation and accuracy. Both metrics consider structural differences between the images.

## *3. Proposed Solution*

In the literature, many solutions involving parallelization or encoding space reduction were discussed. Most hardware-based parallelization has higher cost per compression time gain. Most encoding space reduction algorithms have quality distortion. Thus, there is need for solution with less comparison time without much quality distortion and reduction in compression time. This work suggests a hybrid solution integrating both parallelization and encoding space reduction. The overall encoding process is given in Figure 1. The overall decoding process of the proposed solution is given in Figure 2. The proposed solution has two parts of encoding space reduction technique followed by parallelization. Each of it is detailed below.

### *3.1 Encoding space reduction*

Two different strategies of entropy and structural similarity metric (SSIM) is used in this work for encoding space reduction. Both Entropy and SSIM are effective metrics for measuring similarity between two images and they are independent of image nature. They can work for any kind of images like medical, natural scenery, human portrait etc. These two metrics are selected for these properties in this work.

The image is split to equal sized *N* blocks. The blocks are grouped to cluster based on the entropy or SSIM similarity metric. From each cluster, a representative block is selected. FIC using multi thread parallelism (discussed in next section) is done only for the representative block, skipping the other blocks. In proportion to the number of blocks skipped, the compression time drops.

The pseudo code for the proposed entropy/SSIM encoding algorithm is given below

***Algorithm: Entropy/SSIM encoding***

***Step 1****. Divide the image into blocks, say N*

***Step 2****: Take each block and calculate the SSIM (or entropy) to remaining blocks. If the SSIM (or entropy) > threshold [it means blocks are similar], then add the blocks to same cluster.*

***Step 3****: At end of step 2, a cluster is created in such a way that blocks within the cluster are similar.*

***Step 4****: For each cluster, take one representative block. (One block selected from each cluster).*

***Step 5:***
*encoded_rep_block ←Do_Parallel_FIC_encode (representative blocks) and compress using Fractal compression. For the rest of blocks in the cluster skip the Fractal compression and copy the result of representative block*

***Step 6****: write the result to a binary file.*

The decoding process is the reverse of the encoding process. The representative blocks are first generated for each cluster using FIC decoding process. The rest of cluster blocks are generated from their corresponding representative block. They are fitted into their position to get the original image block. The pseudocode for the proposed entropy/SSIM decoding algorithm is given below.

***Algorithm: Entropy/SSIM decoding***

***Step 1****. Extract the encoded representative block from compressed file (Repblock generated in encoding)*

***Step 2****: Repblock ←Parallel_FIC_decode (encoded representative blocks).*

***Step 3****: Fit Repblock and generate image*

***Step 4****: return image.*

### *3.2 Parallelization*

The parallel FIC encoding/decoding algorithm used by the Entropy/SSIM encoding and Entropy/SSIM decoding is implemented by adding multithreading level parallelism to serial version of FIC algorithm (Abdul-Malik et al 2018). The multithread level parallelism is added in such a way that it is a data level parallelism without much inter thread communication. In the serial FIC, an initial image of size M*M is split to m non-overlapping range regions of size r*r where $m = M/r^2$ and n overlapping domain blocks of size 2r*2r where $n = (M - 2r + 1)^2$.

For each range block an approximate domain block and a relevant contractive affine transformation is selected such that $d(R_i, w_{ik}(D_k)) = \min d(R_i, w_{ij}(D_j))$

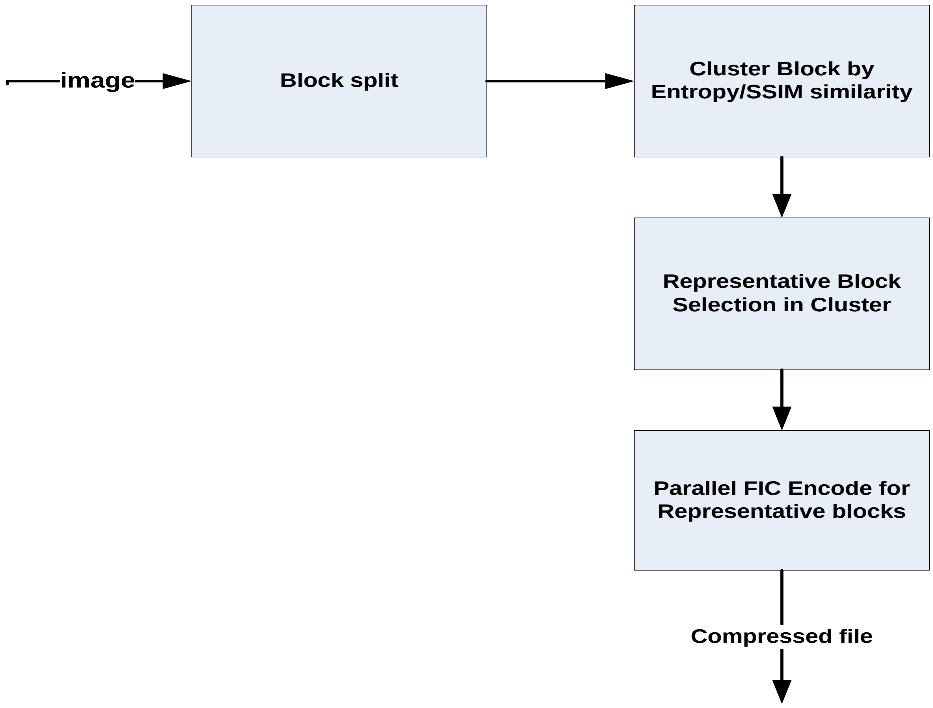



**Fig. 1:** Overall encoding process

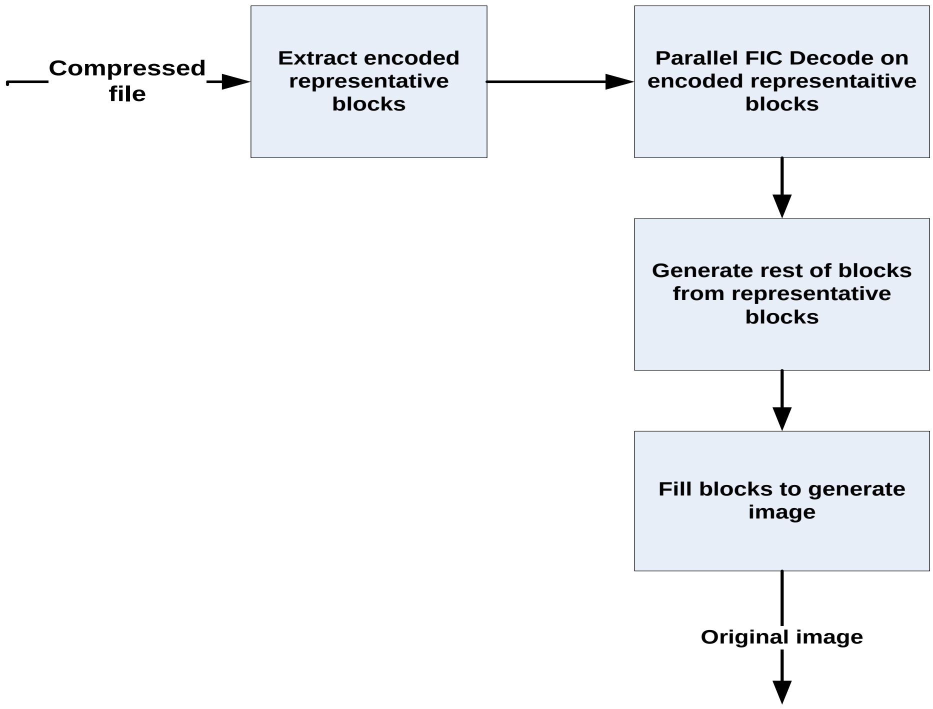



**Fig. 2:** Overall decoding process

Where $w_{ik}$ is the contractive affine transformation from k-th domain block and the i-th range block i.e., $D_k$ to $R_i$. This is done in a way such that mean square error distance represented as $d(R_i, w_{ij}(D_j)$ from range block $R_i$ and the transformed domain block $w_{ij}(D_j))$ is minimized.

The biggest contribution to encoding time is time spent in searching for similarity between range block and domain block (Plovera et al 2000). This time is reduced by adding parallelism to this search process. This search process parallelism has additional advantage of no associated issues of inter inter-thread communication, resource contention, racing, deadlocks etc.

Each thread processes a range block in the range block pool. The thread does transformation of domain blocks, finds the matching transformed blocks based on minimization of mean square error between the range block and the transformed domain block. Each thread generates the transformed parameters for its range block which is returned as a parameter to the Entropy/SSIM encoding algorithm.

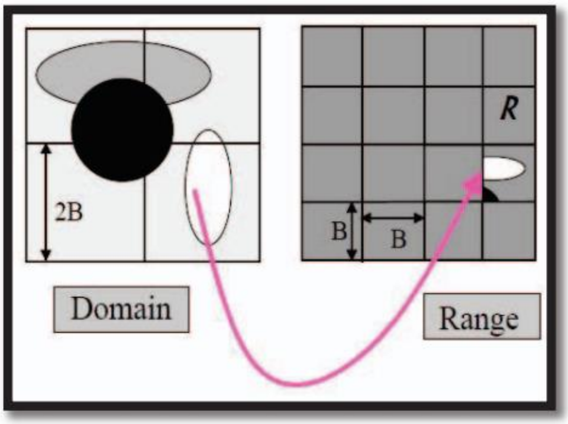



**Fig. 3:** Domain block to range block mapping

The steps 5 to 10 in the serial FIC are selected for multithread parallelism that constitute one thread. Once all the threads complete, the results are collected and returned.

The pseudo code of the parallel FIC encoding algorithm is given below.

**Algorithm: Parallel_FIC_Encode**

***Step 1****: A given image m is split into non overlapping range regions.*
***Step 2****: A given image m is split into overlapping domain regions.*
***Step 3:*** *For each range block start a thread*
*In each thread for each domain block the approximate range block by selecting the range block with lowest MSE distance between the range block and transformed domain block.*
***Step 4****: Wait for all threads to complete*
***Step 5****: Return all the transformations.*

The decoding process is also parallelized by executing on each transformation in parallel. The pseudo code of the parallel FIC decode algorithm is given below.

**Algorithm: Parallel_FIC_Decode**

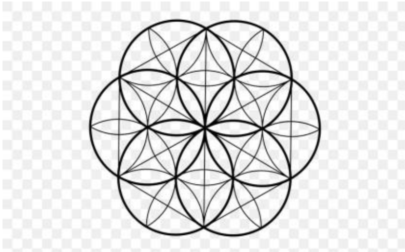
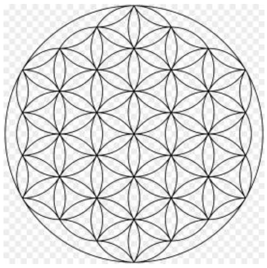
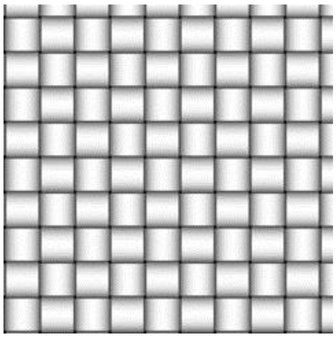
***Step 1****. Generate range blocks from the encoding result file*
***Step 2****: Select an initial image with same size as that of the original image.*
*Step 3: Split range block matrix to K partition and start K threads*
***Step 3****: In each thread do the following*
*Apply the stored transformations that resulted from the transformed block for each range block.*
*Replace the pixels of the range block with the pixels obtained from the transform block.*
*The transformations and mappings are applied on the initial image iteratively until the image is restored.*
***Step 4****: Wait for all threads to complete*
***Step 5****: Return the reconstructed image.*

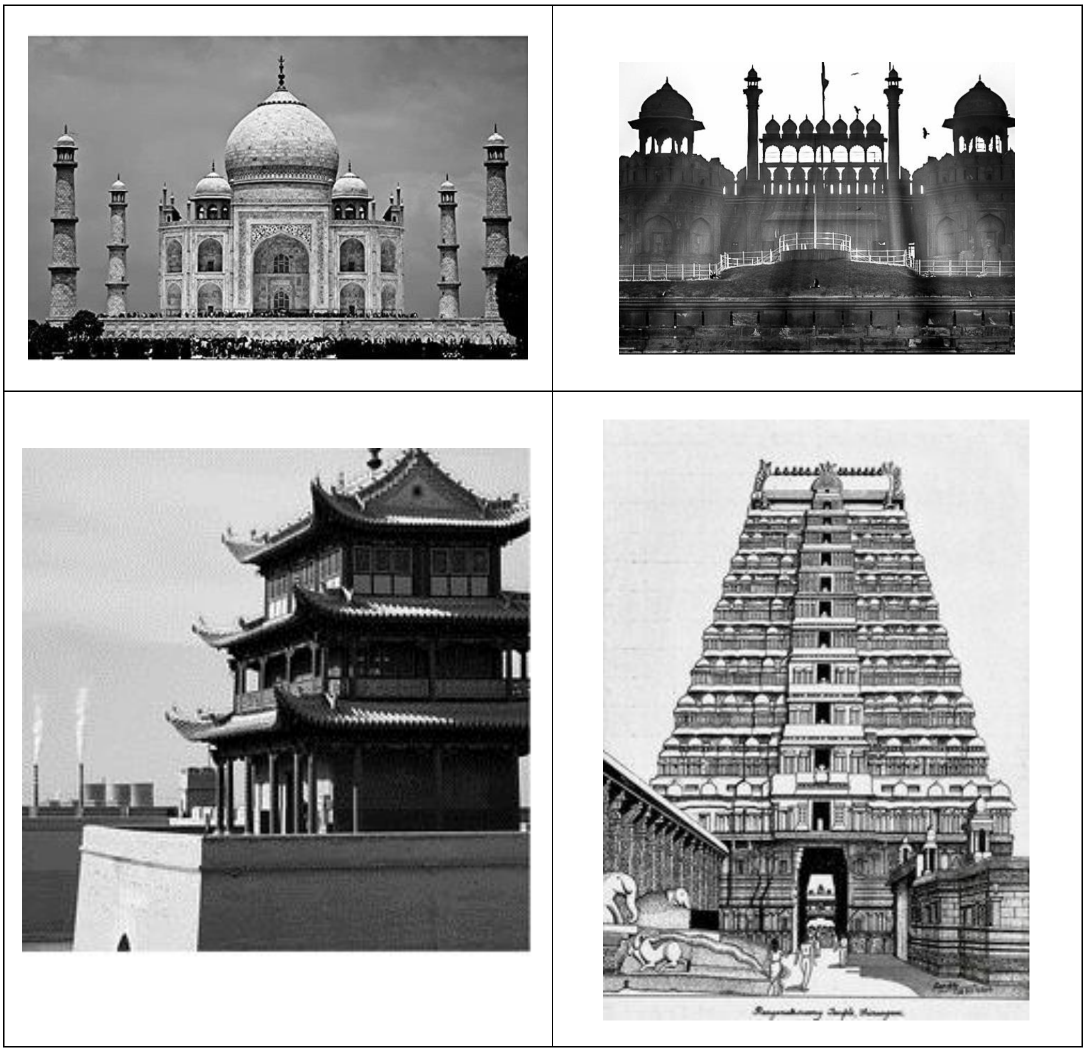
## *4. Results*

The performance assessment of the proposed solution was conducted in 10th generation Intel core i5 10300H, 2.5 GHZ, 4 cores, 8 threads, 8GB RAM. The dataset image for performance testing was obtained by taking 14 different images in resolution of 512*512 and rescaling them to dimensions of 64*64, 200*200, 225*225. The proposed solution was compared with serial and parallel version of FIC implementation. The encoding and decoding times are measured for different images (Table 1)

The results for comparison of compression time across the solutions are given in Table 2. The plot of it is given in Figure 4. From the results, it can be seen that Entropy based FIC methods were able to provide lower compression time followed by SSIM based methods. Grouping of blocks based on entropy and SSIM reduced the effective number of blocks to be considered for FIC encoding. This has reduced the compression time in entropy based and SSIM based solutions.

**Tab. 1**: Images used for experimentation

| | |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |



Compared to SSIM, entropy was more effective in reducing the number of blocks to be considered for FIC. This is because entropy better measures the joint probability of co-occurrence than means and variance used in SSIM.

Entropy parallel FIC achieved the lowest compression time, and it is on average 12.64 times lower than serial FIC, 4.4 times lower than parallel FIC and 1.92 times lower than SSIM parallel FIC.

The results of comparison of decoding time across the solutions are given in Table 3. The plot of results is given in Figure 5. The decoding time is lower in Entropy parallel FIC compared to all other methods. But the difference is very less compared to compression time. The decoding time in Entropy parallel FIC is on average 4.14% lower compared to serial FIC, 2.5 times lower compared to parallel FIC and 1.53 times lower compared to SSIM parallel FIC.

The decoding time is reduced in Entropy parallel FIC due to reduction in number of blocks to be decoded. Since Entropy schemes has reduced blocks compared to Serial scheme, the decoding time is comparatively lower in Entropy parallel FIC. The decoding time is reduced in Entropy parallel FIC due to reduction in number of blocks to be decoded. Since Entropy schemes has reduced blocks compared to Serial scheme, the decoding time is comparatively lower in Entropy parallel FIC.

The results of comparison of PSNR across the solutions for different image resolutions are given in Table 4. The box and whisker plot of it is presented in Figure 6. In case of 64*64 resolution images, the difference in PSNR between solutions is less than 2 dB, in-case of 200*200 resolution images, the difference in PSNR between solutions is less than 5dB, in case of 225*225 resolution images, the difference in PSNR between solution is less than 3dB and incase of 512*512 resolution images, the difference in PSNR is less than 7 dB. There is not much difference in PSNR across the solutions for different image resolutions.

The results of comparison of MSE across the solutions for different image resolutions are given in Table 5. The box and whisker plot of it is presented in Figure 7. In case of 64*64 resolution images, the difference in MSE between solutions is less than 0.01, in-case of 200*200 resolution images, the difference in MSE between solutions is less than 0.01, in case of 225*225 resolution images, the difference in MSE between solution is less than 0.05 and incase of 512*512 resolution images, the difference in MSE is less than 0.01.

There is not much difference in MSE across the solutions for different image resolutions. The proposed hybrid strategy of SSIM/Entropy with parallel FIC has not created a visual quality distortion as there is only a minor difference in PSNR and MSE.

The results of comparison of Compression ratio (CR) across the solutions are given in Table 6. The box and whisker plot of it is presented in Figure 8. In case of 64*64 resolution images, the difference in CR between solutions is greater than 1, in-case of 200*200 resolution images, the difference in CR between solutions is greater than 3, in case of 225*225 resolution images, the difference in CR between solutions is greater than 3 and incase of 512*512 resolution images, the difference in CR is greater than 7. Maximum values of compression ratio are in Entropy and SSIM parallel FIC, as the transformation of only the representative blocks are stored in the compression file compared to all block's transformations in other FIC solution.

**Tab 2:** Compression time results

| | Compression time (seconds) | | | | | |
|---|---|---|---|---|---|---|
| **Size** | **Serial FIC** | **Parallel FIC** | **SSIM Serial FIC** | **SSIM Parallel FIC** | **Entropy Serial FIC** | **Entropy Parallel FIC** |
| 64*64 | 36.46 | 7.56 | 0.32 | 0.27 | 0.077 | 0.058 |
| 200*200 | 155.31 | 29.61 | 26.88 | 21.76 | 17.34 | 11.98 |
| 225*225 | 228.33 | 93.36 | 44.45 | 41.76 | 23.76 | 22.24 |
| 512*512 | 4058.1 | 1824.3 | 1415.6 | 576.7 | 313.62 | 274.4 |

**Tab 3:** Decompression time results

| | Decompression time (seconds) | | | | | |
|---|---|---|---|---|---|---|
| **Size** | **Serial FIC** | **Parallel FIC** | **SSIM Serial FIC** | **SSIM Parallel FIC** | **Entropy Serial FIC** | **Entropy Parallel FIC** |
| 64*64 | 0.945 | 0.93 | 0.22 | 0.14 | 0.09 | 0.07 |
| 200*200 | 3.244 | 1.85 | 2.61 | 1.254 | 1.00 | 0.994 |
| 225*225 | 4.34 | 2.771 | 3.21 | 1.93 | 1.17 | 1.17 |
| 512*512 | 18.1 | 11.10 | 17.66 | 6.77 | 5.301 | 4.25 |

**Tab 4:** PSNR results

| | PSNR (dB) | | | | | |
|---|---|---|---|---|---|---|
| **Size** | **Serial FIC** | **Parallel FIC** | **SSIM Serial FIC** | **SSIM Parallel FIC** | **Entropy Serial FIC** | **Entropy Parallel FIC** |
| 64*64 | 30.858 | 30.858 | 31.496 | 31.496 | 31.80 | 31.80 |
| 200*200 | 32.33 | 32.33 | 34.35 | 34.35 | 35.23 | 35.23 |
| 225*225 | 32.65 | 32.65 | 34.91 | 34.91 | 35.59 | 35.59 |
| 512*512 | 35.76 | 35.76 | 40.86 | 40.86 | 43.45 | 43.45 |

**Tab 5:** MSE results

| | MSE | | | | | |
|---|---|---|---|---|---|---|
| **Size** | **Serial FIC** | **Parallel FIC** | **SSIM Serial FIC** | **SSIM Parallel FIC** | **Entropy Serial FIC** | **Entropy Parallel FIC** |
| 64*64 | 1.98 | 1.98 | 1.986 | 1.986 | 1.986 | 1.986 |
| 200*200 | 1.96 | 1.96 | 1.973 | 1.973 | 1.97 | 1.97 |
| 225*225 | 1.97 | 1.97 | 1.967 | 1.967 | 2.011 | 2.011 |
| 512*512 | 1.93 | 1.93 | 1.94 | 1.94 | 1.942 | 1.942 |

**Tab 6:** Compression ratio results

| | Compression ratio | | | | | |
|---|---|---|---|---|---|---|
| **Size** | **Serial FIC** | **Parallel FIC** | **SSIM Serial FIC** | **SSIM Parallel FIC** | **Entropy Serial FIC** | **Entropy Parallel FIC** |
| 64*64 | 2.62 | 2.62 | 3.26 | 3.26 | 3.6 | 3.6 |
| 200*200 | 4.0 | 4.0 | 6 | 6 | 7.0 | 7.0 |
| 225*225 | 4.24 | 4.24 | 6.5 | 6.5 | 7.6 | 7.6 |
| 512*512 | 7.12 | 7.12 | 12.24 | 12.24 | 14.8 | 14.8 |

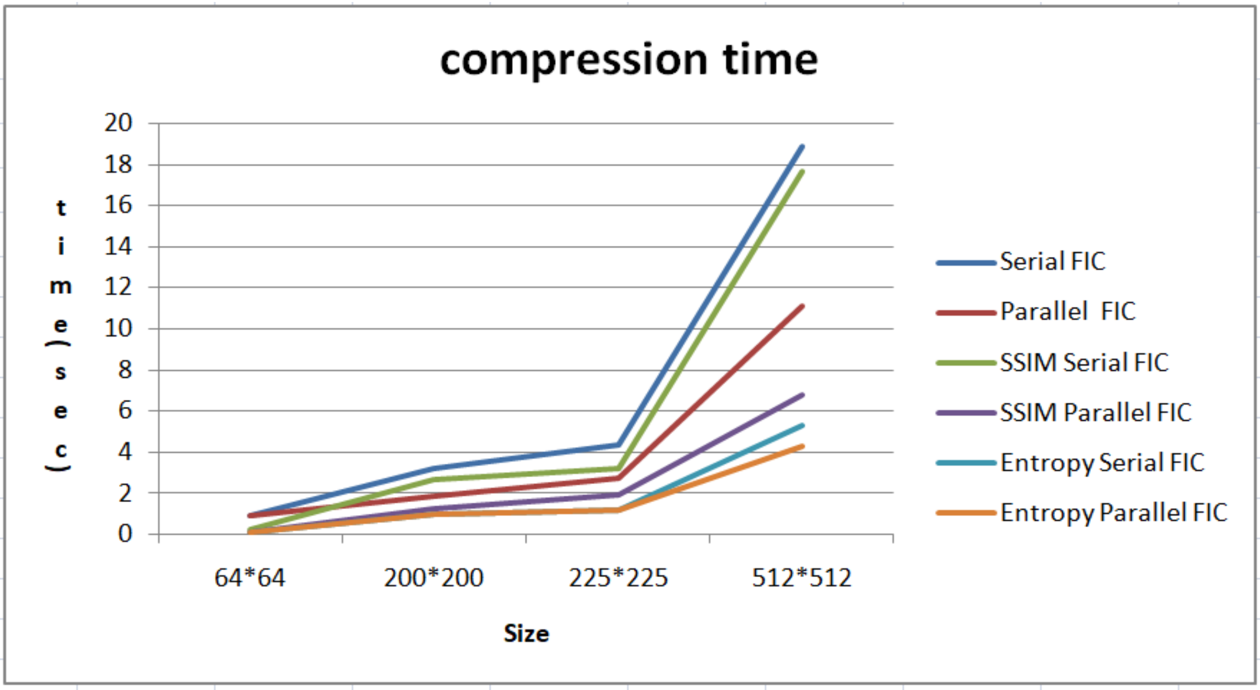



**Fig. 4:** Comparison of compression time

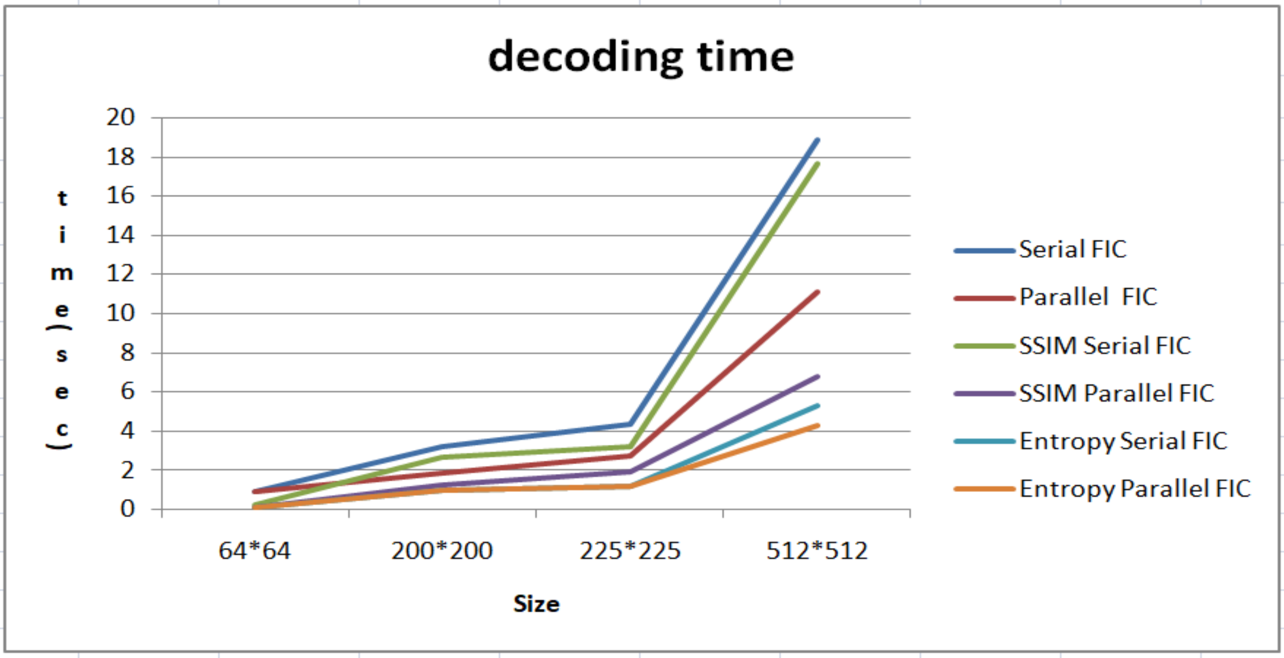



**Fig. 5:** Comparison of decoding time

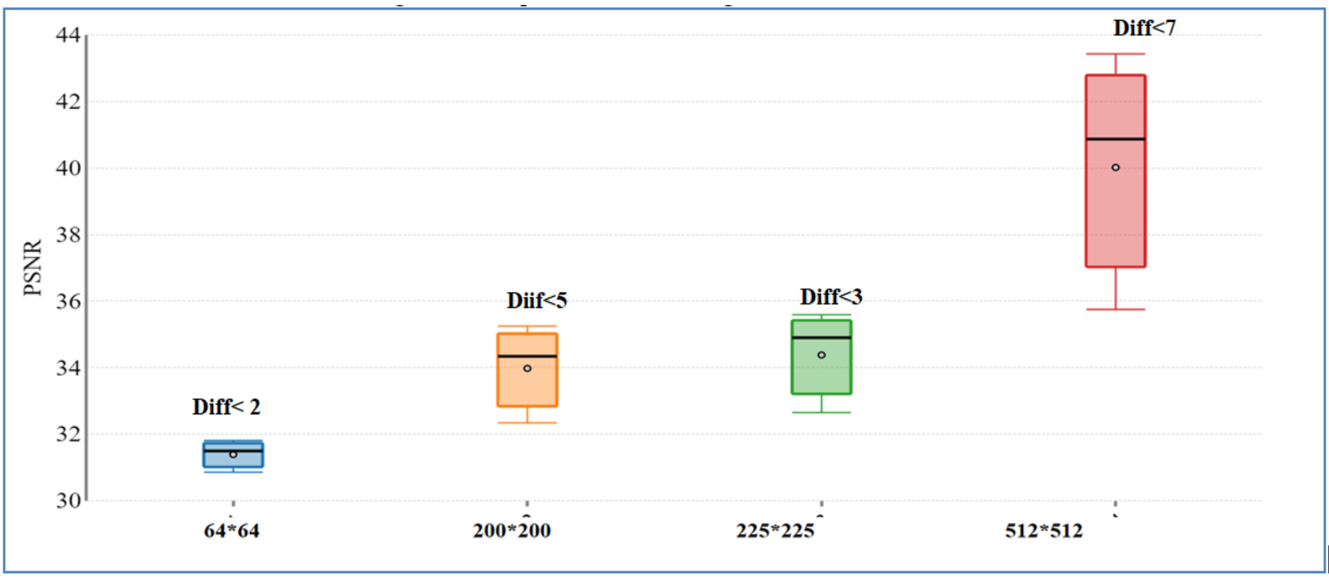



**Fig. 6:** Comparison of PSNR

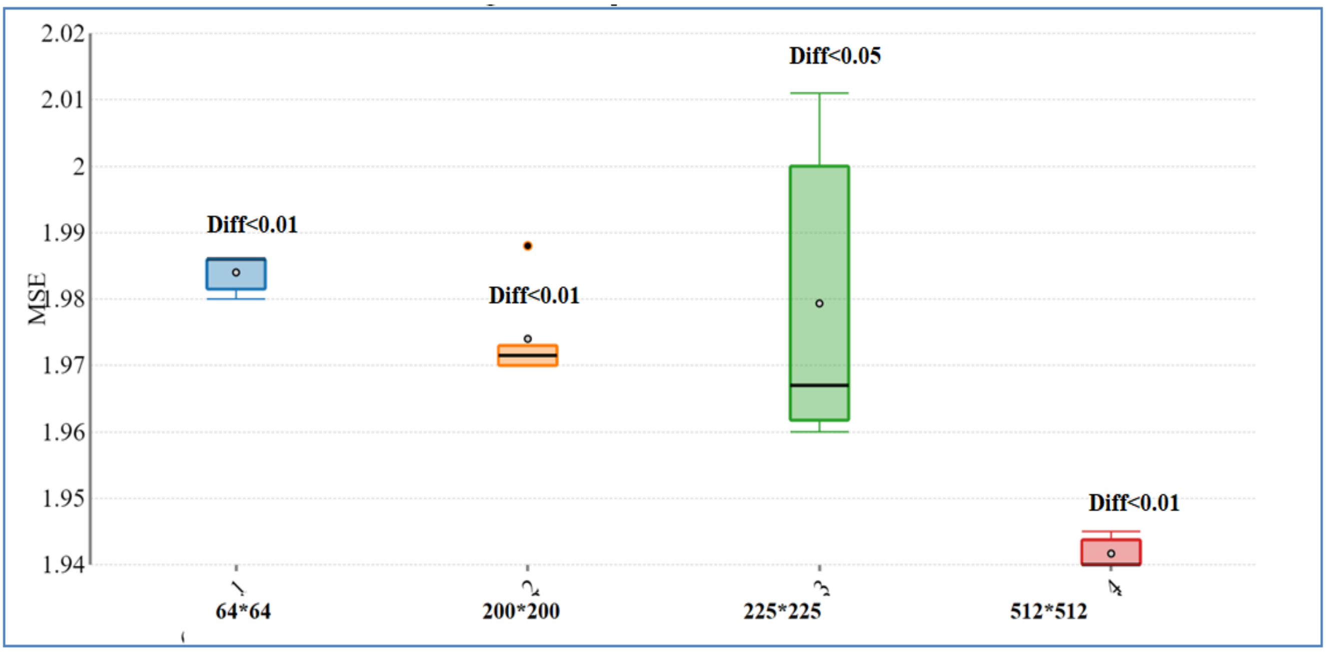



**Fig. 7:** Comparison of MSE

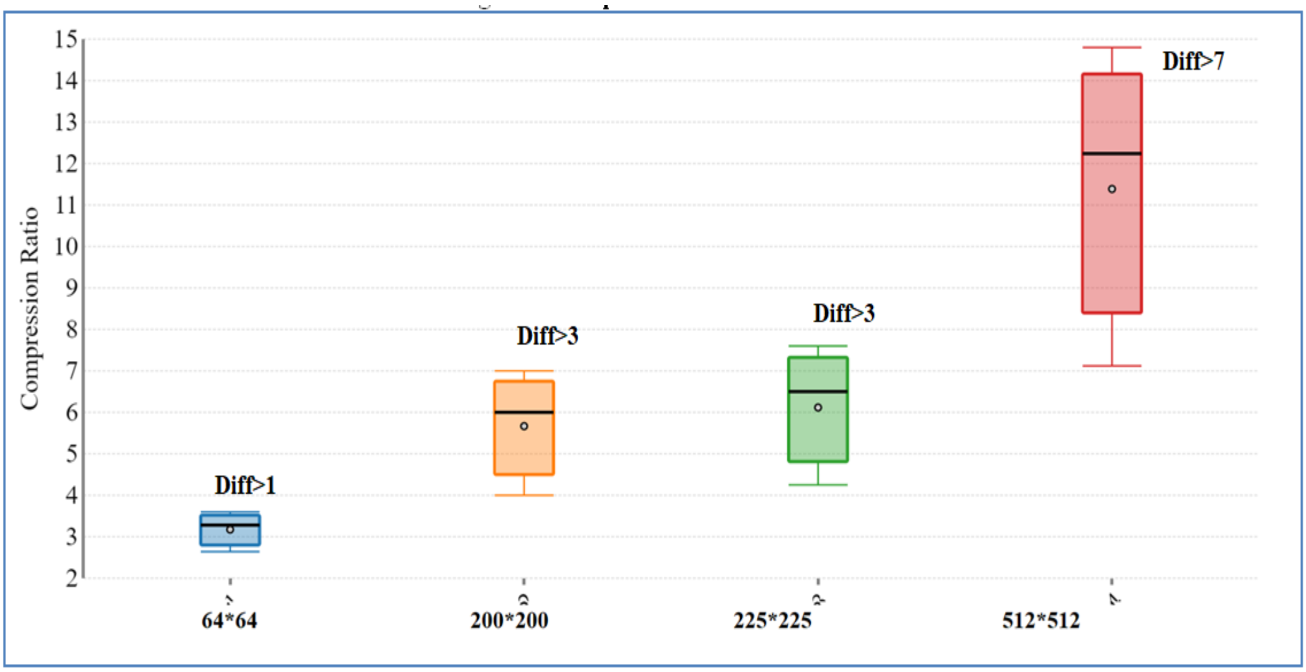



**Fig. 8:** Comparison of compression ratio

*Conclusion*

In this work, a hybrid solution combining encoding space reduction and parallelism is proposed to reduce the compression time without much impact on image quality and compression ratio. The solution used entropy/SSIM metric-based grouping of blocks in image and replacing the transformation for similar blocks with transformation representative block. This along with multi thread-based FIC reduced the compression time. The performance test results with different image resolutions proved there is not much distortion to quality and compression ratio with the proposed solution. Exploring different metrics in line of entropy, SSIM etc. for grouping similar blocks and representative block selection using error differences are in scope of future work.

## References

Aljanabi, M. A., Hussain, Z. M., & Lu, S. F. (2018). An Entropy-Histogram Approach for Image Similarity and Face Recognition. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2018/9801308.

Al-Saidi, N., & Abdulaal W. (2015). An Improved Differential Box Counting Method to Estimate Fractal Dimension. *Eng. & Tech. Journal, 33*(4), 714-722.

Asati, R., Raghuwanshi, M., & Singh, K. R. (2022). Fractal Image Coding-Based Image Compression Using Multithreaded Parallelization. In *Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT*: *Applications and Social Interfaces* (pp. 559-569). Singapore: Springer Nature.

Chen, D., & Singh, D. (2013). Fractal video compression in OpenCL: An evaluation of CPUs, GPUs, and FPGAs as acceleration platforms. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)* (pp. 297-304). IEEE.

Drakopoulos, V. (2013). Fractal-based image encoding and compression techniques. *Communications-Scientificic letters of the University of Zilina*, *15*(3), 48-55.

Fisher, Y. (1995a). *Fractal encoding of video sequences. Fractal image encoding and analysis*. Trondheim: ATO Advanced Study Institute on Fractal Image Encoding and Analysis.

Fisher, Y. (1995b). *Fractal Image Compression: Theory and Application*. New York: SpringerVerlag.

Girouard, G., Bannari, A., El Harti, A., & Desrochers, A. (2004). Validated spectral angle mapper algorithm for geological mapping: comparative study between QuickBird and Landsat-TM. In *XXth ISPRS congress, geo-imagery bridging continents, Istanbul, Turkey* (pp. 12-23).

Haque, M., Kaisan, A. A., Saniat, M. R., & Rahman, A. (2014). *GPU accelerated fractal image compression for medical imaging in parallel computing platform.* Retrieved from https://arxiv.org/abs/1404.0774

Hore, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. In *2010 20th international conference on pattern recognition* (pp. 2366-2369). IEEE.

Jacquin, E (1989). *A fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*. [PhD. Thesis]. Georgia Institute of Technology.

Lu, Y. (2019). The level weighted structural similarity loss: A step away from MSE. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), 9989-9990.

Mandelbrot, B. B. (1983). *The fractal geometry of nature*. San Francisco: Macmillan.

Palubinskas, G. (2017). Image similarity/distance measures: what is really behind MSE and SSIM?. *International Journal of Image and Data Fusion, 8*(1), 32-53.

Polvere, M., & Nappi, M. (2000). Speed-up in fractal image coding: comparison of methods. *IEEE transactions on Image Processing, 9*(6), 1002-1009.

Saad, A. M. H., & Abdullah, M. Z. (2018). High-speed fractal image compression featuring deep data pipelining strategy. *IEEE Access, 6*, 71389-71403.

Sankar, D., & Thomas, T. (2010). Fractal features based on differential box counting method for the categorization of digital mammograms. I*nternational Journal of Computer Information System and Industrial Management Applications, 2*, 9-11.

Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on image processing, 15*(2), 430-444.

Vallejos, R., Mancilla, D., & Acosta, J. (2016). Image similarity assessment based on coefficients of spatial asso-ciation. *Journal of Mathematical Imaging and Vision, 56*, 77-98. https://doi.org/10.1007/s10851-016-0635-y.

Wald, L. (2000). Quality of high resolution synthesised images: Is there a simple criterion? In *Third conference" Fusion of Earth data: merging point measurements, raster maps and remotely sensed images"* (pp. 99-103). SEE/URISCA.

Wang, Z., & Bovik, A. C. (2002). A universal image quality index. *IEEE signal processing letters, 9*(3), 81-84.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2* (pp. 1398-1402). IEEE.

Wohlberg, B., & Jager, G. d. (1999). A Review of the Fractal Image Coding Literature. *IEEE Transactions on Image Processing, 8*(12), 1716-1729. doi: 10.1109/83.806618.

Woon, W. M., Ho, A. T. S., Yu, T., Tam, S. C., Tan, S. C., & Yap, L. T. (2000). Achieving high data compression of self-similar satellite images using fractal. In *IGARSS 2000. IEEE 2000 International Geoscience and Remote Sensing Symposium. Taking the Pulse of the Planet: The Role of Remote Sensing in Managing the Environment. Proceedings (Cat. No. 00CH37120)* Vol. 2, (pp. 609-611). IEEE. doi:10.1109/IGARSS.2000.861646.