Improved Algorithm of Image Compression Based on IFS.

Vladimir Grebenik
Moscow State University
Introduction

Each encoded image in fractal compression is represented via contractive mapping.

All real implementation employ blockwise coding: the entire image is split on nonoverlapping block called ranges that tile entire image, and possibly overlapping domains that can be taken from any area of the original picture.
1. Mathematical background

Let us consider original signal
\[ x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n \]
and affine mapping
\[ W : x \rightarrow W(x) = Ax + b \]
that satisfies contractivity constraint:
\[ \exists s \in [0,1) : d(W(x), W(y)) < sd(x, y) \ \forall x, y \in \mathbb{R}^n \]

Then exists unique fixed point
\[ \exists x_f = W(x_f) \]
1. Mathematical background (cont)

Encoding process is the search of $A$ and $b$ such that $d(x_0, x_f)$ is minimal.

Reconstruction of original image:

$$x_f = W(x_f)$$

For any initial approach $x_0 \in \mathbb{R}^n$ the sequence $x_k = W^k(x_0)$ converges to the fixed point

$$x_f = \lim_{k \to \infty} x_k$$
1. Mathematical background (cont)

*Collage theorem* allows to minimize $d(x_0, W(x_0))$ instead of $d(x_0, x_f)$.

Although suboptimal, this approach can be really used in practice.
2. Improved fractal algorithm

Main features

- Independent encoding of color planes
- Tree-like data structure optimized for domain search
- Classification of range cells
- Hybrid algorithm that consists of:
  - Fast fractal encoding
  - Original algorithm for restoration of smooth cells
  - Vector Quantization (VQ) method for edge-like cells
2. Improved fractal algorithm (cont)

"Normalizing" of domains

All domains are transformed into common representation with the same dynamic range and average color

\[ D^1[i] = \alpha D^0[i] + \beta \]

\[ \alpha = \frac{C_{max} - C_{min}}{\max_{j=0,N} D^0[j] - \min_{k=0,N} D^0[k]} \]

\[ \beta = C_{avg} - \frac{\alpha}{N} \sum_{l=0,N} D^0[l] \]

\[ i = 0..N, C_{max} = 255, C_{min} = 0, C_{avg} = 128 \]
2. Improved fractal algorithm (cont)

Data structure

Every domain is split into 4 quadrants and average color for each quadrant is calculated:

Each of these averages is put into appropriate level of the tree. Level 5 stores numbers of domains that have common quadrant structure. Domains are ordered by dynamic range.
2. Improved fractal algorithm (cont)

Classification of Ranges

All ranges are split into 3 classes based on their dynamic range:

- **edge** (big dynamic range)
  - VQ method is applied
- **smooth** (small dynamic range)
  - Smooth encoding method is applied
- **midrange** (average dynamic range)
  - Main fractal encoding is applied
2. Improved fractal algorithm (cont)

Domain-range matching

To find a match for a range, it is transformed:

\[ d(R_i, R^D_i) \rightarrow \min, \]
\[ R^D_i[l] = \alpha I^i_k(D^j_i[l]) + \beta \]

\[ \alpha = \frac{\max R_i[p] - \min R_i[q]}{\max D^j_i[r'] - \min D^j_i[t']} \]

\[ \beta = \frac{1}{N} \sum_{u=0,N} R_i[u] - \frac{\alpha}{N} \sum_{v=0,N} D^j_i[v] \]
2. Improved fractal algorithm (cont)

Tree search procedure

- “Normalize” a Range, calculate quadrant averages (QA);
- For each isometry do:
  - Find a match for QA1 on the first level of the tree;
  - Obtain 2\textsuperscript{nd} level items and repeat the same for QA2;
  - Obtain 3\textsuperscript{rd} level items and repeat the same for QA3;
  - Obtain 4\textsuperscript{th} level items and repeat the same for QA4;
- Obtain list of domains and choose ones with appropriate dynamic range;
  - For all chosen domains calculate mapping parameters and choose one that gives minimal distortion
2. Improved fractal algorithm (cont)

Restoration of smooth areas

Each smooth area is divided into 4 quadrants and for each of them linear smoothing is performed, taking into account "main" colors of smoothened area itself and neighbouring areas as well:

\[ R_A[i, j] = \frac{C_1 - C_0}{h} i + \frac{C_2 - C_0}{h} j + \frac{C_0 - C_1 - C_2 + C_3}{h^2} ij + C_0 \]
2. Improved fractal algorithm (cont)

Vector Quantization

Our implementation uses special set of 16 vectors that are constructed to represent well edge-like blocks of image. Each vector is 2-color; these colors take part of the fractal code built.
3. Implementation and results

Color image 600x600 is encoded in 60 sec. (about 1100 range matches per second) Range size used: 4, distance between neighbouring domains: 8.

Achieved compression: 15-30 times with PSNR about 31db.

Full search would need $1.2 \times 10^8$ range-domain comparisons ($1.9 \times 10^9$ "elementary" comparisons).

The proposed algorithm uses $4.8 \times 10^6$ "elementary" comparisons. Amount of "effective" compare operations and ones used in tree navigation are approximately equal.
3. Implementation and results (cont)
3. Implementation and results (cont)
4. Conclusion

Presented algorithm delivers fast compression on color images.

Further improvements can be done by studying the correlation between color planes and human image recognition features.