iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VI **Month of publication:** June 2017

DOI:

www.ijraset.com Volume 5 Issue VI, June 2017
IC Value: 45.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Comparative Study of Some Image Compression Techniques

Nitin Ingle[1]
[1]*Department of Electronics, Dr. Annasaheb G. D. Bendale Mahila Mahavidyalaya, Jalgaon 425001, India*

Abstract: A remarkable progress has been made in the field of data compression and its application, digital image compression is associated with removing redundant information of image data. Compression of digital images has a two-fold advantage. It reduces transmission time, bandwidth and requires less amount of storage memory. The comparative study of image compression techniques presented here are Huffman Coding, Predictive Coding, Block Truncation Coding and the Wavelet Transformed based SPIHT technique. The GPU computing technique and two different GPUs are used to speed up the designed algorithms.
Keywords: lossy, lossless, tree coding, cuda, gpu computing.

I. INTRODUCTION

With the continuing growth of modern communications technology, demand for digital image transmission and storage is increasing rapidly. Digital technology has given a great benefit to the area of image processing. Image compression is an economically viable alternative to these constraints by reducing the bit size required to store and represent the images, while maintaining relevant information content. The improvement of computer hardware including processing power and storage power has made it possible to utilize many-sophisticated signal processing techniques in advanced image compression algorithms[1]. Huge amount of data is required to represent digital images, to decrease the requirement of the memory space for storage the digital images and to alter the bandwidth requirement for transmission of images, it is essential to compress an image efficiently. In the data compression, the number of bits is reduced to store or transmit data. Data compression can be lossy or lossless. Data compressed using lossless techniques can be decompressed to exactly its original value, whereas lossy data compression techniques discards the partial data to represent the content. Image compression addresses the problem of reducing the amount of data required to represent a digital image. Data redundancy is a central issue in digital image compression. For the comparative study algorithms of Huffman Coding, Predictive Coding, Block Truncation Coding and Wavelet Transformed based SPIHT technique are developed, the algorithms for all above techniques are designed and implemented on both the INTEL and NVIDIAs processors.

II. IMAGE COMPRESSION TECHNIQUES

There are varieties of techniques that can be used to compress digital images. These techniques can be lossless or lossy. In lossless compression (also known as bit-preserving or reversible compression), the reconstructed image after compression is numerically identical to the original image on pixel-by-pixel basis, in lossy compression (also known as irreversible compression), the reconstructed image contains degradation relative to the original, this results in higher compression as compared to the lossless compression[2]. Some of techniques which have been presented in this work are as given below.

A. Huffman Coding

It is lossless compression technique based on the principal to encode the more probability data by the less bit code. Huffman coding is a classical data compression technique. It has been used in various compression applications. It uses the statistical property of characters in the source stream and then produces respective codes for these characters[1]. These codes are of variable code length using an integral number of bits. The codes for characters having a higher frequency of occurrence' are shorter than those codes for characters having lower frequency. This simple idea causes a reduction in the average code length, and thus the overall size of compressed data is smaller than the original. Huffman coding is based on building a binary tree that holds all characters in the source at its leaf nodes, and with their corresponding character's probabilities at the side. Step-by-step combination of nodes results in code tree structure, using bottom-up procedure the algorithm combines two nodes. The new interior node gets the sum of frequencies of both child nodes. Huffman codes assign every symbol to a leaf node of a binary code tree. These nodes are weighted by the number of occurrences of the corresponding symbol called frequency or cost. The binary values 0 and 1 according to the rules for common prefix-free code tree are assigned to the branches of tree. The particular code word is represented by the path from the root tree to the corresponding leaf node defines. Huffman codes are prefix-free binary code trees, therefore all substantial considerations apply accordingly. Codes generated by the Huffman algorithm achieve the ideal code length up to the bit boundary. The maximum deviation is less than 1 bit.

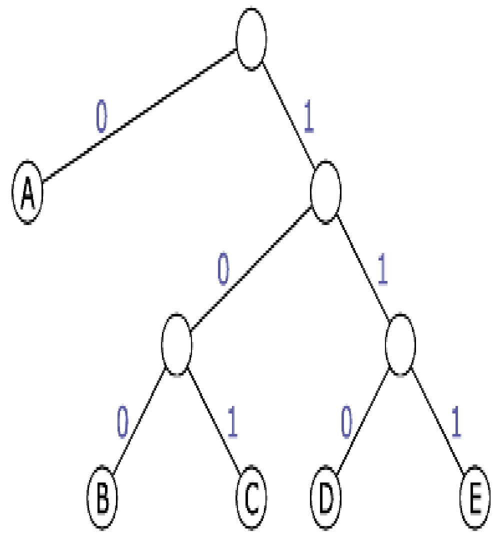


Fig. 1 Huffman Code Tree

B. Predictive Coding

Lossless predictive coding predicts the value of each pixel by using the value of its neighbouring pixels. Therefore, every pixel encoded with a prediction error rather than its original value. These errors are much smaller compared with original value so that fewer bits are required to store them. Such as DPCM (Differential Pulse Code Modulation) is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element. A variation of the lossless predictive coding is adaptive prediction that splits the image into blocks and computes the prediction coefficients independently for each block to high prediction performance[4]. Most images are having redundancies among adjacent pixel values. Thus, a current pixel can be predicted reasonably well based on a neighborhood of pixels. The prediction is obtained by subtracting the prediction from the original pixel, has smaller entropy than the original pixels. Hence, the prediction error can be encoded with fewer bits. In the predictive coding, the correlation between adjacent pixels is removed and the remaining values are encoded.

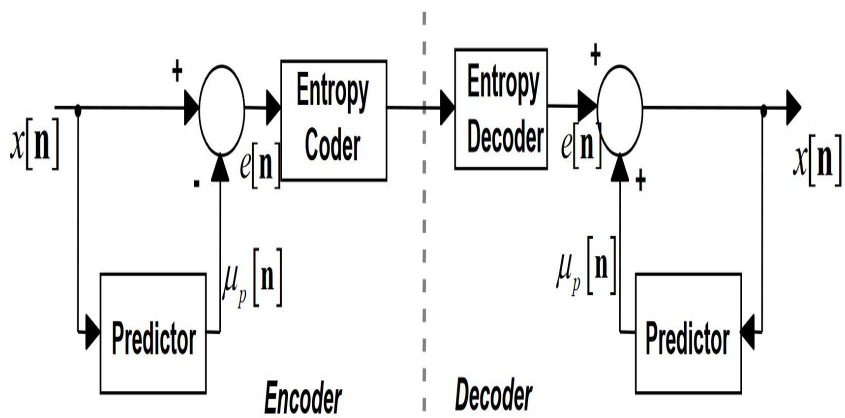


Fig. 2 Predictive Coder and Decoder

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Prediction µp[n] is calculated for x[n] for previous samples XN+n , e[n] is prediction error. Entropy coding is a type of lossless coding to compress digital data by representing frequently occurring patterns with few bits and rarely occurring patterns with many bits.

C. Block Truncation Coding

Block truncation coding (BTC) is a lossy moment preserving quantization method for compressing digital gray-level images. Its advantages are simplicity, fault tolerance, the relatively high compression efficiency and good image quality of the decoded image[5]. In Block Truncation Code input image is segmented into NxN blocks of pixels, a two level output is chosen for every block and bitmap is encoded using one bit for every pixel. The main idea is that the two levels are selected independently for every NxN block. So, encoding the block is a simple binarization process because every pixel of the block is truncated to 1 or 0 binary representation choosing the reconstruction level. However, BTC has a fundamental limit in that each block is reconstructed by only two representative values

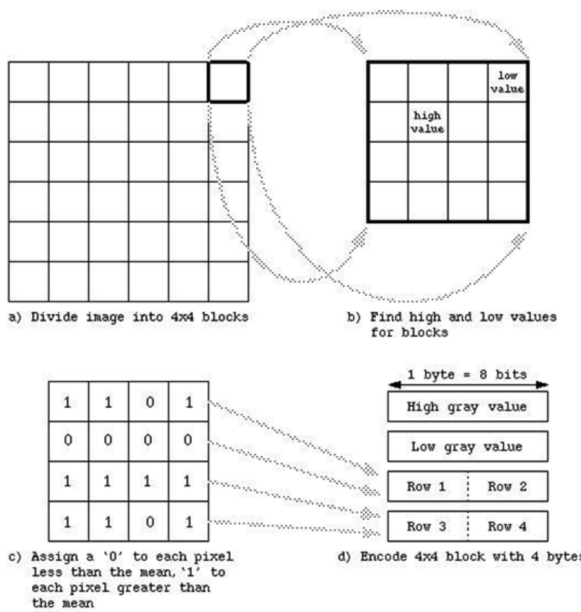


Fig. 3 Basic Block Truncation Coding

4x4 blocks of pixels gives the compression ratio 4:1, larger blocks gives greater ration for compression. Quality is reduced with the increase in block size.

D. The SPIHT Technique

Set Partitioning in Hierarchical Trees (SPIHT) belongs to the more sophisticated next generation of wavelet encoders. Properties of the wavelet-transformation are used to increase the efficiency of SPIHT. The SPIHT advantage is even more pronounced in encoding color images, because the bits are allocated automatically for local optimality among the color components, unlike other algorithms that encode the color components separately based on global statistics of the individual components. You will be amazed to see that visually lossless color compression is obtained with some images at compression ratios from 100-200:1[6]. SPIHT coding is important to have the encoder and decoder test sets for significances, so the coding algorithm uses three lists called List of Significant Pixels (LSP), List of Insignificant Pixels (LIP), and List of Insignificant Sets (LIS). These are coefficient location lists that contain their coordinates. After the initialization, the algorithm takes two stages for each level of threshold-the sorting pass (in which lists are organized) and the refinement pass (which does the actual progressive coding transmission). The result is in the form of bit stream. The LIP contains coordinates of coefficients that were insignificant in the previous sorting pass. In the current pass they were tested, and those that test significant are moved to the LSP. In a similar way, sets in the LIS are tested in sequential order, and when a set is found to be significant, it is removed from LIS and is partitioned. The new subsets with more than one coefficient

Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

are placed back in the LIS, to be tested latter, and the subsets with one element are tested and appended to the LIP or the LSP, depending on the result of the test, the refinement pass transmits the nth most significant bit of the entries in the LSP[7]. Fig. 4 shows how a spatial orientation tree is defined in a pyramid constructed with recursive six iterations splitting. The coefficients are ordered in hierarchies. According to this relationship, the SPIHT algorithm saves many bits that specify insignificant coefficients[8].

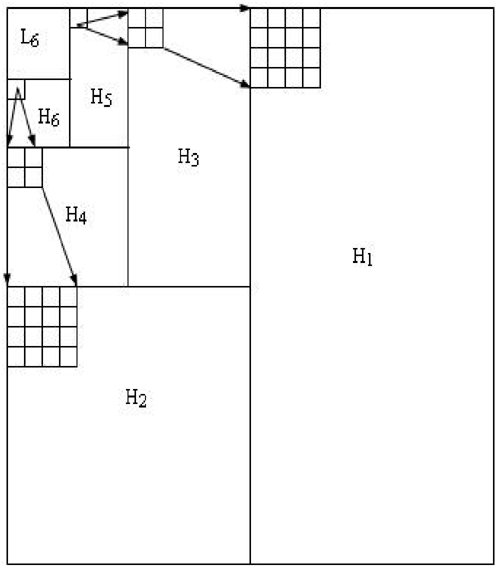


Fig. 4 Spatial orientation tree in SPIHT

III. GPU ACCELERATED COMPUTING

GPU-accelerated computing is the use of a graphics processing unit (GPU) together with a CPU to accelerate deep learning, analytics, and engineering applications. Pioneered in 2007 by NVIDIA, GPU accelerators now power energy-efficient data centres in government labs, universities, enterprises, and small-and-medium businesses around the world. They play a huge role in accelerating applications in platforms ranging from artificial intelligence to cars, drones, and robots [10]. The advent of modern GPUs for general purpose computing has provided a platform to write applications that can run on hundreds of small cores. CUDA (Compute Unified Device Architecture) is NVIDIA's implementation of this parallel architecture on their GPUs and provides APIs for programmers to develop parallel applications using the C programming language. CUDA provides a software abstraction for the hardware called blocks which are a group of threads that can share memory. These blocks are then assigned to the many scalar processors that are available with the hardware. Eight scalar processors make up a multiprocessor and different models of GPUs containdifferent multiprocessor counts[9].

NVIDIA GPUs (Graphics Processing Units) power millions of desktops, notebooks, workstations and supercomputers around the world, accelerating computationally-intensive tasks for consumers, professionals, scientists, and researchers[12]. A GPU is a processor designed to accelerate the computation of graphics operations. The term GPU is often used in contrast or comparison with central processing unit (CPU), GPUs are placed on graphics boards where they are used to speed up 3D graphics rasterization, the task of taking an image described as a series of shapes and converting it into a raster image for output on a video display. This process can be controlled using small programs called shaders. The shader instruction set has evolved over the years to the point that it is now possible to use GPUs for general purpose computations. GPU computing started as an effort by the scientific community to exploit the raw processing power of GPUs to make intensive computations. In fact, the power of the most recent GPUs is comparable to the computational power of a cluster with hundreds of CPU cores [11]. The purpose of using GPU for computation is to reduce or alter the execution time and to manage huge data required to process images. The GPU with CUDA (Compute Unified Device Architecture) parallel computing architecture will provide compelling benefits for data mining applications. In addition, its superior floating-point computation capability and low cost will definitely appeal to medium-sized business and individuals[13].

www.ijraset.com
IC Value: 45.98
Volume 5 Issue VI, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

IV. EXPERIMENTAL WORK

Different algorithm for CPU and GPU are designed and tested on two different CPUs and GPUs. The CPUs used for processing are Intel Core i5-5200U Processors made for mobile devices running at maximum frequency of 2.70GHz having 2 cores, 4 threads and 25.6 GB/s memory bandwidth [14]. Intel Core i3-6100 Processors made for desktop computing devices running at maximum frequency of 3.70GHz having 2 cores, 4 threads and 34.1 GB/s memory bandwidth [15]. The GPUs are NVIDIA GeForce GTX 920M, GTX 920M is based on 28nm Kepler GK208 architecture and operating at 575MHz; It features 384 CUDA cores, 2,048 MB DDR3 memory, which are connected using a 64-bit memory interface and memory bandwidth is 14.4GB/s [16]. The second GPU is NVIDIA GeForce GTX 750Ti, 750Ti is a gaming performance graphics card, built on Maxwell architecture, and based on the 700 series graphics processor. It features 640 CUDA cores, NVIDIA has placed 2,048 MB GDDR5 memory on the card, which are connected using a 128-bit memory interface. The GPU is operating at a frequency of 1020MHz and can be boost up to 1085MHz of clock frequncy. Memory bandwidth of GTX 750Ti is 86.4 GB/s both the GPUs are having different architectures and are made for very different application, GPU GTX 920M is made for portable computing devices and GTX 750Ti is intended for gaming desktops and personal computers[17]. The GPU algorithm requires sending data form CPU to GPU before the application compression/decompression algorithm to the input data; GPU uses its massively parallel cores to process the algorithm.

V. RESULTS AND DISCUSSION

Gray level and color images are considered for the proposed algorithms and the performance of different algorithms is shown in Table 1, compression ratio (CR), data redundancy (RD), MSE and PSNR value are compared for gray and color images. All given results are generated for variety of images having different dimensions and size.

TABLE I

SUMMARY OF RESULTS

Compression Techniques	C_R	R_D	MSE	PSNR (dB)
Huffman Coding Gray Level Color Images	3.45 1.31	0.71 0.23	0 0	-
Predictive Coding Gray Level Color Images	3.63 3.66	0.724 0.726	0 0	-
BTC Gray Level Color Images	22.24 8.02	0.95 0.87	36.2 111.6	32.54 27.65
SPIHT Gray Level Color Images	5.56 5.55	0.82 0.82	14.15 33.81	36.62 33.81

Table 1 shows the summary of results, Huffman codes and predictive codes are belongs to lossless image compression techniques and contributes less CR where as BTC belongs to lossy compression and thus having high CR. Image compressed using lossless compression schemes can be reconstructed as replica of original one and thus having zero mean square error. SPIHT produces high PSNR, High PSNR means good image quality and less ERROR introduced to the image.

VI. CONCLUSION

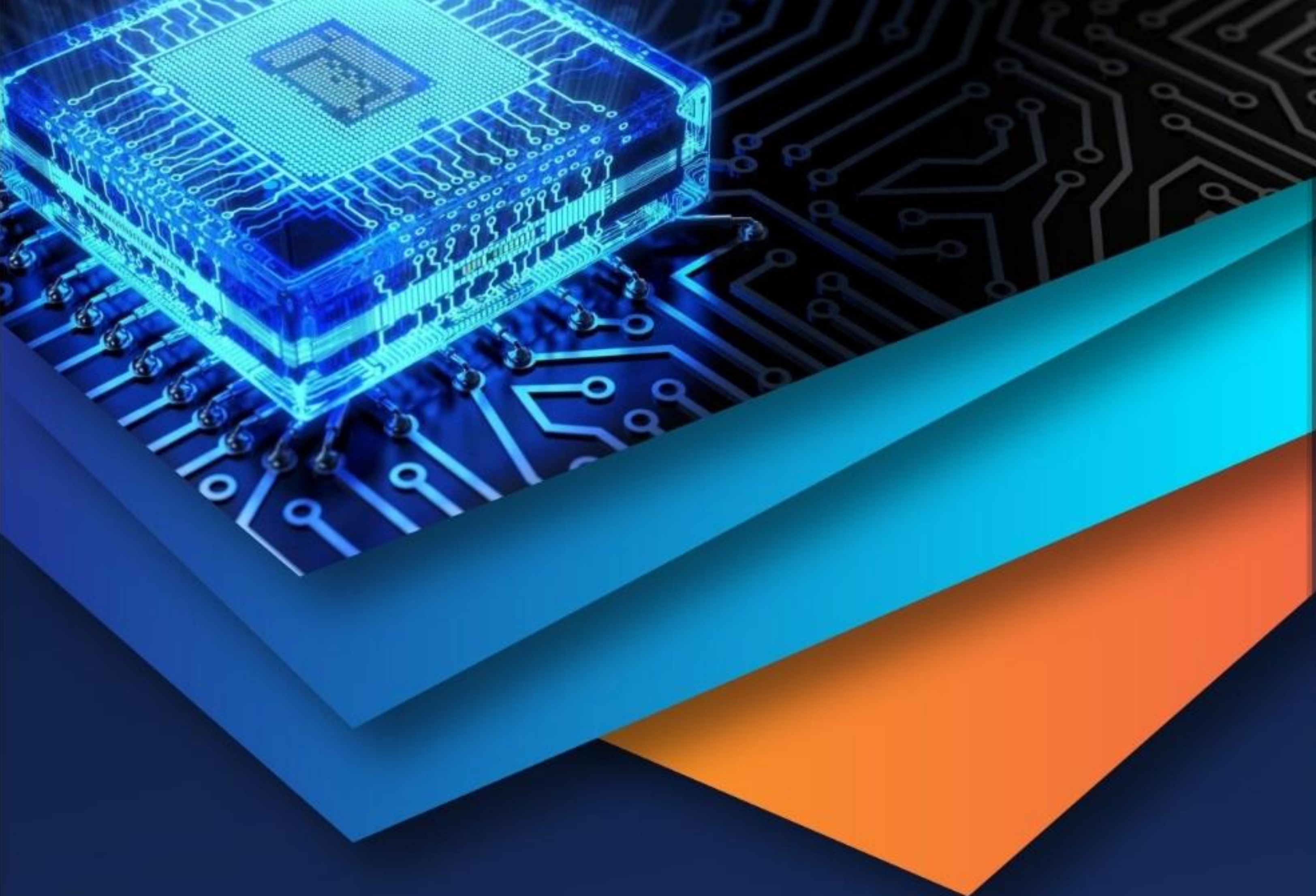
In this paper, the various techniques of image compression are compared for their compression ration, redundancy, MSE and PSNR. Sometimes various compression techniques are combined to achieve the largest possible data reduction. The SPIHT technique achieves compression which is nearly lossless because the wavelet transform filters have non integer tap weights which are truncated to finite precision. For perfectly reversible compression one must use an integer multi-resolution transform. It is observed that there is a reduction of MSE or increase PSNR in respect to the different images. The designed algorithm is tested on two different GPUs, NVIDIA GeForce GTX 920M and NVIDIA GeForce GTX 750Ti both are having different compute capability, the compute capability of GTX 920M is 3.5 and 5 is for GTX 750Ti and both the GPUs accelerates the designed algorithm up to 2.5 to 3 times faster than host CPUs. For future work the high performance computing indented GPUs like NVIDIA Tesla or Quadro products can be used to achieve maximum computing acceleration.

VII. ACKNOWLEDGMENT

The author express his gratitude towards the Dr. S. J. Sharma, Head, Department of Electronics, R.T.M. Nagpur University Nagpur, M.S., India for providing research facilities and encouragements. The author wish to record thanks for Mr. Sanjiv U. Dubey, VNIT, Nagpur and staff members of Dr. Annasaheb. G. D. Bendale Mahila Mahavidyalya, Jalgaon (M.S.), India.

REFERENCES

[1] J. Singh, "Image Compression - An Overview," International Journal Of Engineering And Computer Science, Vol. 5, Issue 8, pp. 17535-17539, 2016.
[2] M. Rabbani and P. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, Aug. 1991
[3] A. Pimpalshende, "SPIHT Algorithm Based Color Image Compression," International Journal Of Engineering And Computer Science," Vol. 3, Issue 6, pp. 6538-6543, June 2014
[4] B. Pancholi, R. Shah, and M. Modi, "Tutorial review on existing image compression techniques," International Journal Of Engineering And Computer Science, Vol. 3, Issue 8, pp. 7882-7889, Aug. 2014
[5] P. Fränti, O. Nevalainen, and T. Kaukoranta, "Compression of Digital Images by Block Truncation Coding: A Survey," The Computer Journal, Vol. 37, Issue 4, pp. 308-332. Jan. 1994
[6] Wavelet Image & Video Compression with Scalable Acquisition & Video Encoding using patented SPIHT Wavelet algorithms!.[Online]. Available: http://www.siliconimaging.com/SPIHT.htm
[7] V.Priya, B.Ananthi, "Performance Analysis of Image Compression Using DWT And WPT," International Journal Of Engineering And Computer Science, Vol. 3, Issue 10, pp.8932-8936, Oct. 2014
[8] M. Beladgham, A. Bessaid, A. Taleb-Ahmed, and I. Boucli Hacene, "Medical Image Compression Using Quincunx Wavelets and SPIHT Coding," Journal of Electrical Engineering & Technology, Vol. 7, Issue 2, pp. 264-272, March 2012
[9] P. Patel, J. Wong, M. Tatikonda, and J. Marczewski, "JPEG Compression Algorithm Using CUDA," Department of Computer Engineering University of Toronto, Course Project for ECE 1724, 2009
[10] Accelarated Computing, [Online]. Available:http://www.nvidia.com/object/what-is-gpu-computing.html#sthash.TCZF2s4R.dpuf
[11] CUDA GPUs, NVIDIA accelarated Computing.[Online]. Available: https://developer.nvidia.com/ cuda-gpus
[12] CUDA GPUs, NVIDIA accelarated Computing.[Online]. Available: https://developer.nvidia.com/ cuda-gpus
[13] V. Dhamdhere and R. Ghudji, "K-Means on GPU - A Review````," International Journal Of Engineering And Computer Science, Volume 3(5), pp. 5600-5605, May 2014.
[14] Intel Core i5-5200U Processor. [Online]. Available: http://ark.intel.com/ products /85212 /Intel-Core-i5-5200U-Processor-3M-Cache-up-to-2_70-GHz
[15] Intel® Core™ i3-6100 Processor. [Online]. Available: https://ark.intel.com/products/90729/Intel-Core-i3-6100-Processor-3M-Cache-3_70-GHz
[16] NVIDIA GeForce 920A, Tech Power Up.[Online]. Available: https://www.techpowerup.com/gpudb/2647/geforce-920a
[17] GeFocre GTX 750Ti.[Online]. Available: http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-750-ti/specifications
[18] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd ed., Prentice Hall, 2002,



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY