Fractal Iterated Function System (IFS) Compression for Image and Video

Scott Mionis[1] **Vrishab Commmuri[2]** **Bridget Tan[2]** **Weishan Wang[2]**
Department of Computer Science[1] Department of Electrical and Computer Engineering[2]
Carnegie Mellon University
Pittsburgh, PA 15213
{smionis, vcommuri, bridgett, weishanw}@andrew.cmu.edu

Abstract

Fractal Image compression, a technique pioneered in the late 80's and 90's, attempts to compress 2D signals by downampling and subsequently learning an upsampling function that can be applied iteratively to reconstruct them from a source signal. After each downsampling stage, an interpolator function f is learned for each image patch and attempts to capture the lost information with 'fractal coefficients', or rather, parameterizations of the interpolator for each patch. Initially discarded in favor of more efficient and effective compression algorithms, such a technique warrants revisiting with the rise of application domains like satellite imagery, computer vision, and video classification. This semester we have implemented baseline and parallel versions of existing algorithms, and have shown comparable results to those in recent publications. For further work, we plan to improve this technique and apply it to a number of modern usage areas across computer vision, image processing, and signal compression.

1 Introduction

With the proliferation of 1D, 2D, and 3D signal information across most computer systems today, there is a distinct need for intelligent compression algorithms to reduce the size of data. While traditional approaches like JPEG(DCT)[2] and JPEG2000(DWT)[1] exist, these techniques are traditionally symmetric with respect to compression and decompression time. For non-real-time applications there is a constant need for better compression ratios, and in this domain, long compression times are preferred to larger file sizes.

Additionally, in the fields of image processing and computer vision, feature selection is strongly informed by what transform bases are available. Fourier analysis is universally useful, but non-optimal for certain classification tasks. Specifically with geometric or modulated signal data, we believe that transforming data into new domains such as the "fractal" domain could be illuminating for certain classification tasks.

2 Related Work

Fractal Image Compression (FIC) was first proposed in 1988 by Michael F. Barnsley[7]. The algorithm is implemented as an iterated function system (IFS); images are divided into patches, downsampled, and then coefficients for upsampling the patch are found to minimize error. This technique was explored theoretically, but the first fully automated algorithm for fractal image compression was not proposed until 1992 by Jacquin[8] using the "Partial Iterative Function System," or PIFS[6]. This algorithm certainly improved the viability of FIC, but several problems remained, foremost being the disproportionately large compression time. In terms of common ground for creating new algorithms, most modern algorithms are branches of of the original baseline, which is best described in the reference text by Yuval Fischer[11].

3 Datasets

We are using the University of Waterloo's Greyscale Set 1 dataset[12]. This dataset contains 12 PGM images comprising of photographic pictures, synthetic images, and a montage of smaller photos. It also contains standard image processing images like Lena. Each of these is a 256 x 256 grayscale image. We have resized the images in this dataset to 128 x 128 in order to keep compression times low for running large series of test and evaluation cycles, and to maintain cross-compatibility with other datasets.

4 Working Experiment

We implemented the baseline algorithm described by Fischer[11], which we have discovered from our literature review to be the most recent common-ground for many advanced algorithms (github link: $https://github.com/Scottamattic/18797\_Fractal\_Image\_Compression$). We also implemented a second baseline detailed by Texas Instruments[13]. In both cases, our results matched those in the published works. The Fischer[11] algorithm yielded the best results (as determined by PSNR), so we selected it as our baseline.

4.1 Selected Results

We ran our baseline on the 128 x 128 image of Lena found in the Waterloo dataset. Starting from an all-0 initialized image of size 128x128, we were able to reconstruct Lena accurately using the codebook generated from a single affine mapping and naive 2x2 or 4x4 block segmentation scheme. Each individual iteration is performed by downsampling the input image by 2 and applying the codebook to upsample it back to 128x128. We achieved similar results to those in the first chapter of the Yuval Fischer text.

Figure 1: Lena across codebook iterations, from all 0 input

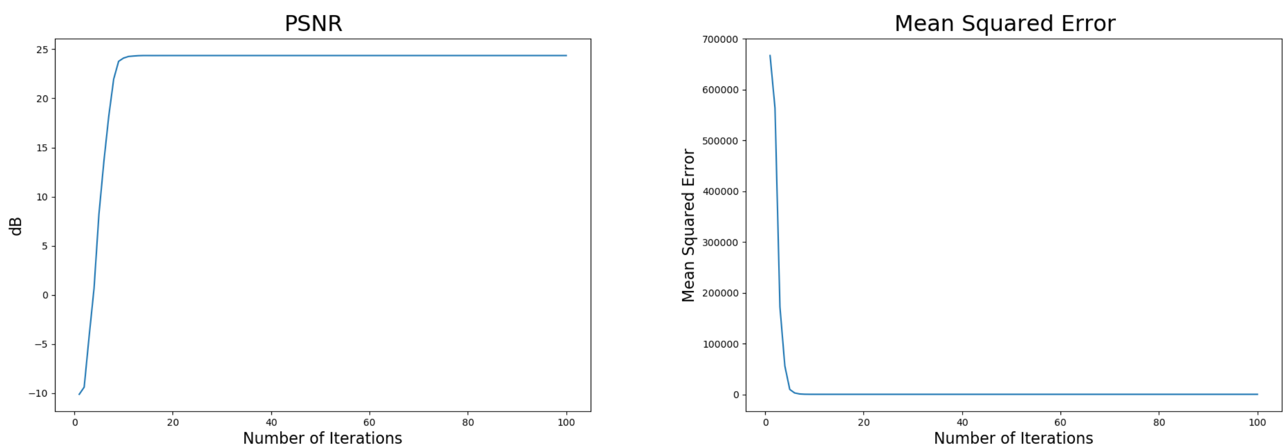

Figure 2: PSNR and MSE per decoding iteration

Evaluation of these results is described in the Evaluation Metrics section.

Additionally, we utilized a MapReduce framework to implement a parallel version of this algorithm. The domain blocks were partitioned equally among the cores (the Mapping phase), and the best matching domain block for each range block was found from the result of each Mapper (the Reduction phase). This approach enables the algorithm to achieve a speedup commensurate to the number of cores on the machine which implements it. Though we do not presently have access to a GPU, we hypothesize that the use of one would enable highly-efficient compression under this method.

5 Not-Working Experiment

The greatest impediment to the widespread usage of fractal image compression is the poor computational complexity of the encoding phase. As it is traditionally implemented, fractal encoding has a roughly $O(N^2)$ complexity – one loop over each range block, and a nested loop to find the best matching domain block – which quickly becomes intractable for many-pixeled images (i.e., > 480p).

We are currently developing a method that disregards the nested loop – achieving an $O(N)$ encoding complexity – by first computing a selection of basis vectors from the domain using the Karhunen-Loève Transform, and then employing a simple heuristic to select the appropriate basis vectors to construct each range block.

6 Evaluation metrics

To evaluate the results of our image decoder, we used Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) to quantify image quality after decompression as seen in Figure 2. Both metrics show that reconstruction of the original image improves exponentially with more decoding iterations.

We also evaluated the compression ratio achieved in comparison to JPEG, PNG and JPEG2000, but the current baseline implementation does not currently achieve any compression ratio due to shortcuts taken in software engineering. After evaluation of how we can better store parameters of our IFS, we expect to achieve a compression ratio inversely proportional to the image segmentation size.

For evaluating performance of the parallel implementation we compared both experimental runtime on standard hardware and theoretical work/span calculations.

Qualitatively, we will evaluate whether deriving fractal coefficients for a signal yields additional insight into the structure of the data, or reveals any powerful features for downstream classifiers. We also will compare our algorithm to existing audio and video encoders as we extend to those domains.

7 Timeline

We have approximately met all relevant deadlines marked by our original timeline. We completed our Texas Instruments[13] baseline early in fact, which gave us time to improve the baseline by incorporating elements from more advanced sources like the Fischer text[11], and eventually this led to a complete reimplementation. Due to implementing multiple baseline implementations (we only had room to show the best results in the 'Chosen Results' section) we have not spent as much time on developing pseudocode for our advanced implementation as is indicated by the timeline, but we believe this tradeoff led to a better understanding of the problem.

Our future work primarily constists of rigorously benchmarking our parallel algorithm, tweaking our baseline to achieve the compression ratios we expect, and implementing our advanced algorithm. For the advanced algorithm we plan to use smarter image segmentation, multiple encoding levels, and a range of scale-invariant representations (like dictionary weights) that could help us better track self-similarities across scales.

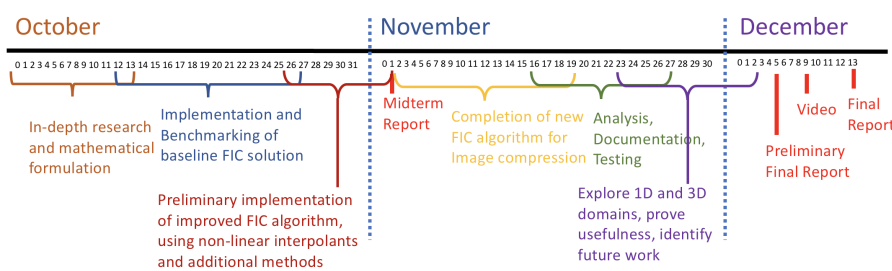

Figure 3: Original Project Timeline

8 Division of work among team members

Work among team members has been split along the lines of our original project proposal. Vrishab and Scott have focused on mathematical derivation and analysis of the algorithms in the reference texts, in order to understand their geometric consequences. Bridget and Weishan have leveraged their ML expertise to choose datasets and investigate how the image processing community shares and evaluates results. In terms of software deliverables, this means that the implementation of the compression algorithms have primarily been undertaken by Vrishab and Scott, while the pre/post-processing and backend data pipelining has been streamlined by Bridget and Weishan. We all collaborate on project discussions regarding direction and future work.

References

[1] J. Zhu, "Image compression using wavelets and JPEG2000: a tutorial," in Electronics Communication Engineering Journal, vol. 14, no. 3, pp. 112-121, June 2002, doi: 10.1049/ecej:20020303.

[2] G. K. Wallace, "The JPEG still picture compression standard," in IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992, doi: 10.1109/30.125072.

[3] Joshi M., Agarwal A.K., Gupta B. (2019) Fractal Image Compression and Its Techniques: A Review. In: Ray K., Sharma T., Rawat S., Saini R., Bandyopadhyay A. (eds) Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 742. Springer, Singapore. https://doi.org/10.1007/978-981-13-0589-4$_2$2

[4] Dan L., Peter K J., (2007) A Survey of Parallel Algorithms for Fractal Image Compression in Journal of Algorithms and Computation Technology, doi: 10.1260/174830107781389021

[5] Gauri Joshi, (2013) Fractal Image Compression and its Application in Image Processing in Internation Journal of Science and Research, issn: 2319-7064

[6] Zhou Wang and Ying Lin Yu "Partial iterated function system-based fractal image coding", Proc. SPIE 2751, Hybrid Image and Signal Processing V, (7 June 1996); https://doi.org/10.1117/12.242020

[7] M.F. Barnsley, Fractals Everywhere, New York: Acacemic, 1988.

[8]Arnaud Jacquin; "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," IEEE Transactions on Image Processing, vol. 1, no. 1, January 1992, pp. 18-30.

[9] Kominek, J. Advances in fractal compression for multimedia applications. Multimedia Systems 5, 255–270 (1997). https://doi.org/10.1007/s005300050059

[10] Jaseela C C., Ajay James, (2014) A New Approach to Fractal Image Compression Using DBSCAN in International Journal of Electrical Energy, doi: 10.12720/ijoee.2.1.18-22

[11] Y. Fisher, Fractal Image Compression: Theory and Application. New York: Springer, 1995.

[12] University of Waterloo, "Image Repository", 2009. [Online]. Available: http://links.uwaterloo.ca/Repository.html.

[13] Texas Instruments Europe. (1997). "An Introduction to Fractal Image Compression" [PDF file]. Retrieved from https://www.ti.com/lit/an/bpra065/bpra065.pdf