Fractal Compression

Related Topic Report

By

Henry Xiao

Queen's University

Kingston, Ontario, Canada

April 2004

Fractal Introduction

Fractal is first introduced in geometry field. The birth of fractal geometry is usually traced back to the IBM mathematician *Benoit B. Mandelbrot* and the 1977 publication of his book "*The Fractal Geometry of Nature*". Later, *Michael Barnsley*, a leading researcher from Georgia Tech, found a way of applying this idea to data representation and compression with the mathematics of *Iterated Functions Systems (IFS)*. Regarding the computational complexity, fractal compression algorithm based on IFS was still not practical to use at that time. And it is *Arnode Jacquin*, one of Barnsley's PHD students, who finally settled down this problem with *Partitioned Iterated Function Systems (PIFS)*, which is modified from IFS by partitioning the domain space into subspaces. Since the development of PIFS, fractal image compression has been widely studied and various schemes have been derived and implemented for experiments.

Wohlberg and de Jager conclude that the fundamental principle of fractal coding consists of the representation of an image by a contractive transform of which the fixed point is close to that image at their review [4]. The image space is accepted as a complete metric space. By the contractive mapping theorem and collage theorem, the contractive transform is always possible within certain threshold. Original approach taken IFS tries to find a number of affine mappings on the entire image, which is rather slow in terms of searching the contractive map function. Jacquin's PFIS takes different approach to find the individual mappings for subsets of the images. The partition of the image has been introduced to get those subsets. Many literatures have been written to explain the idea of partition scheme ([1], [4]). The main concept is to partition the image space twice as "range blocks" and "domain blocks". Both partitions follow the same scheme in geometric sense covering the whole image, but the latter is allowed to overlap with each other. Then each range block is mapped onto one domain block by taking certain transformations such as rotation, flip, etc. One mapping is eventually identified by two variables, named scaling and offset. The compression is achieved by mapping each range block of the image to one similar domain block. The word "similar" is kind of fuzzy here, however, most cases are measured by distance between points under *MSE (Mean-Squared Error)* as commented in [4]. This fundamental principle of fractal coding leaves considerable latitude in design of a particular implementation regarding the partition scheme, mapping scheme, etc. Review [4] lists five interesting aspects of classifying the differences of fractal compression schemes from various literatures.

Up to now, like pointed out by Fisher et al. in [1], fractal compression based on PIFS does not really lead us a new practical compression scheme which is guaranteed to be superior to others, even in image compression. Many comparisons have been done in image compression field with JPEG image encoding standard. Experiments show that at the lower compression ratio, which is below around 40:1, the JPEG still performs better than fractal encoding. In certain sense, we may not want to go higher compression ratio, even we know fractal encoding may give us better performance above 40:1, since the higher ratio means the more lost of information in either case. This is probably the primary reason that fractal encoding is still under academic research. Recent attempt to improve fractal compression is mainly focusing on applying some transform coding techniques with fractal encoding. [1] is some early works trying to combine fractal with

DCT and wavelets. [2] is relatively new result with a jointly fractal/DCT compression scheme, while many researchers are interested with joining wavelets with fractal encoding. [3] is one of the theory establishing paper to join wavelet transform with fractal encoding. The basic principle behind joining fractal with certain transform is to take the advantage of identifying more self-similarities from applying certain transform since some transforms like DCT and wavelets can identify signals by frequency or energy levels. Mathematically, DCT is also an orthogonal transform, and wavelet can always be orthogonized or biorthogonized, which themselves achieve certain compression.

One shall realize that fractal encoding is the same as *Vector Quantization (VQ)* in general. Through some careful observation, it is not hard to see the domain blocks are acted like codebook similarly to VQ. The differences are VQ codebook is explicit and the mapping from coming vector to codevector is one-to-one straight, while in fractal encoding, the domain blocks, some literature ([2], [4]) named “virtual codebook”, is inexplicit and mapping is through certain transformations. Fractal compression scheme design problems can also be formulated very similarly like VQ problems. So, people do realize the potentials of fractal encoding in compression world from this kind of relationship with VQ. Many fractal problems have been addressed or formulated taken VQ approach. And some VQ techniques have been tried with fractal coding. However, in most situations, the theory works of measuring performance with fractal coding are not developed mathematically yet. Above differences from VQ make it relatively hard to formulate the fractal coding problem to measure its performance.

Fractal Encoding

We intend to look at some details of fractal encoding concluded at the review [4] here. Two main parts are considered below, which are partition and mapping. Partition scheme is essential to fractal encoding, hence has been mostly focused among large amount of fractal literatures. Similar to VQ, partition directly decides the encoding process. Furthermore, in fractal partition, the "virtual codebook" is also decided by partition scheme. Under certain sense, partition scheme dominants the whole encoding process and decides the achievable quality of fractal compression. Mapping involves many details works such as transformation, distance calculation, etc. It shows a tradeoff between compression ratio and fidelity. Most developed fractal encodings follow the same mapping from the original idea by Jacquin. Variations occur from literature to literature, but quite small compare with partition. We do not address the decoding here since decoding is straight forward like VQ decoding. Most experiments prove that this is true in general with very efficient fractal decoding.

Review [4] categories the partition schemes into two aspects based on the schemes' geometric means. *Right-angled range partition scheme* is the group with perpendicular polygons such as fixed size square. *Triangular and polygonal range partition scheme* is an irregular group of partitions using triangles, polygons, etc. While many attentions have been drawn on the latter group, especially partition with *Delaney triangulation*, which in theory maybe give us some optimal performance in terms of identifying blocks, the first group is still widely used in most implementations. Review [4] pointed out that the disadvantage of partitions which are not right-angled is the interpolation required in performing the block transforms where there is no simple pixel-to-pixel correspondence between domain and range blocks. Among the first group, *Quadtree* is probably the most popular partition. The tree structure allows us to recursively construct our range blocks, and subdivide or merge certain blocks based on different thresholds, which is more efficient than simply fix block partition algorithmically. The performances of different partitions have not been agreed in academic research field. Different paper gives different comparison results. One may check [4] in detail for further references under this topic.

Domain pool of a fractal compression scheme is referred as virtual codebook, which is highly related with the partition scheme chosen. Some attempts have been tried on domain pool selection to improve the compression process, including applying certain VQ codebook as domain pool. Review [4] gives four different types of selections. The main idea for improvement is to provide better or smaller codebook for the mapping process later. As we can clear expect, it would be rather hard to construct or visualize a domain pool if a nonright-angled partition scheme is chosen in the first place. And because the partition scheme difference, it is impossible to compare the selection of domain pools alone. Common choice of domain pool is based on quardtree partition like in [2]. Most of papers ignore other partition possibilities, and the domain pool is found with partition process in global fashion.

Mapping in fractal encoding involves identifying the domain block for each range block. Compression ratio is highly related with this process. We have explained the difference between fractal encoding and VQ in terms of mapping at the beginning, which differs most in the way that fractal mapping is not one-to-one straight, but through transformations. Different papers allow different transformations. Most common ones in image fractal compression are rotation and flip, which are isometric symmetries. And because the size of range block is different from domain block, the latter needs to be reduced to compare with the first. It is always time-consuming task to find a mapping from a range block to a domain block, which occupies most of the encoding time. The searching is mostly done in global way. Some improvements through reduce domain pool (codebook), clustering domain blocks, using nearest neighbour, etc are briefly discussed in [4]. Most literatures do not address this problem since their choice of partition is quadtree, which has fast algorithm for searching among the tree structure, like in [2]. Still, techniques like clustering may be applied to improve the searching efficiency. But the tradeoff will be the computational cost on constructing required structure, which is also true in quadtree case comparing with simple fix block partition.

Finally, reivew [4] provides some interesting observations of fractal encoding through the link with VQ. Orthogonalization operator is introduced to domain block in order to simplify the domain pool, and make the convergence faster to achieve better compression within fixed number of iterations. Also, idea of optimal bit allocation from VQ is plugged into fractal encoding to efficient assign bits among domain blocks in the virtual codebook. The most important discovery done by different researchers individually is the possibility of improving fractal encoding through joining some transforms, which is going to be discussed at next section.

Joint Scheme of Fractal Encoding

As we have briefly mentioned at the introduction, recent attempts are trying to join fractal with some transforms, such as DCT, wavelets, etc. We mainly focus on the development of the jointly fractal/DCT scheme, which appears on [2]. However, we will still relate some topics with wavelets during our discussion.

It is well-known that fractal image compression still has to be improved to be realistic. The main focus from one perspective is to better identify the self-similarity between blocks. Researchers have tried different things as summarized in [4]. Very intuitively, transforms like DCT, and wavelets, which have been used with *JPEG* and *JPEG2000* standards, fall into our attention. The idea is well explained in [2] that transform coding methods are to decorrelate pixels in an image region by taking advantage of intra-region redundancies, and the fractal method addresses redundancy on the region-to-region basis, which is to say that for every range region, we can find a contractive transformation operating on different part of the image that results in a close approximation of the range under consideration. Simply, more self-similarity is presented through applying transforms. In terms of the compression process, we apply fractal compression on transformed coefficients from the original image. The decorrelation is done by orthogonalization, since the DCT and wavelets do reserve orthogonal property. Furthermore, wavelet transform is recognized as a more advanced transform than DCT. The advantage of applying certain wavelet transform is that we can reserve some useful properties such as symmetry, which help to identify self-similarity in fractal coding, through biorthogonal, not orthogonal wavelet transform. This observation does not appear at the wavelet section in [4], but worth to be pointed out here.

The joint DCT/fractal scheme presented in [2] is a new instance of those joint fractal schemes. The author did a thorough job deriving mathematic formulation of the joint scheme. By applying *Lagrangian* method, the joint scheme is optimized under certain criteria, which may be suboptimal in general sense. The DCT helps to get rid of redundancy within each block like the using in JPEG. Somewhat different is that they found some mechanisms like *zerorun, magnitude*, which are proved to be effective under JPEG, are not significant under fractal coding. The mathematic formulation helps to solve the conventional fractal coding inability to provide good rate control problem under high fidelity requirement. It is also showed from the formulation that allowing a greater domain pool or more isometries, after a certain point, does not lead to increased reconstruction quality. This is a little counter intuitive by the meaning that more bits per transformation may not increase the fidelity. However, we shall keep in mind that the IFS does have a "stable" stage theoretically.

DCT is really an old transform to be combined with fractal. But some great results shown in [2] again prove the potentials of those joint schemes. The most popular transform joined with fractal is still wavelet. Many improvements can be achieved from using certain wavelet transform. Wavelet transform reserves some features different from traditional transforms, such as Multiresolution Analysis (MRA). One important point is that wavelet is recognized as a subband technique, which has the ability to identify subbands from the original signal (image). It would not be surprise that this can help finding more self-similarity since each subband has smaller variance area comparing with the whole signal. Personally, I expect joint scheme with wavelet transform performing better than with DCT. However, both the formulation and computation complexities should increase a lot with wavelet transform from DCT. Generally speaking, DCT still holds the advantage of its simplicity.

Conclusion and Future Works

Review [4] gives a very intense summary of fractal related compression schemes, while paper [2] puts up a new joint DCT/fractal scheme recently. The latter has a good mathematical formulation for the optimal problem. The formulation is basically *Lagrangian multiplier method*, but the relationship established with *Directed Acyclic Graph (DAG)* is somehow a novel approach for solving the optimal problem. With the proof of fractal compression optimal problem is NP hard as mentioned in [4], [2]'s approach is a reasonable deduction from the absolute optimal, and some significant results from their implementation prove their correctness of the formulation empirically.

Review [4] also put some words about possible works to do with fractal compression, including joint schemes. The idea of joining fractal with transforms such as DCT, and wavelets is not new, which appears in some papers as early as [1] and [3]. The concept of joint compression schemes has been mostly done in this field, however, does not like paper [2], still leak of mathematical formulation in most cases. So, one possible aspect to tackle on fractal compression is to establish mathematical models of those joint schemes in order to derive optimal or suboptimal encoding. On the other hand, some detailed research aspects of fractal compression seem stopped such as partitioning, searching, etc. It is still worth to continue researching on some different approaches for those topics, since each one of them is highly related with the compression algorithm efficiency. Of course, it will be very hard to have some achievement in those already well-studied topics. Finally, even though the referenced papers are all on image fractal compression, it does not stop people's curiosity of applying fractal compression on other data, like audio, video, etc. Any extension to a new data type will be a great challenge in this field since fractal idea becomes less intuitive in other types of data. Some limited works have been done to introduce fractal as well as joint schemes to compress other types. We may expect to see more literatures applying fractal related schemes on various data types in the near future.

Reference

[1] Y. Fisher, D. N. Rogovin, and T.-P. J. Shen. *Comparison of fractal methods with discrete cosine transform (DCT) and wavelets*. Neural and Stochastic Methods in Image and Signal Processing III, S.-S. Chen, Ed., San Diego, CA, USA, July 1994, vol. 2304 of SPIE Proceedings, pp. 132-143.

[2] G. Melnikov, and A. K. Katsaggelos. *A jointly optimal Fractal/DCT compression scheme*. IEEE Transactions on Multimedia, vol. 4, no. 4, pp. 413-422, December 2002.

[3] A. van de Walle. *Merging fractal image compression and wavelet transform methods*. Proceedings of the NATO Advanced Study Institute, Trondheim, Norway, July 1995, pp. 8-17.

[4] B. Wohlberg, and G. de Jager. *A review of the fractal image coding literature*. IEEE Transaction on Image Processing, vol. 8, no. 12, pp. 1716-1729, December 1999.