A NOVEL METHOD IN REDUCING THE COMPLEXITY OF
FRAC TAL ENCODING

L.K. Ma, O.C. Au*, and M.L. Liou**
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong.
Tel: +852 2358-7053*, +852 2358-7055**
Email: eeau@ee.ust.hk*, eeliou@ee.ust.hk**

ABSTRACT
Fractal coding is a promising technique for image compression. However, one of the challenges for cost effective implementation is to reduce the huge computational complexity of the encoder. In this paper, we propose a novel algorithm to address this issue. Firstly, we replace mean square error with mean absolute error as distortion measure to reduce multiplication. Secondly, we use statistical normalization to eliminate the need to compute the scaling factor and offset during the search. Thirdly, we change the domain block search to range block search to reduce memory requirement. Simulation results suggest that our algorithm can reduce computation by three order of magnitude for a QCIF image with negligible visual degradation.

1. INTRODUCTION
In recent years, there has been much interest in applying fractals to encode images and video sequences due to the potential of achieving very large compression ratios. Instead of coding the image itself, fractal coding achieves compression by encoding the self-similarity structure of the image[1, 2]. One major drawback of fractal encoding is its large computational complexity due to the search for suitable affine contractive mappings. While much work has been done to reduce the computation complexity of fractal encoding, most of them only try to reduce the number of comparisons but each comparison itself requires considerable computation. In this paper, we propose a novel algorithm to reduce the amount of computation by computing a different distortion measure, a different scaling factor and a different offset during the search. Also, domain block search is changed to a range block search to reduce memory requirement.

In the traditional fractal encoding[2], an image is divided into non-overlapping blocks called range blocks. For each range block, an exhaustive search is performed to find the optimal domain block which minimizes the mean square error (MSE) under certain affine contractive mapping. For each possible range block, the brute force domain-block search involves computing the optimal scaling factor and offset for each of the eight possible rotation and flipping isometrics. Let the image size be P×Q, the range block size be N×N and the domain block size be 2N×2N. Assuming P and Q to be much larger than N, the brute force search requires approximately
\[
\left(\frac{P}{N}\right)\left(\frac{Q}{N}\right)(P−2N+1)(Q−2N+1)(9N^2+48) = 9P^2Q^2
\]
multiplications and
\[
\left[\left(\frac{P}{N}\right)\left(\frac{Q}{N}\right) + 2(P−2N+1)(Q−2N+1)\right]N^2
\]
additions. For a QCIF sized image (P=176, Q=144) with N=8, this corresponds to 5.8x10^6 multiplications and 5.1x10^6 additions, which is excessive for encoding an image.

2. MOTIVATION
Traditionally mean square error (MSE) is used as distortion measure in the domain block search because of the relatively simple mathematical analysis. One drawback of MSE is the need of much multiplication in the encoding. To reduce multiplication, we propose to use mean absolute error (MAE) instead as the distortion measure. In general, neither MSE nor MAE are good measure of perceptual distortion but both are reasonably good in measuring distortion.

Traditionally, the optimal scaling factor s and offset α in the affine transform are derived by minimizing the MSE, which is the distortion measure used in the domain block search. These optimal scaling factor s and offset α must be computed for each possible pair of domain block and range block, which is computationally very expensive. As we change the MSE to MAE, the minimization is more difficult and the resulting equations even more difficult to
compute.
As a result, we choose not to minimize the MAE. Instead, we use $s$ and $o$ to match the mean and variance of the domain block and range block. This is reasonable because, when a domain block is perceptually close to a range block, they should have similar mean and variance. However, straightforward implementation would still require significant computation during the search. To reduce computation, we normalize all the blocks to zero mean and unit variance before the search. This way no scaling and offset adjustment are needed during the search and the block searching is similar to motion estimation in Motion Picture Expert Group (MPEG) encoding.

One problem with the normalization is that it takes excessive memory to store all the normalized domain blocks. To control the memory requirement, we change the domain block search to a range block search. Before the search, all the range blocks are normalized and stored. During the search, only one domain block needs to be normalized and stored at a time, which greatly reduces the memory requirement.

3. Proposed Fast Fractal Encoding

Here is the proposed fast fractal encoding algorithm:

1. (Statistical normalization) For each range block $r$, compute the mean $m_r$ and variance $\sigma_r^2$ and transform each pixel $x_{r,i}$ in $r$ by

   $$ f(x_{r,i}) = \frac{x_{r,i} - m_r}{\sigma_r} $$

   to yield a zero-mean, unit-variance normalized range block $r_{nor}$. Store all normalized range blocks $r_{nor}$ with associated $m_r$ and $\sigma_r^2$. For each range block, initialize the “best domain block” to be the first domain block with infinitely large mean absolute error $MAE_r$.

2. (Range block search) For each domain block $d$ (decimated from $2N \times 2N$ to $N \times N$), compute the mean $m_d$ and variance $\sigma_d^2$ and transform each pixel in $d$ by

   $$ f(x_{d,i}) = \frac{x_{d,i} - m_d}{\sigma_d} $$

   to yield a zero-mean, unit-variance normalized domain block $d_{nor}$. For each isometric of $d$, compute the mean absolute error $MAE_1$ and $MAE_2$ between $d_{nor}$ and each of the stored range block $r_{nor}$ as follows:

   $$ MAE_1 = \sum_{i=1}^{N^2} |x_{r,i} - x_{d,i}| $$

   $$ MAE_2 = \sum_{i=1}^{N^2} |x_{r,i} + x_{d,i}| $$

   If either $MAE_1$ or $MAE_2$ is less than $MAE_r$ of $r_{nor}$ set $MAE_r$ to be the smaller of $MAE_1$ and $MAE_2$ and set the “best domain block” of $r_{nor}$ to be $d$ with mean $m_r$ and variance $\sigma_r^2$. Store the current isometric, and whether $MAE_1$ or $MAE_2$ is used.

3. Stop. The best affine contracitive mapping for each range block $r$ is the associated “best domain block” $d$, scaled and offset adjusted as follows:

   $$ r = \frac{(\sigma_d^2)}{(\sigma_r^2)}(d_{nor} - m_d)(-1)^i + m_r $$

   where $d_i$ is $d$ with the associated isometric. $i=0$ if $MAE_1$ is used and $i=1$ if $MAE_2$ is used.

In the algorithm, both $MAE_1$ and $MAE_2$ are considered because both $d_{nor}$ and $d_{nor}$ are zero-mean, unit variance blocks, equally suitable for matching the normalized range blocks.

The proposed algorithm is found in experiments to yield slightly lower visual quality than the traditional method. To improve the visual quality, we modify step 3 of the algorithm by using the traditional formula for optimal scaling factor $s$ and offset $o$, minimizing the MSE rather than matching the mean and variance. This is done only once after the range block search and thus has negligible contribution to overall computation.

4. Computational and Memory Requirements

Assuming that $P$ and $Q$ are much larger than $N$, the proposed algorithm requires approximately

$$ (P - 2N + 1)(Q - 2N + 1)(N^2 + N + 1) $$

$$ + \left( \frac{P}{N} \right) \left( \frac{Q}{N} \right) (2N^2 + 1) = N^2PQ $$

multiplications and
These suggests that the proposed algorithm can indeed reduce the amount of multiplication significantly with negligible visual degradation.

6. ACKNOWLEDGEMENT

This work was supported partially by the Research Grant Council and partially by the Hongkong Telecom Institute of Information Technology under contract HKTIT93/94 EG02.

7. REFERENCES


<table>
<thead>
<tr>
<th>Image</th>
<th>Encoding method</th>
<th>Bit per pixel</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>traditional</td>
<td>0.453</td>
<td>30.54</td>
</tr>
<tr>
<td>Pepper</td>
<td>traditional</td>
<td>0.451</td>
<td>29.86</td>
</tr>
<tr>
<td>Pepper</td>
<td>proposed</td>
<td>0.450</td>
<td>29.75</td>
</tr>
<tr>
<td>MissA</td>
<td>traditional</td>
<td>0.417</td>
<td>33.77</td>
</tr>
<tr>
<td>MissA</td>
<td>proposed</td>
<td>0.413</td>
<td>33.78</td>
</tr>
</tbody>
</table>

Table 1: Simulation Results
Fig. 1 “Lenna” (512x512) coded with traditional method. (BR=0.452 bpp. PSNR=30.54 dB)

Fig. 2 “Lenna” (512x512) encoded with proposed fast algorithm. (BR=0.452 bpp. PSNR=30.56 dB)

Fig. 3 “Pepper” (512x512) coded with traditional method. (BR=0.451 bpp. PSNR=29.86 dB)

Fig. 4 “Pepper” (512x512) encoded with proposed fast algorithm. (BR=0.450 bpp. PSNR=29.75 dB)