*International Journal of Engineering Applied Sciences and Technology, 2020*
**Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 579-585**
Published Online August 2020 in IJEAST (http://www.ijeast.com)

# PARALLELIZATION OF DIGITAL IMAGE WATERMARKING IN THE FREQUENCY DOMAIN WITH DCT

Shital P. Shinde
Department of Information Technology,
Walchand College of Engineering, Sangli
Maharashtra, India.

Dinesh B. Kulkarni
Department of Information Technology,
Walchand College of Engineering, Sangli
Maharashtra, India.

*Abstract*— **In today's digital world, in each area, there is the massive use of digital content. A huge amount of data can be transmitted or scattered easily with no cost, there are more chances of the copying of such digital information. So, there is a need to restrict such illegal copies of digital content. One of the solutions to this problem is digital watermarking. Frequency domain watermarking techniques use different transformation approaches like Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), and Discrete Wavelet Transform (DWT), etc. The parallelization techniques are used for DCT as it consists of various time consuming mathematical computations. The DCT transformation work on blocks of an image. This helps for parallelization as it does not contain dependency within work. In this paper, we have demonstrated the parallel execution of image watermarking, also the comparative analysis of parallel execution and sequential execution of image watermarking. We used parallel MATLAB for the parallel programming and then listed out all the parameters and their execution cost to be specifically determined.**

*Keywords*— **Parallel computing, Digital image watermarking, Frequency domain, Discrete Cosine Transform (DCT), General Purpose Graphics Processing Unit (GPGPU)**

## I. INTRODUCTION

With the fast improvement in digital multimedia technology, preventing the copyright of digital media has become almost essential. The dominant solution for this is Digital Watermarking [1]. Digital watermarking is a competent approach to data hiding. The two main domains are there in which digital image watermarking works, spatial domain, and frequency domain. In this paper, the main aim is to figure out the research work in the transform domain [2]. Image transformation is needed to get more information about the image. Even though the frequency domain technique takes more time and is more complex in terms of computations as a correlate to spatial domain techniques, the embedded watermark cannot get identify easily using frequency domain techniques [3]. The time required for Digital Image Watermarking in the transform domain is reduced by parallel computing. With the sequential implementation, it takes more time. Better results will be established with multi-core and many-core environment.

Even if there are different methods for inserting watermarks in digital images that have been discussed within the literature, very few techniques support parallelization with GPUs.

### A. Digital Image Watermarking –

Digital Watermarking is a technique that is used in the digital signal processing of embedding hidden information into multimedia data. This information is not usually visible, only a dedicated detector or extractor can see and extracts that information. Digital Image Watermarking uses the digital image for embedding the hidden information, after embedding the watermarked image is generated and the watermarked image is more robust against attacks. The frequency-domain is getting much more attention due to the use of wavelets which have a high degree of resemblance to the human visual system. In digital watermarking, secret information is embedded with original data for maintaining ownership rights of the digital content. Spatial domain watermarking techniques work over pixel characteristics and frequency domain watermarks concerned about different transformations that can be used with digital content. Imperceptibility, robustness, security, complexity, and capacity are some requirements of digital watermarking which completely depends on the algorithm used for watermarking.

### B. Multi-Core CPU's-

In a multi-core CPU, the number of available resources is limited; it is the number of available cores. CPU cores are computationally powerful tools designed to handle sequential codes efficiently. The recent advances in CPU architectures enable multiprocessing through multi-core CPU. The entire core of the CPU can access the main memory. Also, concurrent programming for multi-core CPUs is affordable due to the existence of different parallel programming libraries such as OpenMP. Parallelization can be done using Matlab also.

### C. Many-Core CPU's-

The utilization of GPU for speeding up the sequential code is successfully reported in different directions. In several works, the parallel code execution-time is less than the sequential counterpart by two or three orders of magnitude. GPU has hundreds or thousands of less powerful cores, in comparison to CPU cores. GPU is designed to provide a high level of parallelism through several physical streaming multiprocessors (SMs), where each SM consists of tens or hundreds of physical cores. The cores are managed to perform synchronized tasks at a high level. The purpose of supplying a GPU with many SMs is to guarantee a high level of concurrency where each SM has its code control unit, registers, and shared memory. Also, the global memory space the entire SMs can access it.

## II. LITERATURE REVIEW

The following literature survey provides the basis for the research in digital image watermarking which contains frequency domain techniques like DCT. Also, it gives information about how parallel computing plays an important role in watermarking.

**Caiwei Lin, Lei Zhao, and Jiwen Yang [7][13]** propose an implementation of an image authentication scheme on CUDA-enabled GPU, which has been widely equipped in today's PCs. Experimental results show that the author's CUDA-based method for image authentication gains a good performance in both authentication results and operation speedup. The elaborate on the implementation based on CUDA for the image authentication algorithmic program. As compared to the sequential performance in CPU, their implementation with CUDA gives 19- 36x speedup. The experimental outcome shows that the performance using CUDA gives a pleasing achievement not only in authentication results but also in operation speedup. DCT transform is centralized by a huge amount of calculations, which highly point towards parallel processing. As per methodology, implementation of CUDA contains the serial approach of two kernels for the formation of the watermarked image. One kernel performs DCT transformation while another one is used to perform IDCT transform to get watermarked data. The CPU implementation was performed with the single-threaded Clanguage. In the experiment, three BMP format images are used to show the effect of proposed work on CPU and GPU. The work of CUDA is examined by calculating the time required for CPU and CUDA version. As the size of the image increases, the CUDA version also performs well. The speedup of this CUDA implementation ranges from 19x for 512*512 image to 36x for 8192*8192. Their focus is only on improving performance.

**Caiwei Lin, Lei Zhao, and Jiwen Yang [8]** extend their previous work and gives the result for parallel computing on GPU for fragile watermarking in the frequency domain. The CUDA is used to get high performance, flexibility in the image authentication field. The change in this work is that they try to improve the security of algorithms by using a Logistic chaotic model and make a unique serial number for each block. This work tries to utilize more GPU as compared to the previous one. The proposed CUDA implementation achieves speedup which ranges from 20x to 50x. The imperceptibility which is measured by PSNR gives the better output as correlated to the previous one. The speedup of a single GPU with CPU is 32x while the speedup of dual GPU with CPU is 40x. The result shows that the performance of dual GPU gives better results as compare to single GPU and CPU.

**Mohammadabadi and A. Chalechale [9]** proposed a contribution of the General-Purpose Graphics Processing Unit (GPGPU) in watermarking. Here, they choose a color DCT watermarking algorithm recommended by Su et al. [10] to perform parallelization with the help of CUDA. Su's algorithm is considered as the very latest algorithm for watermarking using DCT that exceeds the earlier color watermarking approach in case of robustness and invisibility. But, this Su's technique has more computational complexity because of its time-exhausting steps like color space transforming for the original image, DCT performing for blocks for host image and watermark image. To trim this computational complexity, they proposed this contribution in sequential implementation. They modify Su's algorithm to achieve more quality and low complexity. With this approach, complexity is decreased from O(N4 ) to O(N2 ). It is examined that, As relate to Su's method [10], the PSNR value is improved up to 21.99%. Authors achieve speedup using GPU for various image sizes. The drawback is that there is a significant reduction in speedup for both GPUs as the size of the image is increased.

**García Cano et al.[11]** have proposed a system to examine a large number of digital images especially in grayscale JPG format, to check the original content with the help of the watermarking approach. Author focus on exploiting a GPU to enhance this approach. In the experiment, there is no need for the host image during the extraction of the watermark is done. García Cano et al. refer to the algorithm suggested by Shieh et al.[12] for sequential implementation to embed and extract the watermark. The author used an input of size 512x512, it is desirable to break down it into 64x64 blocks as in the DCT. The 64x64 matrix is easily profiled to the equal count of blocks in the GPU, and the composition of the threads will be based on activities to be completed. The speedup of GPU is 91.80, where the time is required for data transfer from CPU to GPU is excluded. If the time essential for data condition is considered, then the speedup is 6.01. But this proposed work not concentrates on how to increase the quality of the watermarking algorithm. It only focuses on the acceleration of performance. García Cano et al.[11] accelerate the watermarking algorithm which is elaborated by Shieh et al.[12] by many-core architecture.

**Hong et al.[13]** elaborate the watermarking in any one channel of RGB image using the frequency-domain technique.

In their experiment, it is found that the time required for Tesla K20 GPU to perform watermark embedding for one image is less than 0.1 ms and this result is 170 times faster than the Intel Core i7-3820 CPU. The Authors follow the approaches described in[14]. Hong et al.[13] do not use an additional watermark to embed in host image, instead of that, a part of coefficients in the frequency domain is altered. Initially, authors chose one channel in which watermarking is done. After that, divide that original channel into blocks and that is converted into the frequency domain. Hong et al.[13] choose 256 ( 16*16) blocks on the top left corner for operations. Hong et al.[13] multiply the top left coefficient of each block by 1.01. Simply means, authors modify the coefficients of the channel. And then perform IDCT to come back in the spatial domain. And finally, combine that watermarked channel with the remaining two channels of the host image.

**D. Bollman, A. Bustillo, and E. Morales [15]** elaborate parallel computing for a different group of watermarking algorithms in the transform domain and also analyze the results of sequential, OpenMP, MPI, and CUDA implementations of image watermarking. There is no previous work involving application on OpenMP, MPI. The authors compare all the results in terms of a speedup as per the increase in the size of the image. The result shows that, for a single node, CUDA is 287 times rapid as compared to the sequential, 42 times than OpenMP, and 24 times than MPI. Authors also apply software StriMark to check the robustness for algorithm and then calculate PSNR and NC value.

**P. Lenarczyk and Z. Piotrowski [16]** introduces a new Watson’s visual model which is used for checking imperceptibility of images. Watson’s visual model is based on a 2D-DCT and it is a blockoriented approach. However, it aims to notify the difference between the two images. The authors calculate the robustness by applying different attacks in the watermarked image. Attacks are mostly divided into two types as unintentional and intentional attacks. These attacks are maybe geometrical or image processing attacks. It gives high robustness to attacks of desynchronization, different filtering, noise, etc.

**H. Khor, S. Liew, and J. Zain [17]** represent the parallelization of ultrasound medical image watermarking. According to their proposed work, authors embed the watermark into an ultrasound medical image and give that watermarked image as input for the authentication process. The sequential performance of the watermarking of frames is time-consuming and that problem is overcome by a parallel implementation. To achieve this, the ultrasound frames are divided into tasks and assigned to each core respectively, and then watermarking will be performed in each one concurrently. In the parallel implementation, the total frames in ultrasound image are distributed within the cores of processor equally.

## III. METHODOLOGY

The proposed methodology contains the transformation domain-based Digital Image Watermarking technique like Discrete Cosine Transform (DCT). Adding a digital watermark in the frequency domain means applying watermarking steps on transformed data of an image. For that, different transform techniques are used for transformation from a spatial domain to frequency domain such as Fourier transform, discrete cosine transform, wavelet transform, Polar Transform, etc. So, Digital Image Watermarking can be done via different transform technique and all techniques have their advantages and disadvantages.

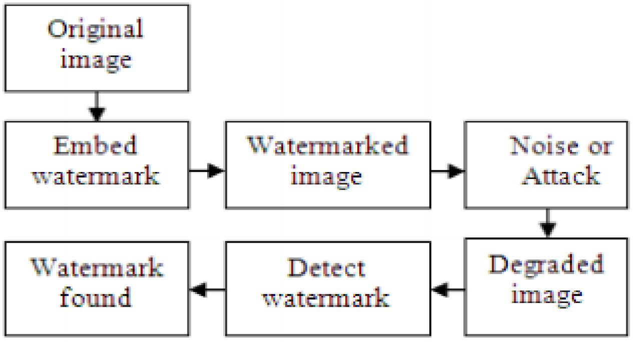


Fig 1.Typical watermarking block diagram

In the Watermarking scheme, a watermark is employed into the original image, to authenticate the host [18]. Fig (1) reveals the step by step process of embedding watermark into the original image. Later on, because of noise or other attacks degraded image Digital Image Watermarking uses the digital image for embedding the hidden information, after embedding the watermarked image is generated and the watermarked image is more robust against attacks.

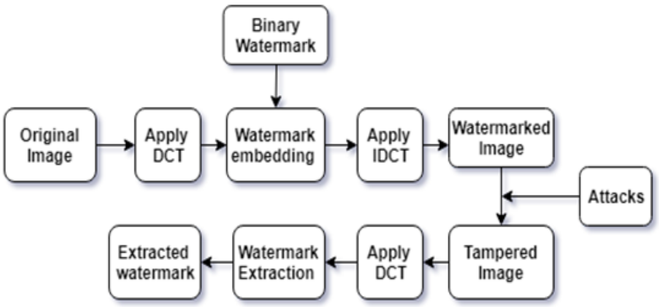


Fig 2. Block diagram of Digital Image Watermarking using DCT

Working of Digital Image Watermarking can be divided into three stages and it explained properly with an example:

### 1. Embedding Stage

The embedding stage is the first stage in which the watermark is embedded in the original image by using the embedding algorithm and the secret key. Then the watermarked image is generated. So the watermarked image is transmitted over the network.

i. Apply DCT on an image with $8 \times 8$ blocks and save these blocks for further process.

ii. Select the required number of blocks to embed the watermark.

For example:-

Input image size:- $512 \times 512$

$8 \times 8$ DCT blocks:- $\left(\frac{512}{8} \times \frac{512}{8}\right) = 4096$

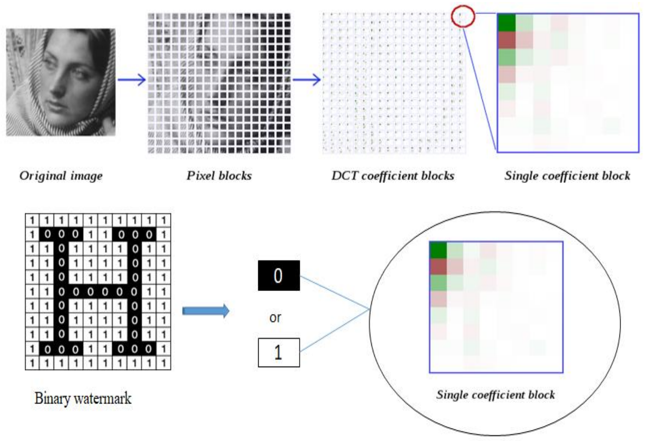
& suppose,

Binary watermark size:- $48 \times 48$

So, total no of bits:- $(48 \times 48) = 2304.$

So, out of 4096 blocks, only 2304 blocks are used to embed the watermark.

iii. Then, embed each bit into a particular coefficient of the corresponding block.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Fig 3. Steps of embedding watermark

iv. After embedding bit, apply IDCT to blocks and recombine these blocks to get a watermarked image.

v. Peak Signal to Noise Ratio (PSNR) is calculated to check the imperceptibility of an embedded image.

### 2. Distortion/Attack Stage

In this stage, when the data is transmitted over the network. Either some noise is added with the watermarked image or some attacks are performed on the watermarked image.

i. The attacks are both types, intentional and non-intentional

ii. To check the robustness, intentionally some noise is added into the embedded image.

iii. Applied Noise- salt & pepper with noise density 0.003.

iv. The attacked image is transmitted for the extraction process.

### 3. Detection/Extraction Stage

In the detection stage, the watermark is detected or extracted by the dedicated detector from the watermarked image by applying some detection algorithm. The watermarked image is given as input for the extraction process. The location of the watermark embedding is used during the extraction process.

i. Apply DCT on the embedded image.
ii. Check the DCT values for the extraction of a watermark.
iii. Store the extracted bits into the vector to get extracted watermark.
iv. These extracted bits are mapped together into $48 \times 48$ size to get extracted watermark.

**Parallel computing on many-core architecture**

Matlab is one of the most widely used mathematical computing environments in technical computing. It has an interactive environment that provides high-performance computing (HPC) procedures and easy to use. Parallel computing with Matlab has been an interesting area for scientists of parallel computing researches for several years. Where there are many attempts to parallel Matlab.

Parallel computing is one of the computing methods which execute much computation (processes) simultaneously. Where the principle of parallel computing is often can be divided the large problem into smaller pieces, then solved that concurrently ("in parallel")[19].

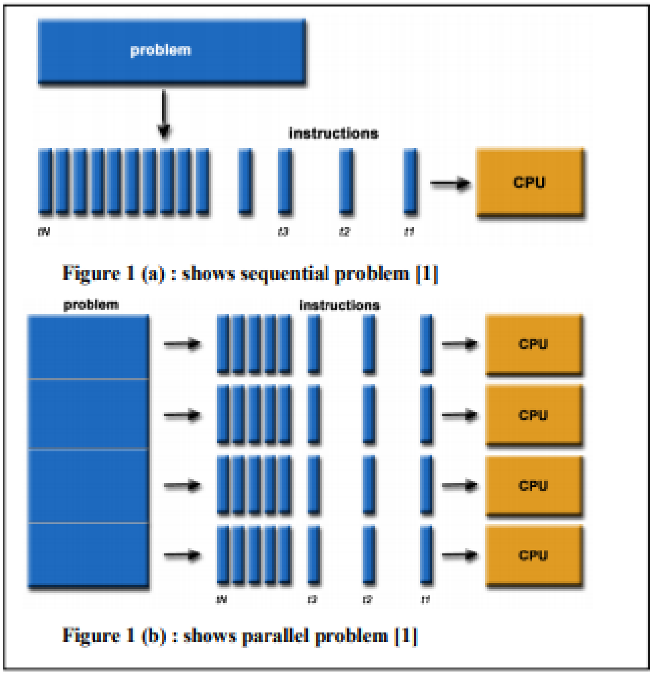

problem
instructions
CPU
tN
t3
t2
t1
Figure 1 (a) : shows sequential problem [1]
problem
instructions
CPU
CPU
CPU
CPU
tN
t3
t2
t1
Figure 1 (b) : shows parallel problem [1]

Fig 4. Parallel problem-solving approach in Matlab

**Experimental Setup**

The Digital image watermarking is implemented on the PC which consists of the following specifications.

| | | |
|---|---|---|
| Processor | : | Intel(R) Core(TM) i7-4790 CPU @3.60GHz |
| RAM | : | 12 GB |
| Operating System | : | Ubuntu 18.04 LTE. |
| MatLab | : | Matlab R2019b |
| GPU | : | NVIDIA GeForce GT 730 |

The actual implementation of CPU and parallelization is done using MatLab. For the sequential implementation of digital image watermarking, the image processing toolbox is used for performing various operations on the image. While the parallel implementation of digital image watermarking is done on the same platform using the Parallel computing toolbox which provides the parallel working as per the number of workers.

## IV. EXPERIMENTAL RESULTS

### A. Original Image-

There are different size images are used to perform image watermarking to check the time required for image watermarking. As the size of image increases, the time required for process is also increases. The cameraman image is used as main image with different sizes as 512*512, 1024*1024, 2048*2048, 4096*4096, etc. The binary watermark is used with the size 48*48. To check the imperceptibility and robustness, the intentional noise is added in the watermarked image which is salt and pepper noise with density 0.003.

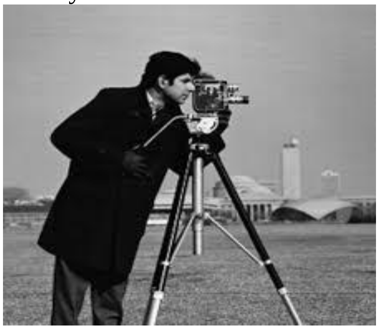


Fig 5(a)Original Image

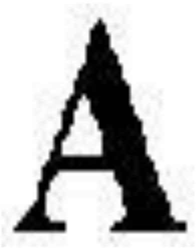
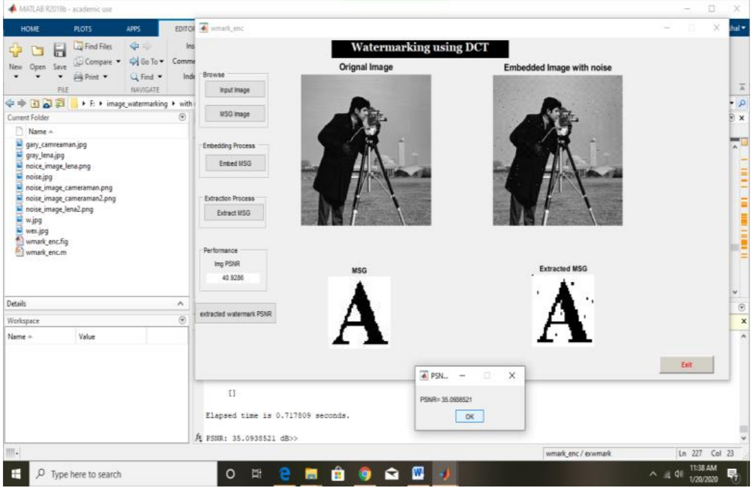


Fig 5(b)Binary Watermark

### B. Sequential Image watermarking

a) Digital image watermarking on 512*512 image



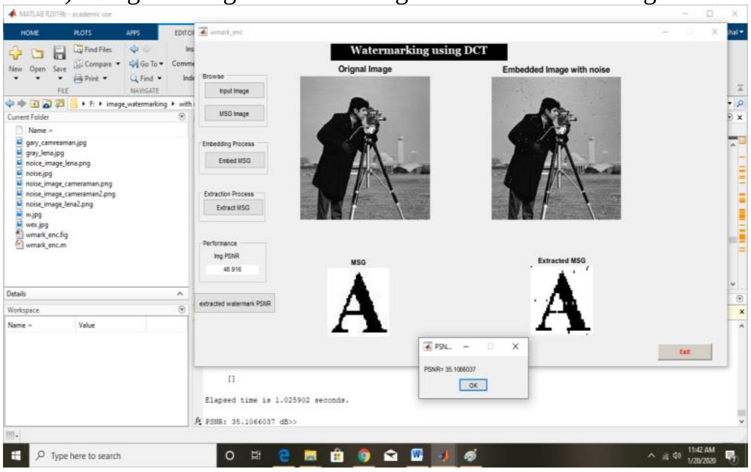
PSNR= 35.0938521

```
    []

Elapsed time is 0.717809 seconds.
PSNR: 35.0938521 dB>>
```

Fig 6. Digital Image Watermarking on 512*512 size image

b) Digital Image Watermarking on 512*512 size image



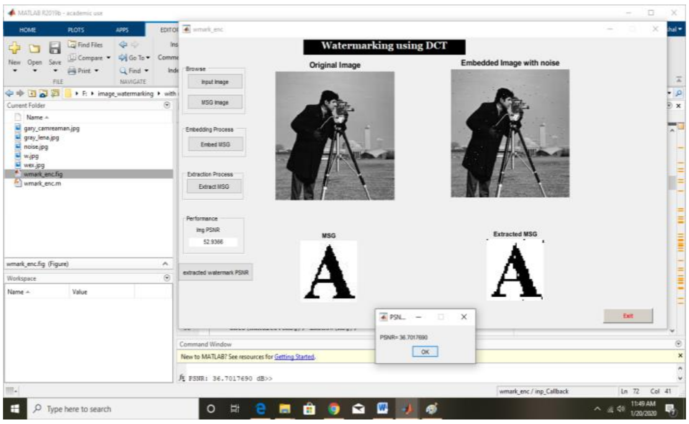
PSNR= 35.1066037

```
    []

Elapsed time is 1.025902 seconds.
PSNR: 35.1066037 dB>>
```

Fig 7. Digital image watermarking on 1024*1024 size image

c) Digital image watermarking on 2048*2048 image



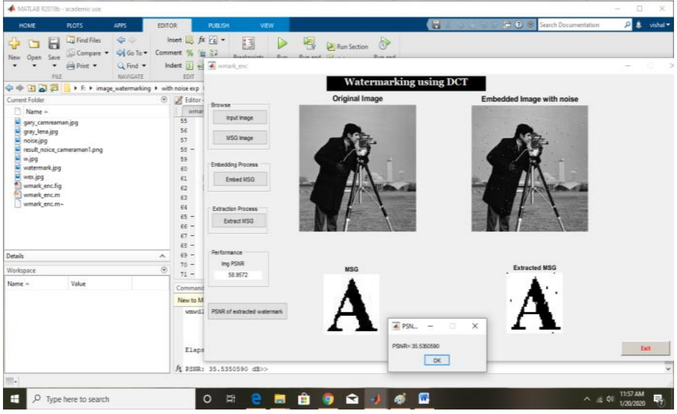
52.9366

PSNR= 36.7017690

```
PSNR: 36.7017690 dB>>
```

Fig 8. Digital image watermarking on 2048*2048 size image

d) Digital image watermarking on 512*512 image



58.9572

PSNR= 35.5350590

```
Elaps
PSNR: 35.5350590 dB>>
```

Fig 9. Digital image watermarking on 4096*4096 size image

### C. Parallel digital image watermarking

The same results are generated in matlab by using parallel computing toolbox. Main aim is to reduce time required for image watermarking on large size images.

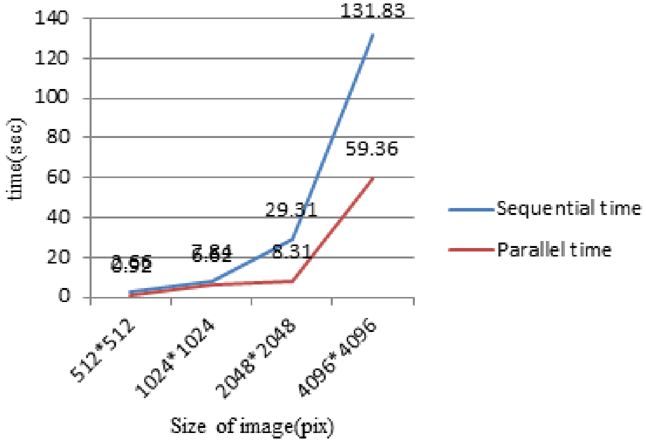


Fig 10. Comparison of execution time required for image watermarking

### D. Analysis

The below table shows the analysis of digital image watermarking in parallel and sequential domain using DCT. It also gives the detail information about speedup for different size of images.

Table 1. Analysis of digital image watermarking

| Size of image | Sequential time(sec) | Parallel time(sec) | Speedup |
|---|---|---|---|
| 512*512 | 2.66 | 0.92 | 2.89x |
| 1024*1024 | 7.84 | 6.62 | 1.18x |
| 2048*2048 | 29.31 | 8.31 | 3.52x |
| 4096*4096 | 131.83 | 59.36 | 2.22x |

## V. CONCLUSION

This study presents a new embedding technique for image watermarking using Discrete Cosine Transformation. A watermark image is not directly embedded into a host image, rather, certain coefficients get modified as per the bit of watermark. The watermark image is generally binary. The proposed technique also provides distorted watermark extraction under image-noise attack. The proposed scheme also proves the superiority in terms of robustness under multiple combinations of attacks. The main work is parallelization of image watermarking on many-core architecture to improve the performance. This work also compares the time required for image watermarking on different sizes of images.

## VI. FUTURE SCOPE

The work carried out by H. Khor, S. Liew, and J. Zain [17] on the parallel digital watermarking pattern will be more enhanced with the help of the graphics processing unit (GPU). The CPU and GPU both have the feature to run thousands of threads simultaneously, but the GPU has awesome results as compared to the CPU because it has the number of cores processed as compared to CPU. The existing work is mostly for image processing attacks like filtering, noise, etc. and use common techniques like DCT, DFT, etc. In the future, the work may focus on geometrical attacks with the help of a polar transform. This digital image watermarking will be done on printed material, halftone images, etc

## VII. ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks to all the staff members of the Information Technology Department for their help whenever required. Finally, I express my sincere thanks to all those who helped me directly and indirectly throughout this dissertation phase work.

## VIII. REFERENCES

[1] Hosny. K. M, Darwish M.M and K. Li and Salah A.(2018). Parallel Multi-Core CPU and GPU for Fast

and Robust Medical Image Watermarking *in IEEE Access*, vol. 6, pp. 77212- 77225.

[2] Shehab et al.(2016,2017,2018). Secure and Robust Fragile Watermarking Scheme for Medical Images *in IEEE Access*, vol. 6, pp. 10269-10278 .

[3] Kaur A. and Singh J.(2017). Digital image watermarking techniques: a review *International Journal of Advanced Research in Computer Science*, Vol 8, pp 42113.

[4] Chandrakar N. and Baggaa J.(2013). Performance Comparison of Digital Image Watermarking Techniques: A Survey International Journal of Computer Application Technology and Research, vol. 2, no. 2, pp. 126-130.

[5] Mistry D.(2010). Comparison of Digital Watermarking Methods International Journal on Computer Science and Engineering, vol. 02, no. 09, pp. 2905-2909.

[6] Fridrich J.,Goljan M., and Du R.(2001). Invertible Authentication," in Proc. SPIE Conf. Security and Watermarking of Multimedia Contents III San Jose, Calif, USA, vol. 43, no. 14, pp. 197–208, January.

[7] Lin C.,Zhao L., and Yang J.(2010). A CUDA Based Implementation of an Image Authentication Algorithm IEEE.

[8] Lin C.,Zhao L., and Yang J.(2011). A High-Performance Image Authentication Algorithm on GPU with CUDA .J. Intelligent Systems and Applications.

[9] Mohammadabadi A. and AChalechale M.(2016). Parallelization of a Color DCT Watermarking Algorithm using a CUDA-based Approach 6th International Conference on Computer and Knowledge Engineering, IEEE.

[10] Su Q., Niu Y.,Liu X. and Yao T.(2015). A Novel Blind Digital Watermarking Algorithm for Embedding Color Image into Color Image Optik - International Journal for Light and Electron Optics.

[11] Cano G. et al.(2010,2011,2012). A parallel Watermarking application on a GPU International Congress of New technologies and Trends in Education.

[12] Shieh et al.(2002,2003,2004). Genetic watermarking based on transform-domain techniques Pattern Recognition.

[13] Fan H.et al.(2014,2015,2016). Accelerating DCT-based color image watermarking on GPUs Proc. IEEE Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV).

[14] Obukhov A. and Kharlamov A.(2012). Discrete cosine transform for 8x8 blocks with CUDA White paper V1.0.

[15] Bollman D.,Bustillo A., and Morales E.(2015). Parallel Watermarking Of Images In The Frequency Domain Scalable Computing: Practice and Experience Volume 16, Number 2.

[16] Lenarczyk P.and Piotrowski Z.(2013) Parallel blind digital image watermarking in spatial and frequency domains Faculty of Electronics, Military University of Technology.

[17] Khor H., Liew S., and Zain J.(2016). Parallel Digital Watermarking Process on Ultrasound Medical Images in Multicores Environment International Journal of Biomedical Imaging.

[18] Mingwei Z., and Dang Y.(2008). Color image copyright protection digital watermarking algorithm based on DWT & DCT Wireless Communications, Networking, and Mobile Computing, 2008. WiCOM'08. 4th International Conference on. IEEE.

[19] Alyasseri Z.(2014). Survey of Parallel Computing with MATLAB. University Sains Malaysia.

[20] Sharma G., and Martin J.(2009). MATLAB®: A Language for Parallel Computing Springer, International Journal Of Parallel Programming, Volume 37.

[21] Michael E. Thomadakis(2011). Introduction to GPU Architectures for HPC Computing. Supercomputing Facility Texas A&M University.

[22] Messmer P.(2010). Introduction to GPU Computing. Tech- Tech-X Corporation.