CONTACT NAME
Milind Kulkarni: milind.kulkarni@vit.edu



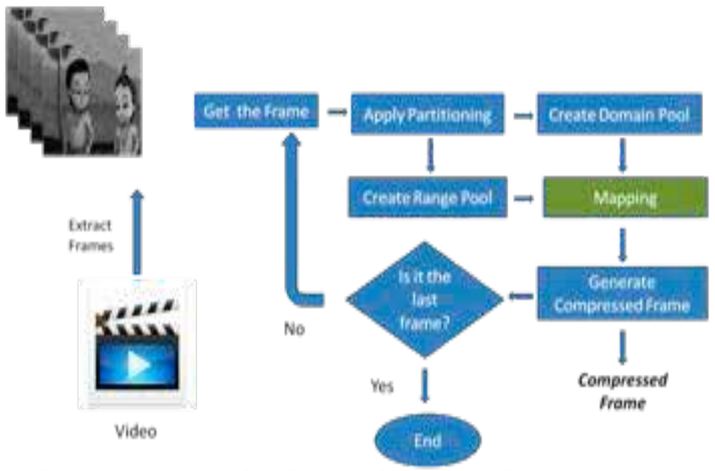
# Parallel Computing using CUDA-GPU in Fractal Video Coding

Mr. Milind V. Kulkarni, Dr. Dinesh B. Kulkarni

## Introduction

Digital videos find applications in many fields ranging from telecommunications to broadcasting and so on. Hence, in past two to three decades, there has been tremendous progress in video compression technologies. One of the lossy compression techniques is fractals. Despite its various advantages such as resolution independence and fast decompression, fractal compression methods cannot compete with the existing techniques due to high complexity of the encoder. In this paper an inter frame video compression technique, which is a combination of block matching and fractal compression, has been proposed.

**Video Encoding**




In order to overcome the drawback of fractal compression methods, i.e. high computational time, a parallel approach using CUDA enabled GPU has been proposed. The proposed approach significantly reduces the encoding time of video and provides very high compression ratio with reasonable quality of decoded video.

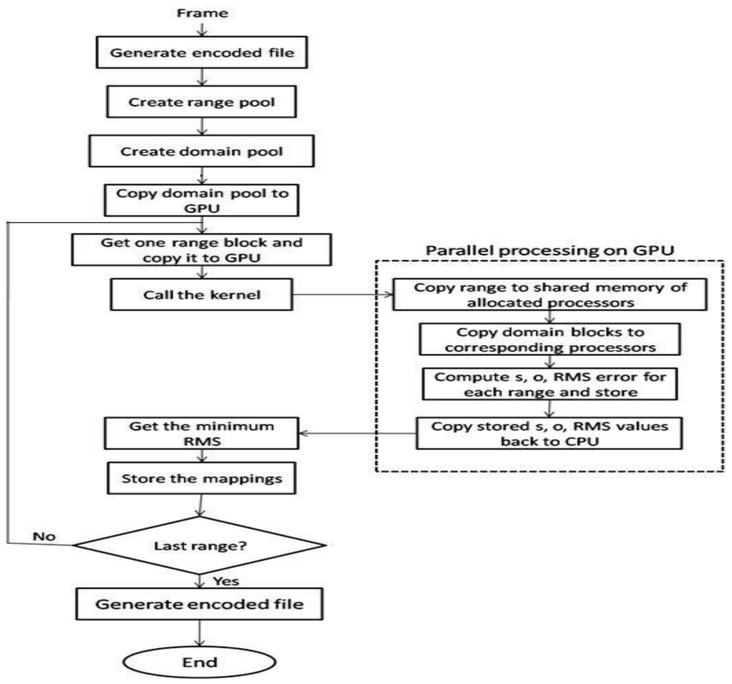
An inter frame video coding method is proposed here..

The first frame is encoded independently with quad-tree partitioning image encoding method described in [1]. That is, the image is iteratively partitioned into four sub-squares until a suitable mapping is found or predefined maximum depth of partitioning is reached. For encoding the next frame, the previous frame and current frame (frame to be encoded) are subdivided into fixed size blocks. Each block in the current frame is compared with the corresponding block in the previous frame. If the block difference is less than the predefined threshold then, the blocks are considered as matched and the location of that block is stored in the encoded file else the quad-tree partitioning is applied on the block for encoding.

Using this approach, significant reduction in time for encoding is achieved. Results are shown in figure 1. and figure 2.

## Parallel Algorithm

**Encoding of a frame using Parallel Approach**




The proposed algorithm was implemented on CPU and GPU. For CPU the mapping of range and domain was implemented sequentially and for GPU the proposed parallel approach was implemented. On CPU the proposed method was implemented using JAVA whereas for parallel approach with GPU, CUDA C was used. Three test videos with different formats (.avi, .mov, .wmv) were tested on CPU and GPU. For each video format, 80 frames of size 256 x 256 were encoded and decoded using both the sequential and parallel approach.

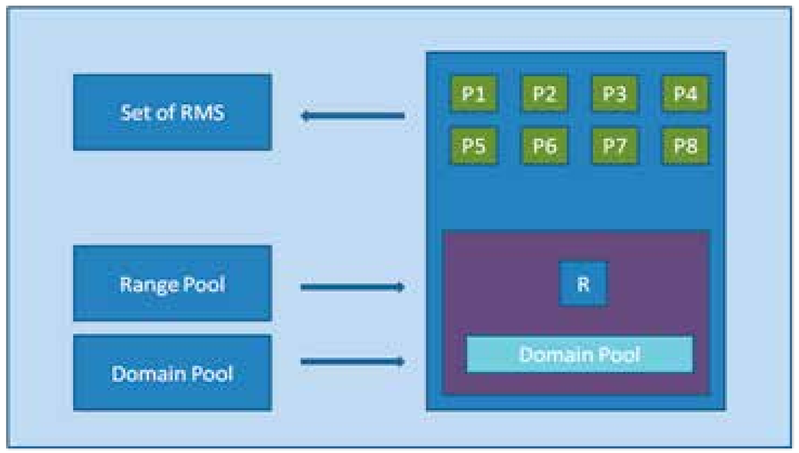
The parallel approach is an endeavor towards reducing the encoding time of sequential approach. Since the range domain mapping is the most time consuming step of the encoding phase in fractal compression, and search of a suitable domain is for each range is independent of each other, this operation can be carried out in parallel. In the proposed approach the domain pool is copied to the GPU memory and ranges are copied one after another.

## References

1. Yuval Fisher, “Fractal Image Compression: Theory and Application”, Springer-Verlag publication, 1995.

2. August Sodora, Patrick Carter. “Adapting Fractal Image Coding Techniques to a Parallel GPU Environment”, 2008

## Performance Results




The operations of creating range pool and domain pool are carried out on the CPU. For parallel processing on the GPU, the whole domain pool is copied to the GPU global memory. Also, one range block is copied to the GPU global memory. The number of processors allocated for parallel computing is equal to the number of domain blocks in the domain pool and the number of threads in each processor is equal to the number of pixels in the domain block or range block. For example, if there are 8 domain blocks of size 2 x 2 then, 8 processors will be allocated each having 2 x2 =4 threads. Therefore, total number of threads in this case would be 8x2x2=32.

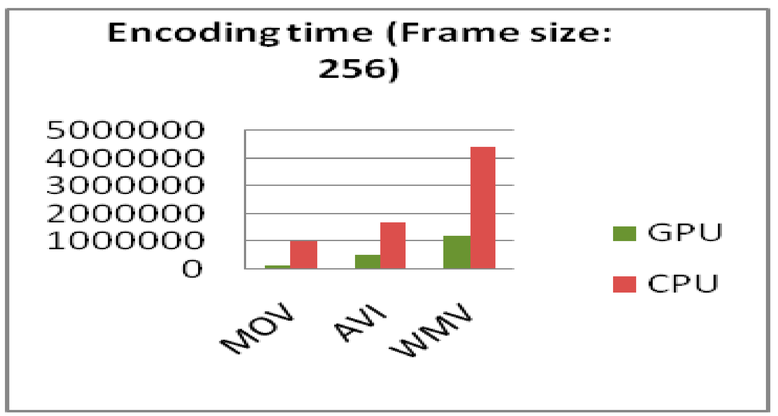



**Figure 1.**

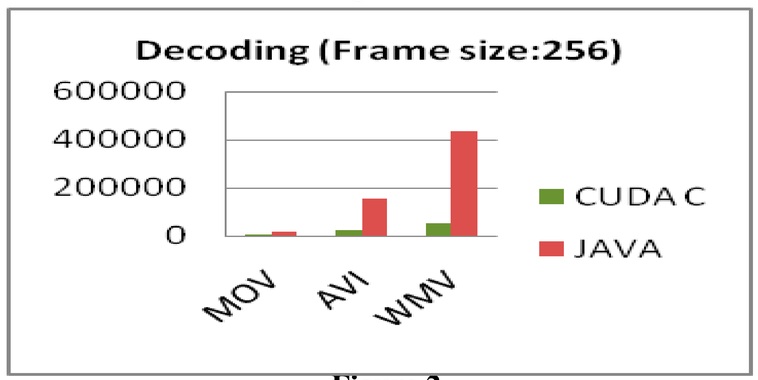



**Figure 2.**