Fast Fractal Image Coding Using Pyramids

H. Lin and A. N. Venetsanopoulos

Department of Electrical and Computer Engineering,
University of Toronto, Toronto, Ontario, Canada, M5S 1A4

Abstract: In this paper, we present a fast fractal image encoding algorithm which is based on a refinement of the fractal code from an initial coarse level of the pyramid. Assuming that the distribution of the matching error is described by an independent, identically distributed(*i.i.d.*) Laplacian random process, we derive the threshold sequence for the objective function in each pyramidal level. Computational efficiency depends on the depth of the pyramid and the search step size and could be improved up to two orders of magnitude compared with the full search of the original image.

1. Introduction

Fractal image compression is based on the observation that all real-world images are rich in affine redundancy. That is, under suitable affine transformations, large blocks of the image look like smaller ones of the same image. For a given image block, the compression algorithm needs to search through the whole image to find the best matching domain block under an affine transform. This search process is very computationally intensive as compared to the JPEG algorithm. Jacquin[1] used a classification scheme, which restricts the domain block search to the same class as the range block. As the number of the classes is only 3, the computational savings are relatively small. We therefore propose a fast encoding scheme based on pyramidal image representation. The search is first carried out on an initial coarse level of the pyramid. This initial search increases encoding speed significantly, because not only the number of the domain blocks to be searched is reduced, but also the data within each domain block are only $1/4^m$ of those in the finest level, where m is the pyramidal level. Then, only a few numbers of the fractal codes from the promising domain blocks in the coarse level are refined through the pyramid to the finest level with little effect.

2. Fast Pyramidal Domain Block Search Algorithm

Pyramidal image models employ several copies of the same image at different resolutions. Let $f(x,y)$ be the original image of size $2^M \times 2^M$. An image pyramid is a set of image arrays $f_k(x, y)$, $k=0, 1, ..., M$, each having size $2^k \times 2^k$. The pyramid is formed by low pass filtering and resolution subsampling of the original image. The pixel $f_k(x, y)$ at level k is obtained from the average of its four neighbours $f_{k+1}(x', y')$ at level $(k+1)$:

$$f_k(i,j)=\frac{1}{4}\sum_{r=0}^{1}\sum_{s=0}^{1}f_{k+1}(2x+r,2y+s) \tag{1}$$

The coarsest level (k=0) image has size 1 and represents the average grey level of the original image. The finest level image f_M is the original image of size $2^M\times2^M$. As the number of the levels decreases, the image details are gradually suppressed and spurious low spatial frequency components are introduced due to the effect of aliasing. Because the pyramidal structures offer an abstraction from image details, they have been proven to be very efficient in certain image analysis and image compression applications[2].

Our encoding process starts with partitioning $f(x,y)$ into a set of nonoverlapping range blocks of size $2^m\times2^m$. Similarly, the same image is partitioned into a set of overlapping domain blocks that are larger in size than range blocks to meet the contractivity condition. The previous study[3] showed that the general optimization objective function for the best matched domain block search can be written as:

$$E=\frac{1}{4^m}\sum_{x=0}^{2^m-1}\sum_{y=0}^{2^m-1}\left[D(x,y,s,t)-R(x,y)\right]^2 \tag{2}$$

where $D(x, y, s, t)=s\,f_{M-1}(x, y)+t$ is an affine of the scaled domain block and $R(x, y)=f(x, y)$ is the range block to be encoded. The fractal code is fully specified by parameters: (1)θ_i: the index of rotations/reflections; (2)D_{xi}, D_{yi}: the position of domain block D_i; (3)s_i: contractive factor; (4)t_i: grey level shift.

Instead of a constant contractive factor s, a nonlinear contractive function $s(x, y, a, b, c)$ can be used for fast decoding [3]:

$$s(x,y,a,b,c)=\pm\frac{1}{1+e^{ax+by+c}} \tag{3}$$

where (a, b, c) are parameters of the contractive function. From the original image a pyramid is created, the depth of which is determined by the range block size. Because the range block is defined in the image, the range block pyramid will be contained in the image pyramid with the k-th level of the range block pyramid corresponding to the $(M-m+k)$-th level of the image pyramid. Instead of a direct search of the minimum of the objective function at the finest level m, we propose a fast algorithm by introducing a smaller, approximate version of the problem at a coarser level k of the range block pyramid:

$$E^k=\frac{1}{4^k}\sum_{x=0}^{2^k-1}\sum_{y=0}^{2^k-1}\left[D^k(x,y,s^k,t^k)-R^k(x,y)\right]^2 \tag{4}$$

for $k_0\le k\le m$. Therefore, at range block pyramid level k, the encoding amounts to finding the best matching domain block of size $2^k\times2^k$ in the image of the size $2^{M-m+k}\times 2^{M-m+k}$. For example, for an original image of size 512×512 (M=9) and range block size 64×64 (m=6), the search complexity at k_0=2 is that of the image size 64×64 and the

range block of size 4×4. The $k=k_0$ level of the range block pyramid is said to be initial and every location of the image from the $(M-m+k_0)$-th level of the image pyramid needs a test. Now, generate a $2^{k+1}\times 2^{k+1}$ promising location matrix G^{k+1}:

$$(G^{k+1})_{2u,2v}=\begin{cases}1, & if\ E^k(u,v)<T^k\\ 0, & otherwise\end{cases} \tag{5}$$

where (u, v) is the upper left corner coordinates of the domain block and T^k is the threshold at level k. Matrix G^{k+1} is used as a guide in the search of the domain location at the next level $k+1$. Tests are to be performed only at the locations (i, j) for $(G^{k+1})_{i,j}=1$ and its neighbour locations. Other parameters P^k of the promising locations are also propagated to P^{k+1} for further refining at level $k+1$. For the of affine mapping, we have $\theta^{k+1}=\theta^k$, $D_x^{k+1}=2D_x^k$, $D_y^{k+1}=2D_y^k$, s^{k+1} and t^{k+1} need to reevaluate. In the case of nonlinear contractive functions, the initial parameters at level $(k+1)$ are: $a^{k+1}=\frac{1}{2}a^k$, $b^{k+1}=\frac{1}{2}b^k$, $c^{k+1}=c^k$ and $t^{k+1}=t^k$. The ½ gain before the a^k and b^k is due to the resolution increase in the x and y directions. The algorithm provides a gradual refinement of the fractal code. The process is repeated recursively until the finest level m is reached as shown in Fig. 1. The iterations are over the promissing locations. At the finest level, if there exist more than one locations (u, v) such that $(G^M)_{u,v}=1$, select the parameters with the smallest match error as the fractal code. An important feature of the algorithm is estimating of the threshold T^k. The next section shows how to estimate these thresholds under certain assumptions.

3. Determination of Thresholds

Let $\mathbf{x}_i$ denote the grey level difference of a pixel between an affine transformed domain block D and a range block R at the finest level m, i.e., $\mathbf{x}_i = D_i - R_i$, for $i=0,1,\ldots, (2^m\times 2^m-1)$. At the match location (u^*, v^*), $\mathbf{x}_i$ is significantly less correlated. Thus, we may consider $\mathbf{x}_i$ as independent, identically distributed(*i.i.d.*) random variables with an approximately Laplacian density function:

$$f(x)=\frac{\alpha}{2}e^{-\alpha|x|} \tag{6}$$

$f(x)$ has mean $\mu_0=0$ and variance $\sigma_0^2=2/\alpha^2$. Our experimental data showed a reasonable approximation to the density function. Then, it can be shown that the function of the random variable $\mathbf{y}_i=\mathbf{x}_i^2$ is exponentially distributed and has mean $\mu_y=\sigma_0^2$ and variance $\sigma_y^2=5\sigma_0^4$. The next step towards the goal is to find the distribution of the mismatch measure as in (2) which can be rewritten as:

$$E=\frac{1}{n}\sum_{i=0}^{n-1}x_i^2=\frac{1}{n}\sum_{i=0}^{n-1}y_i \tag{7}$$

where $n=2^m\times 2^m$. By the central limit theorem, which says that the density of the sum of n independent random variables tends to a normal density as n increases, regardless of the shapes of the densities of the given random variables[4], E is approximately

normal with:

$$\mu_E=\sigma_0^2, \quad \sigma_E^2=\frac{5}{4^m}\sigma_0^4 \tag{8}$$

Let P_α be the probability of finding the best match (u^*,v^*), i.e. $P(E < T^m)= P_\alpha$, then the threshold will be:

$$T^m=\mu_E+x_\alpha\sigma_E=\sigma_0^2(1+\frac{\sqrt{5}}{2^m}x_\alpha) \tag{9}$$

where x_α is the P_α point of standard normal distribution. For example, when P_α=0.9, x_α=1.28.

It can be shown[5] that the thresholds at a coarse level k are:

$$T^k=\frac{\sigma_0^2}{4^{m-k}}(1+\frac{x_\alpha}{2^{(k+\frac{1}{2})}})^2 \tag{10}$$

for $k=k_0, ..., m$-1.

4. Computational Efficiency

The computational efficiency of the pyramid algorithm can be evaluated based on the following theoretical considerations. For a given range block, assume each domain block needs the same number of operations to determine the parameters. Then the computational cost is proportional to the product of the number of domain blocks searched and the number of pixel in each block. For an original image of size $2^M\times2^M$ and range block of size $2^m\times2^m$, when D_i is chosen in each dimension twice the size of the R_i, the search domain image is $2^{M-1}\times2^{M-1}$ with the contracted domain block of size $2^m\times2^m$. This number becomes:

$$C_1=8\,(\frac{2^{M-1}-2^m}{h}+1)^2\,2^{2m} \tag{11}$$

where h is the step size of the domain block search. When a pyramidal search is applied, the computational resources for the algorithm are determined by the average number of the promising locations n_p on every pyramid level and the number of the shifts n_s around each promising location:

$$C_2=8\,(\frac{2^{(M-m+k_0-1)}-2^{k_0}}{h(k_0)}+1)^2\,2^{2k_0}+n_pn_s\sum_{i=k_0+1}^{m}2^{2i} \tag{12}$$

where the first term corresponds to the initial step of the algorithm, testing every domain block on the initial range pyramid level k_0. The search step size $h(k_0)$ is related to th finest level step size h as follows:

$$h(k_0) = \max(1, \frac{h}{2^{(m-k_0)}}) \tag{13}$$

where we assume only the integer search step is used in level k_0, although, in general, the search with sub-pixel accuracy is possible.

The number of operations required to create an image pyramid will be proportional to the number of pixels:

$$C_3 = K_1 \sum_{i=M-m+k_0}^{M-1} 2^{2(M-i)} \tag{14}$$

Compared with the optimization operation during the domain block search, this part can be neglected.

The benefit in computational saving using pyramids relative to the full search of the original image is estimated as:

$$Q = \frac{C_1}{C_2 + C_3} \approx \frac{C_1}{C_2} \tag{15}$$

For a given image and range block size, the value of Q depends on the depth of the pyramid and the search step size. For example, for the image of size 512×512, range block 32×32, when h=2, n_p=20, n_s=16 and k_0=2, the computational saving factor will be 194. The actual Q value is expected to be smaller than the theorical one. For example, encoding a 32×32 range block with affine contractive mapping needs 95.89 *CPU* seconds by full search and 0.79 seconds by pyramidal search (Serial implementation on KSR1 parallel computer without optimization of codes), which gives Q=121.

5. Experimental Results

Fig. 2 is 512×512×8 bits original Lenna image. Quadtree partition is used for range blocks. The initial range block size is 64×64. The mean square error was determined for each range block. Blocks which had an error exceeding 81 (corresponding rms value 9.0) and were larger than 8×8 in size were split. Fig.3 shows our reconstructed image using nonlinear contractive function by full search algorithm at bit rate 0.2 bpp (compression ratio 40:1) and PSNR=30.2 dB. Fig. 4 is the result of this paper at the same bit rate and PSNR=29.9 dB. Thus, the pyramid search algorithm is quasi-optimal in terms of minimizing the mean square error. The main advantage of the pyramid algorithm is the greatly decreased computational complexity, when compared to full search.

References

1. A. E. Jacquin: Image coding based on a fractal theory of iterated contractive image

transformation. IEEE Trans. image Process. 1 (1992) 18-30
2. P. J. Burt, and E. H. Adelson: The Laplacian pyramid as a compact image code. IEEE Trans. Comm. 3 (1983) 532-540
3. H. Lin and A. N. Venetsanpoulos: Incorporating nonlinear contractive functions into the fractal coding. Proceedings of the International Workshop on Intelligent Signal processing and Communication Systems, Seoul, Korea. (1994) 169-172
4. A. Papoulis: Probability, Random Variables, and Stochastic Processes, McGrawHill, Inc. (1991)
5. H. Lin and A.N. Venetsanpoulos: Fast Fractal Image Compression Using Pyramidal Search. in: Scientific Information Guild (Ed.): Circuits & Systems. India: Research Signpost (to appear)

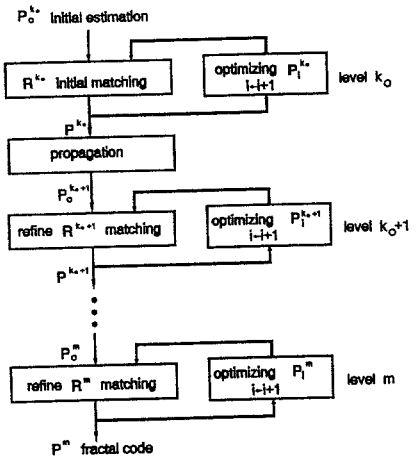

Fig. 1. Refining the fractal code from coarse to fine level

Fig. 2. Original image

Fig. 3. Full search, 0.2 bpp, 30.2 dB

Fig. 4. Pyramid search, 0.2 bpp, 29.9 dB