THE FUTILITY OF SQUARE ISOMETRIES IN FRACTAL IMAGE COMPRESSION

Dietmar Saupe

Universität Freiburg, Institut für Informatik
Am Flughafen 17, 79110 Freiburg, Germany
saupe@informatik.uni-freiburg.de

ABSTRACT

In fractal image compression an image is partitioned into a set of image blocks, called ranges. The ranges are matched with blocks taken from a codebook of filtered and subsampled domain image blocks up to an affine transformation of intensity values. It is common practise in fractal image compression to include all 8 isometric versions of a codebook block in the codebook. It is reasoned that such enlarged domain pools yield better rate-distortion curves. However, this is not a valid argument supporting the use of isometries. A fair test must compare the performance of the method using a codebook including isometries with that obtained when using a plain codebook *of the same size.* We have performed such analysis and our results show that codebooks with isometries offer no advantages in terms of fidelity — in contrast to the prevalent belief. A similar study is carried out for the effects of including respectively excluding negative scaling factors in the fractal code.

1. INTRODUCTION

Fractal image compression [1] began with the visionary conception of Barnsley in 1987 and the ground breaking work of Jacquin in 1989. In the encoding a partitioning of the image into disjoint image blocks, called ranges, is used. A pool of (larger) image blocks, called domains, serves as a source of blocks from which ranges can be approximated as the sum of a DC component and a scaled copy of a domain block. In the work of Jacquin ranges and domains are square image blocks and the domain pool is enlarged by including all 8 rotated and reflected isometric versions of the domains. With this larger domain pool a better match for ranges can be achieved. Of course, the enlarged search causes the bitrate to increase. Still it can be shown that with the enlarged domain pool typically better rate-distortion curves can be achieved, of course, at the cost of increased encoding time. However, no analysis or experiment was given in order to show that enlarging the domain pool using isometries is better than enlarging a plain domain pool just by extracting more domains from the image. Yet almost all practitioners of fractal image compression are followers of the original approach using isometries, and it seems as if isometries are an intrinsic part of fractal image compression. However, this is not the case and, thus, the question we address in this study is whether there is a justification for the increased algorithm complexity involved with isometries.

In [2] Popescu, Dimca, and Yan consider an extension of square isometries by introducing rotations by an arbitrary angle. This is accomplished by referring to the Riemann Mapping Theorem. A square can be conformally mapped to a disk, then rotated by an arbitrary angle, and finally mapped back to the square by the inverse conformal transformation. It is shown that by considering such generalized square isometries improved performance in the rate-distortion sense is obtained. But, again, can the same improvement not also be attained by enlarging a plain domain pool just by extracting more domains from the image?

The usefulness of square isometries has been questioned before. In his review article [3] Jacquin presents statistics of a fractal code which show that isometries other than the identity transformation are seldom used. Bedford et al [4] argue that for a given domain block its rotated and reflected copies may already be available in the plain domain pool. Thus, allowing isometries in fact may introduce new redundancies. The experiments showed that the use of isometries improves the quality in terms of the rate-distortion function by a few percent, however, at the cost of much increased compute time due to the larger search space. Monro and Wolley investigated the usefulness of isometries in [5]. They come to the negative result, even in the rate-distortion sense: "When the extra bits required in the fractal code to specify the symmetries are taken into account, the fidelity/compression performance of fast

Pool size	Range size	Comp. ratio	512 × 512 Lenna PSNR (dB) with iso.	no iso.	diff.	512 × 512 Peppers PSNR (dB) with iso.	no iso.	diff.	512 × 512 Mandrill PSNR (dB) with iso.	no iso.	diff.
small	4×4	5.3	36.0	36.1	+0.1	34.6	34.8	+0.2	25.6	26.0	+0.4
	8×8	23.3	29.8	29.9	+0.1	29.6	29.5	−0.1	21.5	21.5	0.0
	16×16	102.2	25.1	25.1	0.0	24.3	24.4	+0.1	19.9	19.8	−0.1
medium	4×4	4.9	36.8	36.9	+0.1	35.9	35.8	−0.1	26.6	26.8	+0.2
	8×8	21.3	30.6	30.7	+0.1	30.7	30.4	−0.3	21.9	21.8	−0.1
	16×16	92.9	25.7	26.0	+0.3	25.1	25.5	+0.4	20.0	20.0	0.0
large	4×4	4.6	37.5	37.5	0.0	36.4	36.4	0.0	27.3	27.3	0.0
	8×8	19.7	31.3	31.4	+0.1	31.6	31.4	−0.2	22.2	22.1	−0.1
	16×16	85.2	26.4	26.5	+0.1	26.0	26.2	+0.2	20.2	20.1	−0.1

Table 1: Performance of fractal image compression with and without isometries. The partitioning of the images are of fixed range block size.

fractal image compression is degraded." However, their research considered only the "zero-searching" case employed in their Bath Fractal Transform.

Summarizing the previous work we have that up to this date no conclusive analysis has been presented to prove or disprove the positive effects of square isometries in fractal image compression. This empirical study settles the question with a negative answer. It is generally not justified to use isometries, since with a plain domain pool of the appropriate size one achieves the same results.

A similar analysis can be applied to the question of whether one should allow or disallow negative scaling factors in the fractal code. A negative scaling factor effectively reverses the dynamic part of a domain. Thus, similar to the case of isometries we may suggest the interpretation that with negative scaling factors the domain pool is "doubled". Without negative scaling factors the sign bits can be saved and, thus, the plain domain pool can be doubled producing a fractal code with unchanged compression ratio. We discuss the problem whether this simplification of the algorithm causes a degradation of image quality.

2. ISOMETRIES IN ENCODINGS WITH FIXED BLOCK SIZE

In this section we test the performance of fractal image compression with and without isometries using fixed uniform partitionings of block sizes ranging from 4 by 4 pixels up to 16 by 16 pixels. We test several images of different types of size 512 by 512 pixels. The domain pools are generated by shifting a window of twice the range size over the image with fixed spacings in the horizontal and vertical direction. A small size pool is obtained from non-overlapping domains, i.e., if d denotes the horizontal and vertical number of pixels in a domain, then the spacing in horizontal and vertical direction is also d pixels. For a 512 × 512 image this yields 4096 blocks of size 8 × 8, 1024 blocks of size 16 × 16, and 256 of size 32 × 32. With isometries we use spacings of $2d$ (horizontal) and $4d$ (vertical) obtaining a domain pool of about the same size counting isometries. The usefulness of isometries may depend on the domain pool size, so we use three different sizes of domain pools. The medium size pool is 4 times as large as the small one and obtained by halving the domain step sizes. For the large pool (16 times the size of the small one) step sizes are halved one more time. The computer code for the tests is a derivate of Fisher's quadtree program in [1]. In order to isolate the effects of isometries no classification strategy for complexity reduction was employed. All encodings in this paper were computed using full search, i.e., all domains (with all of their isometric versions, if needed) are checked.

The results for the well known test images Lenna, Peppers, and Mandrill are given in Table 1. The images are encoded 18 times with differing parameters: the domain pool size is large, medium, or small; the partitioning is made from blocks of size 4×4, 8×8, or 16×16, and the encodings use the codebook generated from the domain pool with isometries or a codebook of the same size obtained from a plain domain pool. The fractal codes are decompressed and the peak-signal-to-noise ratios (PSNR) are computed and listed.

The table shows that the resulting PSNR values are mostly identical for the encodings obtained with and without isometries. In some cases they differ by very little, mostly in favor of the plain domain pool.

512 × 512 Lenna				512 × 512 Peppers				512 × 512 Mandrill			
comp.	PSNR (dB)			comp.	PSNR (dB)			comp.	PSNR (dB)		
ratio	with iso.	no iso.	diff.	ratio	with iso.	no iso.	diff.	ratio	with iso.	no iso.	diff.
6.3	37.1	36.9	−0.2	4.9	36.1	36.1	0.0	4.8	27.0	27.0	0.0
10.9	35.5	35.5	0.0	10.8	36.7	36.6	−0.1	9.5	25.0	25.0	0.0
19.2	33.0	33.0	0.0	19.4	32.8	32.7	−0.1	18.4	22.5	22.6	+0.1
38.8	30.0	30.0	0.0	33.1	30.7	30.6	−0.1	41.5	20.9	20.9	0.0
80.4	26.6	26.8	+0.2	74.8	26.4	26.4	0.0	85.2	20.1	20.0	−0.1

Table 2: Performance of quadtree type fractal image compression with and without isometries.

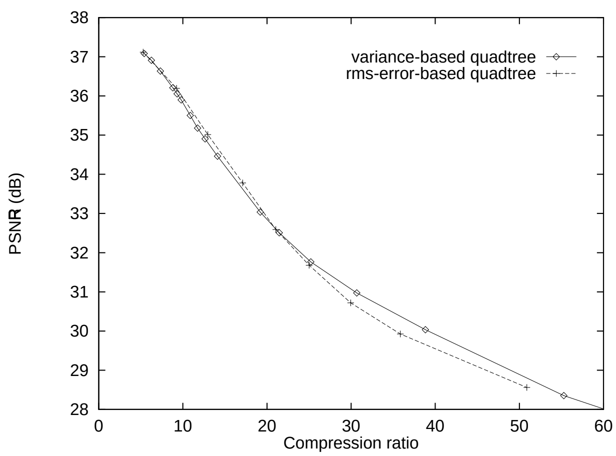

Figure 1: Comparison of the standard rms-error based quadtree encodings with those derived from quadtrees that are produced based on a variance criterion for the Lenna test image.

3. ISOMETRIES IN ENCODINGS WITH QUADTREES

From the results of the above section we conjecture that fractal image compression with adaptive quadtree partitionings also cannot profit from considering the isometric versions of codebook blocks. The experiments reported in this section confirm this statement. In fractal image compression the quadtree partitionings are typically derived using an rms-error-based criterion. A square image region is recursively subdivided into its four quadrants until a prescribed tolerance for the rms collage error is reached or until the quadrants have a minimal size such as 4×4 pixels. We found that when we replace the isometric codebook by a nonisometric one of the same size very different quadtrees are produced showing inferior rate-distortion curves. Thus, it is not a good idea to remove the isometries when using such quadtrees. However, this seems to be an artifact of the rms-error-based quadtree decision criterion rather than due to the missing isometries. To see this we generate the quadtrees in a different way based on the variance of the ranges. A square image region is recursively subdivided as long as the n-fold variance (where n is the number of pixels in the region) is above a given threshold. In [6] we showed that this modification not only accelerates the encoding significantly (all unsuccessful searches including rms-error computations are avoided), but also the quality in terms of rate-distortion curves is maintained (see Figure 1).

With such quadtree partitionings the experiment with and without isometries yields the results given in Table 2. The PSNR differences between results with and without isometries are very small and indifferent. Thus, in fact, also for adaptive quadtree partitions fractal image compression does not profit from using isometries in the domain pool.

4. ENCODINGS WITH NON-NEGATIVE SCALING FACTORS

In this section we consider disallowing negative scaling factors in exchange for enlarging the plain domain pool by a factor of two. No isometries are involved. This comparison is somewhat different in character from the one above for isometries. Here the encoding time is not decreased by disallowing negative scaling factors, as the least squares optimization for the scaling and offset coefficients yields a positive or a negative scaling factor. However, when search reduction strategies such as the classification in [1] are employed, considering only positive scaling factors does cut the computation time in half. Thus, for such cases we are fair and compare encodings with the same time complexity.

Table 3 gives the results for fixed block size partitionings and in Table 4 variable rate encodings with quadtrees are considered as in the last section. Here the situation is not as clear as with isometries, at least not for small ranges of size 4 by 4 pixels, where a quality deterioration of up to 1 dB is observed. However, this reduces the quadtree-based encodings in the low compression, high fidelity range only by 0.2 dB.

Pool size	Range size	Comp. ratio	512 × 512 Lenna PSNR (dB) +/−	+	diff.	512 × 512 Peppers PSNR (dB) +/−	+	diff.	512 × 512 Mandrill PSNR (dB) +/−	+	diff.
large	4 × 4	4.7	37.2	37.0	−0.2	36.1	35.2	−0.9	27.0	26.0	−1.0
	8 × 8	20.5	31.0	30.9	−0.1	31.0	30.9	−0.1	22.0	22.0	0.0
	16 × 16	88.9	26.1	26.0	−0.1	25.4	25.7	+0.3	20.1	20.1	0.0

Table 3: Performance of fractal image compression with (columns '+/−') and without (columns '+') scaling factors.

comp. ratio	512 × 512 Peppers PSNR (dB) +/−	+	diff.
6.9	35.6	35.4	−0.2
10.8	34.7	34.5	−0.2
19.4	32.8	32.7	−0.1
49.6	28.9	28.9	0.0
74.8	26.4	26.5	+0.1

Table 4: Quadtree type fractal compression with (col. '+/−') and without (col. '+') negative scaling factors.

5. DISCUSSION

Our investigation shows that there is no justification for the increased algorithm complexity involved with the incorporation of square isometries. With plain domain pools we can achieve the same or even better quality encodings at the same cost.

Are there any arguments left why one should still use isometries in fractal image compression? One might argue, that the classification scheme [1] reaps most of the benefits of isometries without paying the price for the enlarged search. This is accomplished by defining canonical block orientations from which one can deduce the most likely isometry that yields the least collage error for a given range-domain pair. However, we may use a similar technique to classify ranges and domains of the plain domain pool so that each range is compared only to domains in one of the classes. Thus, the same complexity reduction can be achieved with plain domain pools.

What remains is a small advantage regarding memory: with isometries we do not need to store sums of the pixel values and their squares for isometric codebook blocks as these sums are identical. This may be important when storage limitations are of major concern. Of course, this holds true only when square blocks are used, but not for other blocks, e.g., rectangular ones. Another possible advantage of isometries is given, when using fractal image compression with the orthogonalization technique of Øien and Lepsøy (see their contribution in [1]). In this case there is a natural limit to the size of the codebook. Essentially domain blocks are limited to blocks that are unions of ranges. With isometries we can make the codebook 8 times as large and there is no way to do this with a plain domain pool without violating the restrictions of the method with orthogonalization.

6. CONCLUSION

Isometries are not an intrinsic part of the fractal image encoding method. Moreover, there is no justification for the increased algorithm complexity involved with the incorporation of square isometries. With plain domain pools one can achieve the same or even better quality encodings at the same cost.

Acknowledgment. The author thanks Silke Jacob for programming and running the computer experiments.

7. REFERENCES

[1] Fisher, Y., *Fractal Image Compression — Theory and Application,* Springer-Verlag, New York, 1994.

[2] Popescu, D., Dimca, A., Yan, H., *Generalized square isometries — An Improvement for fractal image compression,* in : *Proc. 8th Int. Conf. Image Analysis and Processing,* San Remo, Sept. 1995.

[3] Jacquin, A. E., *Fractal image coding: A review,* Proceedings of the IEEE 81,10 (1993) 1451–1465.

[4] Bedford, T., Dekking, F. M., Breewer, M., Keane, M. S., van Schooneveld, D., *Fractal coding of monochrome images,* Sig. Proc. 6 (1994) 405–419.

[5] Monro, D. M., Wolley, S. J., *Rate/distortion in fractal compression: order of transform and block symmetries,* Proc. ISSIPNN'94 (Intern. Symp. on Speech, Image Processing and Neural Networks), Hong Kong (1994) 168–171.

[6] Saupe, D., Jacob, S., *Variance-based quadtrees in fractal image compression,* submitted, 1996.