Fast and Efficient Image Compression based on Parallel Computing using MatLab

Prakash
M. Tech Student
Department of Electronics & Communication Engineering
MSEC

Venkateshappa
Associate Professor
Department of Electronics & Communication Engineering
MSEC

Abstract

Image compression technique is used in many applications for example, satellite imaging, medical imaging, video where the size of the image requires more space to store, in such application image compression effectively can be used. There are two types in image compression techniques Lossy and Lossless compression. Both these techniques are used for compression of images, but these techniques are not fast. The image compression techniques both lossy and lossless image compression techniques are not fast, they take more time for compression and decompression. For fast and efficient image compression a parallel computing technique is used in matlab. Matlab is used in this project for parallel computing of images. In this paper we will discuss Regular image compression technique, three alternatives of parallel computing using matlab, comparison of image compression with and without parallel computing.

Keywords: Lossy and Lossless compression, DWT, IDWT, Huffman coding, parallel computing, PCT

I. INTRODUCTION

Image compression technique is used in many applications for example, satellite imaging, medical imaging, video where the size of the image requires more space to store, in such application image compression effectively can be used. There are two types in image compression techniques Lossy and Lossless compression.

- Lossy image compression
- Lossless image compression
  1) Lossy image compression
  In lossy data compression original data is not exactly restored after decompression and accuracy of re-construction is traded with efficiency of compression. Lossy data compression algorithms are transform coding (for example, discrete cosine transform), Karhunen-Loeve Transform (KLT) and wavelet based coding(for example, continuous wavelet transform-CWT and Discrete wavelet transform-DWT).
  2) Lossless Image Compression
  As the name implies, lossless image compression schemes exploit redundancies without incurring any loss of data. Thus, the data stream prior to encoding and after decoding is exactly the same and no distortion in the reconstruction quality is observed. Lossless image compression is therefore exactly reversible.

II. BLOCK DIAGRAM OF IMAGE COMPRESSION SYSTEM

The figure 1 describes the image compression system; this block diagram explains the Encoding and Decoding of image compression technique with Huffman coding technique.

Fig. 1: Block diagram of Image compression system
A. Discrete Wavelet Transform (DWT)
The DWT can be taken as the multi resolution decomposition of a sequence. It takes a sequence \( z(n) \) of length \( N \) and produces a yield of length \( N \). The output can be seen and analyzed as the multi resolution representation of the sequence \( z(n) \), and has \( N/2 \) qualities at the highest resolution and \( N/4 \) values at the next resolution and etc. That is the frequency resolution is low at the higher frequencies and high at the lower frequencies, while the time resolution is high at the higher frequencies and low at the lower frequencies.

It principally comprises of multiplying the input sequence by translates and dilates of the wavelet DWT (Discrete wavelet Transform) module comprising of a HPF (High pass filter) and LPF (Low pass filter) followed by down sampling unit to compute the approximation and detailed coefficients of the input. The input sequence \( X(n) \) is utilizing high pass and low pass filters to produce the output \( Y_d(n) \) and \( Y_a(n) \) representing the approximation and detail samples of the input signal \( X(n) \).

B. Inverse Discrete Wavelet Transform (IDWT)
The IDWT (Inverse Discrete Wavelet Transform) can be dissected as the multi resolution decomposition of a sequence. It takes a \( Y_a(n), Y_b(n) \) as input and up sample these input, and then convolute them with HPF and LPF to produce the output. The output can be seen as the multi resolution representation of \( X(n) \).

C. Quantization Encoding
Assume \( I \) is an \( M \times N \) gray image, its pixels can be described as follows.

\[
I = \{x_{ij} | 1 \leq i \leq M, 1 \leq j \leq N, x_{ij} \in \{0, 1, 2 \ldots 255\} \}
\]  
(1)

After performing DWT on \( I \), we obtain four sub-bands LL, HL, LH, and HH of size \( M/2 \times N/2 \). Conventionally, JPEG2000 performs the uniform quantization on the resulting DWT. Coefficients sub-bands. For most images, after subtracting the average of maximum and minimum, the distribution of coefficients is similar to a zero-mean Laplacian and hence uniform quantization adopts value 0 as the center of quantization [8].

D. Quantization Decoding
The de quantization formula is shown below.

\[
R_q(i, j) = \begin{cases} 
(Q(i, j) - r)\Delta b + \omega, & Q(i, j) > 0 \\
(Q(i, j) - r)\Delta b + \omega, & Q(i, j) < 0 \\
0, & \text{others}
\end{cases}
\]

Where \( R_q(i, j) \) and \( Q(i, j) \) stand for the reconstructed coefficient and the quantized value \( 0 \leq r \leq 1 \) is an optional parameter for controlling the recovering position within quantization interval. Finally, perform the IDWT (Inverse Discrete Wavelet Transform) on the recovered coefficients in all 4 sub-bands to obtain the reconstructed image \( I \) [8].

E. Huffman Coding Steps
1) First
   1) With decreasing probability sort the gray levels.
   2) Two smallest probabilities have to add.
   3) Then the new values into the list have to sort.
   4) Then repeat 1st and 3rd step until only two probabilities remains.
2) Second
   1) For the highest probability give the code 0, and code 1 for the lowest probability in the summed pair.
   2) Then have go backwards through the tree one node and repeat from 1 until all gray levels have a unique code.

III. Parallel Computing Using Matlab(PCT)
Parallel computing has been considered to be "the high end of computing", and has been used to model difficult scientific and engineering problems found in the real world. Parallel computing can be used in image compression for fast result of compression and decompression process. In this project Parallel computing Toolbox (PCT) is used for parallel computing in Matlab. There are three alternatives are there for parallel computing in Matlab, those are bcMPI, Star-P, Parallel computing toolbox (PCT).

A. BcMPI
bcMPI is an open source software library that is an alternative to MatlabMPI and is geared towards large, shared supercomputer centers. The bcMPI library was developed at the Ohio Supercomputer Center (OSC) to provide an efficient, scalable communication mechanism for parallel computing in MATLAB while maintaining compatibility with the MatlabMPI API (Hudak et al., 2007) [7].
B. Star-P
Star-P is a client-server parallel computing platform for MATLAB available from Interactive Supercomputing. Star-P supports fine grained parallel as well as embarrassingly parallel modes of operation. The biggest advantage offered by Star-P is that it eliminates the need for the developer to use explicit Message Passing Interface (MPI) message passing calls for communicating between the back-end processes. By using the “*p” construct, users can simply indicate the variables or data that are meant to be distributed over the back-end processes [7].

C. Parallel Computing Toolbox
The Parallel Computing Toolbox (PCT) along with the MATLAB Distributed Computing Server (MDCS) are commercial products offered by The MathWorks Inc. The PCT provides functions for parallel for-loop execution, creation/manipulation of distributed arrays as well as message passing functions for implementing fine grained parallel algorithms. The MATLAB Distributed Computing Server (MDCS) gives the ability to scale parallel algorithms to larger cluster sizes. The MDCS consists of the MATLAB Worker processes that run on a cluster and is responsible for parallel code execution and process control. The PCT also allows users to run up to 8 MATLAB Labs or Workers on a single machine. This enables interactive development and debugging of parallel code from the desktop. After parallel code has been developed, it can be scaled up to much larger number of Worker or Labs in conjunction with the MDCS [7].

IV. EXPERIMENTAL RESULT
In this project fast and efficient image compression technique based on parallel computing is proposed and developed using MatLab Parallel Computing Toolbox. A set of images are taken to experiment the effectiveness of the algorithm. The figure 2 shows the various images used in the experiment. The experimental result with proposed compression method has been arranged in the Table1. Table 1 show that compression and de compression without parallel computing is not fast as compared to our proposed algorithm.

![Fig. 2: Original images used in this project](image)

<table>
<thead>
<tr>
<th>Image</th>
<th>Width</th>
<th>Height</th>
<th>Compression Ratio</th>
<th>Encoding without parallel (seconds)</th>
<th>Decoding without parallel (seconds)</th>
<th>Parallel computing CR</th>
<th>Encoding With parallel computing (seconds)</th>
<th>Decoding With parallel computing (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>512</td>
<td>467</td>
<td>54.122925</td>
<td>29.487857</td>
<td>32.500330</td>
<td>55.368042</td>
<td>3.502272</td>
<td>4.958386</td>
</tr>
<tr>
<td>2</td>
<td>512</td>
<td>467</td>
<td>54.457092</td>
<td>28.641213</td>
<td>32.273664</td>
<td>55.926514</td>
<td>1.978338</td>
<td>4.829220</td>
</tr>
<tr>
<td>3</td>
<td>450</td>
<td>356</td>
<td>54.280853</td>
<td>28.640774</td>
<td>32.334073</td>
<td>59.237671</td>
<td>2.002130</td>
<td>4.448639</td>
</tr>
<tr>
<td>4</td>
<td>307</td>
<td>230</td>
<td>43.379211</td>
<td>28.769231</td>
<td>33.673322</td>
<td>46.842957</td>
<td>1.980165</td>
<td>5.909391</td>
</tr>
<tr>
<td>5</td>
<td>1419</td>
<td>1001</td>
<td>44.053650</td>
<td>28.744912</td>
<td>33.408402</td>
<td>45.030975</td>
<td>1.962430</td>
<td>6.285207</td>
</tr>
</tbody>
</table>

V. CONCLUSION
The design of fast and efficient image compression using parallel computing is presented in this work. Matlab coding is used to design the proposed architecture. Then the design is simulated on Matlab. The proposed architecture is able to perform fast
compression of images with acceptable compression ratio. Finally, the proposed system is very fast and efficient in compression of all types of images such as nature images, medical images, satellite images etc.

REFERENCES

[7] Ashok Krishnamurthy, Siddharth Samsi and Vijay Gadepally, “Parallel MatLab Techniques”, Ohio Supercomputer Center and Ohio State University U.S.A.