http://www.informatik.uni-leipzig.de/cip/
Germany
Universität Leipzig
Institut für Informatik
Raouf Hamzaoui
Conclusion.

Efficient implementation of local search.

Local search.

Decoding complexity reduction.

Collage coding.

Principle of fractal image compression.

Outline
Encoder: $\mathcal{L} \leftarrow f : f \leftarrow \mathcal{F}$

Decoder: $\mathcal{L} \leftarrow f$.

$f \approx \mathcal{L} f$ (3)

1) Description of $\mathcal{L}$ uses less bits than that of $f$.

2) $\mathcal{L}$ contraction for a norm on $\mathcal{F}$.

3) Initial image $\left( (u)f \right) \mathcal{L} = (1+u)f$ where $(u)f \mid u \left( f \right) \mathcal{L} = \mathcal{L} f$.

Initial image is an arbitrary.
\{o_1, \ldots, o_t\} = \emptyset

called scaling factors

\{s_1, \ldots, s_t\} = S

\mathcal{D}_{\alpha} \mathcal{I}, \mathcal{D}_\beta \mathcal{I} \subseteq \mathcal{D} \mathcal{I} \subseteq \mathcal{D}_\beta \mathcal{I}, \mathcal{D}_{\alpha} \mathcal{I}

\{R_1, \ldots, R_t\} \text{ partition of } \mathcal{I}

A fractal transform \mathcal{I} : \mathcal{I} \to \mathcal{I} is characterized by:

\textbf{Fractal Transform}
$D_1,\ldots, D_n$ are arbitrary $2^{n+1} \times 2^{n+1}$ squares.

$\{R_1,\ldots, R_m\}$ uniform partition into $2^n \times 2^n$ squares.

$I$ image support of size $2^N \times 2^N$.

Example
where $W$ is a $2^n \times 2^n$ downsampling matrix.

$$1_{(\tau)0} + (\tau_{\nu}D) f \times W(\tau)s = \nu f \cdot \tau x$$

$$\{\nu \in \mathbb{R}, \cdots, \nu_{\ell} \} \subseteq I$$

For all $f \in \mathcal{F}$ and all $\ell \in \ell$ such that $f \leftrightarrow \mathcal{F} : \ell$.

In $\nu_{\ell}(\mathcal{O} \times \mathcal{S} \times \mathcal{D})$ we associate a fractal transform

$$(((\nu_{\ell}o)(\nu_{\ell}s)(\nu_{\ell}D)) \cdots ((\nu o)(\nu s)(\nu D)) \cdots ((\ell o)(\ell s)(\ell D))$$

To each tuple of fractal transforms.
(a) $f(0)$, (b) $f(1)$, (c) $f(2)$, (d) $f(10)$, (e) Original $f$.  

Decoding
For a given partition \( \mathcal{P} \), feasible solutions:

\[
\min_{\mathcal{F} \in \mathcal{F}} \| f - f^* \| = (\mathcal{L})^T \mathcal{E} \Theta^T \mathcal{E} \mathcal{F}
\]

**Goal:** Find \( \mathcal{T} \in \mathcal{T} \) that solves the combinatorial problem.

**Given:** \( f \) target image, \( \mathcal{T} \) finite set of fractal transforms.

Encoding problem
Factor of $\mathcal{L}$.

Where $\mathcal{L}$ is a contraction for the norm, and $s(\mathcal{L})$ is the contraction

$$\|(*f)_{\mathcal{L}} - *f\| \frac{(\mathcal{L})^s - 1}{1} \geq \|\mathcal{L}f - *f\|$$

Motivation for collage coding:

Minimize the contraction of $\|(*f) - *f\|^2$ instead of

Greedy algorithm (collage coding) finds suboptimal solution.

Collage coding
\[
\begin{align*}
\min_{\mathcal{H}} & \quad \mathbb{E} \left\| (I(\theta) o + (\theta) A \ast f W(\theta) x) - \mathcal{H} \ast f x \right\|_{O \times S \times A \mathcal{E}(\theta) o (\theta) s (\theta) A} \\
\text{independent minimization problems} & \\
\text{Optimal parameters in collage coding are solutions of the} & \\
\mathcal{H} & \\
\begin{align*}
\mathbb{E} \left\| (I(\theta) o + (\theta) A \ast f W(\theta) x) - \mathcal{H} \ast f x \right\|_{\mathcal{H}_u} & = \\
\mathbb{E} \left\| \mathcal{H} \ast f L x - \mathcal{H} \ast f x \right\|_{\mathcal{H}_u} & = \mathbb{E} \left\| \ast f L - \ast f \right\| 
\end{align*}
\end{align*}
\]
Find a domain \( \mathcal{D} \) that minimizes the error.

Quantitize \( s \) and \( o \) in \( s \) and \( o \), yielding a scaling factor \( s^* \) and an offset \( o^* \).

\[
\min s \in \mathcal{R} \quad \min o \in \mathcal{R} \quad \min \quad \min \quad \min
\]

Let \( s \) and \( o \) denote the solutions of the least squares problem:

\[
\min \| y - X \beta \|^2
\]

Denote \( M \in \mathcal{D} \).

\[
\min \| D \|^2
\]

Solving collage coding with least squares
We need only one image array in the decoding.

\[ o + ( ( I + m, I + l ) ( I + y ) f + ( m, I + l ) ( I + y ) f + ( m, I ) ( I + y ) f ) \frac{4}{s} = ( f, I ) ( I + y ) f \]

We can compute

\[ ( f, I ) ( I + y ) f \]

Suppose that the intensities of pixels \(( I, m)\) are computed before the pixel intensity of \(( I, 0)\) is computed. Suppose that the scaling factor and \( o \) is an offset.

\[ o + ( ( I + m, I + l ) ( y ) f + ( m, I + l ) ( y ) f + ( m, I ) ( y ) f ) \frac{4}{s} = ( f, I ) ( I + y ) f \]

where \( s \) is a scaling factor and \( o \) is an offset.

\[ ( I, m) \]

Suppose that the intensities of pixels \(( I, m)\) are arbitrary initial image.

\[ ( 0 ) f \]

Let \( I \) be a fractal transform and (where (0) f ) I

Decoding complexity reduction
\[ nq + \sum_{\alpha}^{\gamma x} \sum_{\beta}^{n} \cdot \sum_{\gamma}^{\gamma} + (1 + \gamma x n y) = n x \]

For \( n = 1, \ldots, N \) or \( N \in \mathbb{Z} \).

New method:

\[ nq + \sum_{\alpha}^{\gamma x} \sum_{\beta}^{n} \cdot \sum_{\gamma}^{\gamma} = (1 + \gamma x) \]

That for \( n = 1, \ldots, N \) or \( N \in \mathbb{Z} \).

Then there exists a matrix \( A \) such that \((nq) = (\gamma x)^{n} \cdot q \) = \( \alpha \) and a vector \( \beta \) that exists.

Using an ordering of the pixels, let \( x_{(\gamma)} \) be the vector corresponding to the image iteration. Therefore, iterating over the pixels of the image.
to that of the conventional method [Hamzag7].
The rate of convergence of the new method is greater than or equal

If all scaling factors have the same sign, then the asymptotic
Convergence is faster.

The conventional decoding converges to the same fixed point as con-

Proposed decoding converges to the iterative method

\[ q_{I-}(T - I) + (\gamma) x \Omega_{I-}(T - I) = (1 + \gamma) x \]

Then the proposed decoding corresponds to the iterative method

\[ \begin{cases} \alpha^n q & \text{if } n \geq \alpha, \\ 0 & \text{otherwise} \end{cases} = \alpha^n \Omega \]

Let \( \Omega + T = \Lambda \)
After quantization, the PSNR improvement over collage coding is insignificant.

Fix the domain blocks and optimize the scaling factors and the offsets (considered as continuous variables) by gradient descent methods [Vrscaj, Sauppe 99].

Start from the solution found by collage coding.
for all range blocks \( H_i, i = 1, 2, \ldots, n \)

\[
\| \left( I_i^{(i)0} + (i)D \right)^{1-u} f x_i W_i^{(i)S} - \hat{r}_H \| f x \|_O \times S \times D \| ((i)0^{(i)S})^{(i)D} \|_\min
\]

tion problem

each step \( n \), new parameters are found by solving the minimiza-

Construct a sequence of fractal transforms \( I_1, I_2, \ldots \), where at

Start from an original solution \( I_0 \), found by collage coding.

parameters.

Barthel [Barthel94] and Lu [Lu97] suggested to update all pa-
The procedure is time expensive because every step corresponds to a new encoding of the test image.

No guarantee that the reconstruction error decreases after each step.

Substantial PSNR improvements over collage coding.
Problem is NP-hard [Ruh97].

\[ \| Jf - \star f \| = \min_{\mathcal{E} \in \mathcal{E}} \int \mathcal{H} \]

Given: \* target image \( f \) set of \( \mathcal{H} \) fractal transforms.

Combinatorial optimization
If a better solution is found, adopt it and repeat the search from the current solution.

Search for a better solution in its neighborhood.

Define a neighborhood of a solution.

Local search
Local search algorithm
Left: collage coding. Right: local search
dark 512 x 512 images.
Improvement over collage coding can be up to 0.8 dB for stan-
Performance
<table>
<thead>
<tr>
<th>Case</th>
<th>Proposed PSNR</th>
<th>Lu PSNR</th>
<th>College PSNR</th>
<th>Compression Ratio</th>
<th>Image Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.01</td>
<td>26.77</td>
<td>26.63</td>
<td>24.54</td>
<td>18.96:1</td>
<td>256 x 256</td>
</tr>
<tr>
<td>24.92</td>
<td>24.65</td>
<td>24.54</td>
<td>20.45:1</td>
<td>1024</td>
<td>512 x 512</td>
</tr>
<tr>
<td>26.77</td>
<td>26.63</td>
<td>24.54</td>
<td>20.45:1</td>
<td>1024</td>
<td>512 x 512</td>
</tr>
<tr>
<td>30.01</td>
<td>26.77</td>
<td>26.63</td>
<td>24.54</td>
<td>18.96:1</td>
<td>256 x 256</td>
</tr>
<tr>
<td>24.92</td>
<td>24.65</td>
<td>24.54</td>
<td>20.45:1</td>
<td>1024</td>
<td>512 x 512</td>
</tr>
<tr>
<td>26.77</td>
<td>26.63</td>
<td>24.54</td>
<td>20.45:1</td>
<td>1024</td>
<td>512 x 512</td>
</tr>
</tbody>
</table>
The candidate transformation is found by modifying the parameter entries of only one range.

Local search requires successive computations of the fixed points.

Time complexity
Use the dependence graph.

\[ \| \tilde{y} \|_2^2 - \| y^0 \|_2^2 \leq \| \tilde{y} \|_2^2 \leq \| y^0 \|_2^2 \]

Sort ranges according to decreasing error

No computation of the fixed point if parameters are unchanged.

Use the pixel update decoding.

Start the iterations from the current fixed point.

Complexity reduction
Starting from vertex $R^i$, a breadth-first traversal of the dependence graph of the code can be implemented as

$$Lf^{\circ} \leftarrow \cdots \leftarrow (uLf)^{\circ}L^{\circ} \leftarrow (uLf)^{\circ}L \leftarrow uLf$$

are the ranges and $R^i$ is a child of $R^j$. A dependence graph of a code is a directed graph where vertices encode $R^j$ overlaps $R^i$. A range $R^j$ is called a child of a range $R^i$ if the domain that have to be updated.

only ranges whose domains overlap $R^i$ have to be updated:

$$((uLf)^{\circ}L^{\circ}L \leftarrow (uLf)^{\circ}L$$

Dependence graph
over the previously decoded ranges.

For next range, recompute these sums only for domains that
all domain blocks.

sum of pixel intensities and sum of squared pixel intensities for
For range \( R' \), least squares approach requires computation of

Computation savings
Iteration (I) converges to $\mathcal{L}_f$.

where $\mathcal{L}_f$ is a Gauss-Seidel-like operator.

$$u_{n+1} = (0)f \cdot (\gamma f) \mathcal{L}_f = (I + \gamma)f$$

This corresponds to the iteration scheme:

In breadth-first traversal, pixel intensities are updated as soon as available.

Combination with pixel update.
Data Structures
\[ u^s = 32 \text{ and } u^o = 128. \]

\[ u^{D_4} = 16129 \text{ domain blocks of size } 8 \times 8. \]

\[ u^{D_3} = 15625 \text{ domain blocks of size } 16 \times 16. \]

\[ u^{D_2} = 14641 \text{ domain blocks of size } 32 \times 32. \]

\[ u^{D_1} = 12769 \text{ domain blocks of size } 64 \times 64. \]

Upper left pixel \((x, y)\) satisfies \(0 \equiv x \mod 4 \) and \(0 \equiv y \mod 4\).

Domain blocks: 8 \times 8 to 64 \times 64.

Quadtree partition: 4 \times 4 to 32 \times 32. Total of \(n_R = 2254\) ranges.

Test image: 8 bpp 512 \times 512 peppers.

Example
<table>
<thead>
<tr>
<th>Data Structures</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchanged domain blocks</td>
<td>59,164</td>
</tr>
<tr>
<td>intensity sums for range blocks</td>
<td>18,032</td>
</tr>
<tr>
<td>two temporary arrays for parents of $R^7$</td>
<td>1,156</td>
</tr>
<tr>
<td>breadth-first search</td>
<td>6,762</td>
</tr>
<tr>
<td>dependence graph</td>
<td>78,466</td>
</tr>
<tr>
<td>ordering of range blocks</td>
<td>22,540</td>
</tr>
<tr>
<td>two additional image arrays</td>
<td>52,428</td>
</tr>
</tbody>
</table>

Total: 710,588
Experimetal Results
<table>
<thead>
<tr>
<th></th>
<th>13.63</th>
<th>1943 (43.10%)</th>
<th>808</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11.52</td>
<td>489 (21.69%)</td>
<td>2254</td>
</tr>
<tr>
<td>Average number of processed range blocks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of range blocks with unchanged parameters</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Computation savings
Performance limit of fractal image compression remains an open question.

How far is our local optimum from the global one?

Local search algorithm is very fast.

Based on irregular partitions.

Local search is also useful for a state-of-the-art fractal scheme.

Visual quality is better in many cases.

Locally search improves collage coding by 0.2 to 0.8 dB.

Finding an optimal fractal image code is an intractable problem.

Conclusion
that increase the cost function.

Try strategies that escape local minima by accepting neighbors.

Select a better neighborhood.

Future work