

Optimizing the Complexity of Matrix Multiplication Algorithm

Pawan Manoj Rathod

Atharva College of Engineering,
Mumbai, India.

Ankit Bhupendra Vartak

Atharva College of Engineering,
Mumbai, India.

Neha Kunte

Assistant Professor
Atharva College of Engineering,
Mumbai, India

Abstract— With the bird's eye view of many analyst's attentions in the last few years to know how companies are collecting and transmitting enormous amounts of information. As there are problems in transmitting a large amount of data. This is a need of an hour to overcome the problems. The data can be compressed such as word file or image file etc. to send the data efficiently. Compression should be done in such a way that there is no loss of data. Image compression helps to overcome the problem of sending large images. In this DCT plays a crucial role. In Discrete Cosine Transformation (DCT) for compression of JPEG images, we have techniques of Quantization and encoding. In this whole work, we have used Custom matrix multiplication algorithm for reducing the complexity of matrix multiplication problem. The results obtained from the experiment when comparing with Naïve matrix multiplication and Strassen's matrix multiplication shows an increase in the performance of DCT. As the performance have increased due to the less time and space complexity of Strassen's as compared to Naïve. As Custom matrix multiplication algorithm is having less time and space complexity as compared to Strassen's algorithm we are expecting to have more increase in performance. Peak signal to noise ratio (PSNR), and different Compression Ratio (CR) for different images helps to differentiate between the performances.

Keywords— *Strassen's Matrix Multiplication, Custom Matrix Multiplication, PSNR, CR, DCT.*

I. INTRODUCTION

In this world of modernization, from displaying text in a calculator to playing high graphics game we are using the matrices. As we know that, 2D coordinate (i.e. X and Y axis) of the display unit is represented in the form of a matrix. In order to change the position of any object, we need to perform 2D or 3D transformation as per the requirements. Transformation includes Translation, Scaling or Rotation of the objects which eventually leads to the multiplication of the matrices. Matrix multiplication plays an important role in Image processing as from the point of capturing image from the digital camera to developing the images matrix plays an important role.

As there are many Matrix Multiplication algorithm available to increase performance but the most efficient method is still undiscovered. Matrix multiplication must be achieved in such a way that it takes less time and space to compute the process. Time Complexity [4] of any process can be defined as the amount of time required to compute the running state process. Similarly, the space required by the running state process is called as Space Complexity [4].

Image Compression is done to reduce the size of a graphic file without degrading the quality of an image. Due to compression, the space required for storing a single file becomes less and hence many amounts of files can be stored. A large size of file require more time to transfer and due to this one may not able to make transferring of files efficiently. [1] Various methods are available today for image compression. JPEG process is one of the most widely used methods, a form of lossy compression which is based on Discrete Cosine Transform (DCT) [1] [2] which helps to separate images into part of different frequencies. Quantization is the process, where actual part of compression occurs, where the most important frequencies remain which are to be used to retrieve the image in the process of decomposition.

This helps to remove the unwanted frequencies which will help to reduce the size of the image. Also due to the reconstruction of images, it faces lots distortion which can be adjusted in the compression phases. In this, we have proposed a new Custom Matrix Multiplication algorithm which is having less time and space complexity on comparing with existing algorithms. As a result, a decrease in the complexity will lead to increase in the performance of DCT.

II. LITERATURE SURVEY

The matrix multiplication plays a vital role in many numerical algorithms, many kinds of researches have been done to make matrix multiplication algorithms efficient. The Strassen's matrix multiplication [4] is most widely algorithm use to reduce the complexity. Various works have been done in order to implement strassen's algorithm in many applications. Coppersmith-Winograd algorithm was asymptotically fastest known algorithm until 2010. Strassen-Winograd's matrix multiplication plays a vital role in scheduling memory efficiently [7]. In 1984 for the first time, DCT was used for image compression although it was proposed in the year 1974. Discrete Cosine Transform (DCT) is widely used in image compression for conversion of the signal into frequency components. As per the recent work has done it has been proven that Strassen-Winograd's [3] algorithm helps to compute large matrices efficiently. Also, parallel computations [8] helps to reduce the Time and Space Complexity of the algorithm. DCT using strassen's algorithm provide faster output rather than the naïve.

III. EXISTING SYSTEM

A. Naïve Algorithm

The mathematical definition of matrix multiplication algorithm [4] states that if $C = AB$ for $n \times m$ matrix A and $m \times p$ matrix B , then C can be stated as $n \times p$ matrix with entries

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

A simple algorithm can be constructed which loops over the indices i from 1 through n , j from 1 to p

- Input: matrices A and B
- Let C be a new matrix of the appropriate size
- For i from 1 to n ;
 - For j from 1 to p ;
 - Let $sum = 0$
 - For k from 1 to m ;
 - Set $sum \leftarrow sum + A_{ik} \times B_{kj}$
 - Set $C_{ij} \leftarrow sum$
- Return C

The above algorithm takes time $\Theta(nmp)$ (in asymptotic notation). Due to the purpose of algorithm analysis common simplification is to assume that the inputs are all square matrices of size $n \times n$, then the running time is $\Theta(n^3)$.

B. Strassen's Algorithm

Volker Strassen published this algorithm in 1969 and proved that the Naïve algorithm wasn't optimal. The Strassen algorithm [4] is not very optimal as compared to Naïve, but it helped in research of more algorithms which led to faster approaches [5][6].

Consider A, B be two square matrices and we want to calculate the matrix product C as

$$C = AB \quad A, B, C \in \mathbb{R}^{2^n \times 2^n}$$

We partition A, B and C into equally sized block matrices as

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}, C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

with

$$A_{ij}, B_{ij}, C_{ij} \in \mathbb{R}^{2^{n-1} \times 2^{n-1}}$$

then

$$C_{1,1} = A_{1,1}B_{1,1} + A_{1,2}B_{2,1} \quad (1)$$

$$C_{1,2} = A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \quad (2)$$

$$C_{2,1} = A_{2,1}B_{1,1} + A_{2,2}B_{2,1} \quad (3)$$

$$C_{2,2} = A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \quad (4)$$

Strassen was able to reduce the number of multiplications. The important part of is performing suitable computations which are as follows

$$M_1 = (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2}) \quad (1)$$

$$M_2 = (A_{2,1} + A_{2,2})B_{1,1} \quad (2)$$

$$M_3 = A_{1,1}(B_{1,2} - B_{2,2}) \quad (3)$$

$$M_4 = A_{2,2}(B_{2,1} - B_{1,1}) \quad (4)$$

$$M_5 = (A_{1,1} + A_{1,2})B_{2,2} \quad (5)$$

$$M_6 = (A_{2,1} - A_{1,1})(B_{1,1} + B_{1,2}) \quad (6)$$

$$M_7 = (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2}) \quad (7)$$

by using only 7 multiplications strassen we can compute the new matrices. Now we can express the C_{ij} in terms of M_k as

$$C_{1,1} = M_1 + M_4 - M_5 + M_7$$

$$C_{1,2} = M_3 + M_5$$

$$C_{2,1} = M_2 + M_4$$

$$C_{2,2} = M_1 - M_2 + M_3 + M_6$$

The standard matrix multiplication takes approximately $2N^3$ (where $N = 2^n$) arithmetic operations (additions and multiplications); the asymptotic complexity is $\Theta(N^3)$. Strassen's algorithm takes approximately $\Theta(N^{2.8074})$. Strassen algorithm requires more memory compared to naïve algorithm.

IV. PROPOSED SYSTEM

A. Custom Algorithm

In this, we have tried to reduce the number of multiplications by using various loop optimization techniques. This system mainly focuses on reducing the time and space complexity without affecting the output of the system. Custom Algorithm focuses on the values of the inputs in the matrices rather than the size of the matrix which helps to reduce the multiplication operations. In this algorithm, all the multiplications are been replaced using suitable loop optimization techniques. Custom Algorithm with DCT compression can provide effective results which help to decrease the time required. As arithmetic operators have their own complexity so using addition instead of multiplication helps to reduce the complexity. As computations are done according to input values the input values must be an integer.

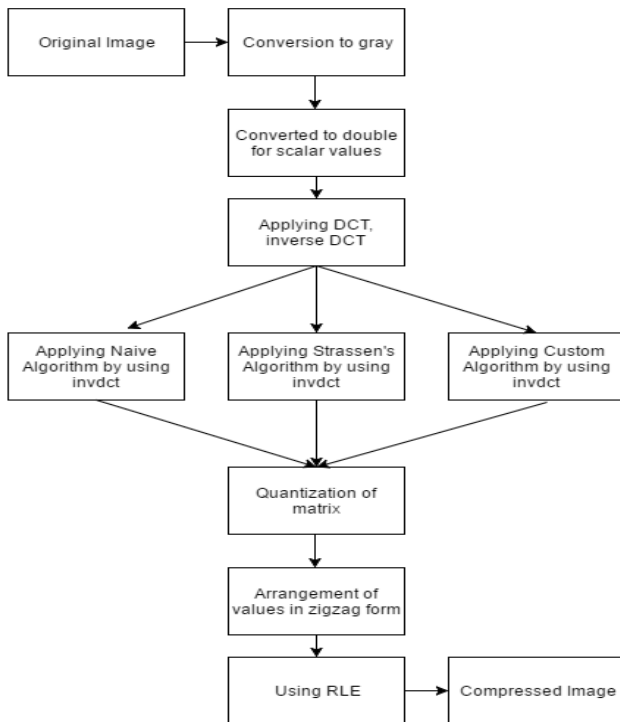


Fig 1. Block diagram of the proposed system.

CONCLUSION

Matrix multiplication is been used on a large scale. An increase in the efficiency for computing the result may lead to a vast increase in performance in the field of film industry, astronomy, computer graphics and much more. In order to achieve great efficiency, many types of research have been done till now. The proposed system would be able to achieve the desired efficiency.

FUTURE SCOPE

This paper illustrates the comparative study of various matrix multiplication algorithm using DCT compression. This paper helps to choose efficient algorithm as per the desired output condition required. Also, it is been expected to achieve matrix multiplication of fraction values which is difficult to evaluate in this model.

REFERENCES

- [1] Manoria, Manish, and Priyanka Dixit. "An Efficient DCT Compression Technique using strassen's Matrix Multiplication Algorithm." International Journal of Computer Applications 60.9 (2012).
- [2] Thota, Nageswara Rao, and Srinivasa Kumar Devireddy. "Image compression using discrete cosine transform." Georgian Electronic Scientific Journal: Computer Science and Telecommunications 17.3 (2008): 35-43.
- [3] Huss-Lederman, S., Jacobson, E. M., Johnson, J. R., Tsao, A., & Turnbull, T. (1996). Implementation of Strassen's algorithm for matrix multiplication. In Supercomputing, 1996. Proceedings of the 1996 ACM/IEEE Conference on (pp. 32-32). IEEE.
- [4] Cormen, Thomas H. Introduction to algorithms. MIT press, 2009.
- [5] Hedtke, Ivo. "Strassen's matrix multiplication algorithm for matrices of arbitrary order." arXiv preprint arXiv:1007.2117 (2010).
- [6] Oh, Seunghyun, and Byung-Ro Moon. "Automatic reproduction of a genius algorithm: Strassen's algorithm revisited by genetic search." IEEE Transactions on Evolutionary Computation 14.2 (2010): 246-251.
- [7] Boyer, Brice, et al. "Memory efficient scheduling of Strassen-Winograd's matrix multiplication algorithm." Proceedings of the 2009 international symposium on Symbolic and algebraic computation. ACM, 2009.
- [8] Li, Junjie, Sanjay Ranka, and Sartaj Sahni. "Strassen's matrix multiplication on GPUs." Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on. IEEE, 2011.