Raouf Hamzaoui

Institut für Informatik
Universität Leipzig
Germany
http://www.informatik.uni-leipzig.de/cgip/

Outline

Principle of fractal image compression.

Collage coding.

Decoding complexity reduction.

Local search.

Efficient implementation of local search.

Conclusion.

Basic idea

$\mathcal{F}$ vector space of digital images

$f : \mathcal{I} = \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, N-1\} \to \mathbb{R}$.

Encoder: $f \mapsto T : \mathcal{F} \to \mathcal{F}$.

1) Description of T uses less bits than that of f.

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

3) $f_T \simeq f$.

Decoder: $T \mapsto f_T$.

$f_T = \lim_n f^{(n)}$ where $f^{(n+1)} = T(f^{(n)})$ and $f^{(0)}$ is an arbitrary initial image.

Fractal transform

A fractal transform $T : \mathcal{F} \to \mathcal{F}$ is characterized by:

a partition $\{R_1, \ldots, R_{n_R}\}$ of $\mathcal{I}$ into pairwise disjoint regions called range blocks,

$\mathcal{D} = \{D_1, \ldots, D_{n_D}\}$, $D_i \subset \mathcal{I}$, set of domain blocks,

$\mathcal{S} = \{s_1, \ldots, s_{n_s}\} \subset [-s_{\text{max}}, s_{\text{max}}]$, $s_{\text{max}} < 1$, set of real numbers called scaling factors,

$\mathcal{O} = \{o_1, \ldots, o_{n_o}\}$ set of real numbers called offsets.

Example

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

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

$D_1, \ldots, D_{n_D}$ are arbitrary $2^{n+1} \times 2^{n+1}$ squares.

Class of fractal transforms

To each tuple

$((D(1), s(1), o(1)), \ldots, (D(i), s(i), o(i)), \ldots, (D(n_R), s(n_R), o(n_R)))$

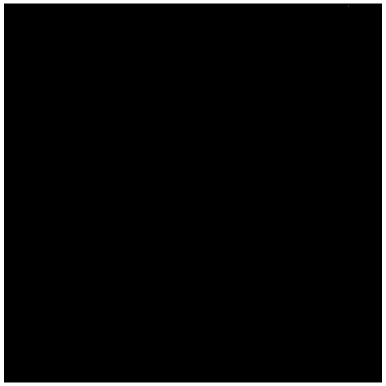
in $(\mathcal{D} \times \mathcal{S} \times \mathcal{O})^{n_R}$, we associate a fractal transform

$T : \mathcal{F} \to \mathcal{F}$ such that for $f \in \mathcal{F}$ and all $i \in \{1, \ldots, n_R\}$

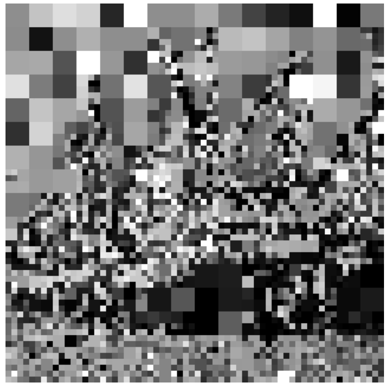
$$\mathbf{x}_{T(f)|R_i} = s(i) M \mathbf{x}_{f|D(i)} + o(i)\mathbf{1},$$

where M is a $2^{2n} \times 2^{2(n+1)}$ downsampling matrix.

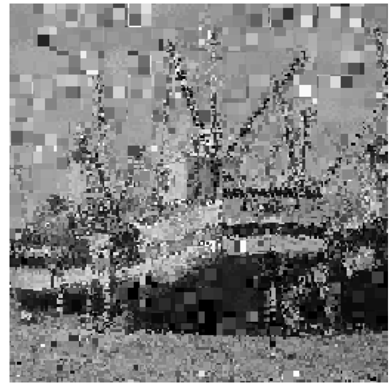
Decoding



(a)



(b)



(c)

(d)

(e)

(a) $f^{(0)}$,(b) $f^{(1)}$, (c) $f^{(2)}$, (d) $f^{(10)}$, (e) Original f.

Encoding problem

Given: f^* target image. $\mathcal{T}$ finite set of fractal transforms.

Goal: Find $T \in \mathcal{T}$ that solves the combinatorial problem

$$\min_{T \in \mathcal{T}} E(T) = \|f^* - f_T\|_2.$$

For a given partition $\rightarrow$ $(n_D n_s n_o)^{n_R}$ feasible solutions.

Enumeration is impractical for large n_R.

Collage coding

Greedy algorithm (collage coding) finds suboptimal solution.

Minimize $\|f^* - T(f^*)\|_2$ instead of $\|f^* - f_T\|_2$.

Motivation for collage coding:

$$\|f^* - f_T\| \leq \frac{1}{1 - s(T)}\|f^* - T(f^*)\|,$$

where T is a contraction for the norm, and $s(T)$ is the contraction factor of T.

$$\begin{aligned}\|f^* - T(f^*)\|_2^2 &= \sum_{i=1}^{n_R} \|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{T(f^*)|R_i}\|_2^2 \\ &= \sum_{i=1}^{n_R} \|\mathbf{x}_{f^*|R_i} - (s(i)M\mathbf{x}_{f^*|D(i)} + o(i)\mathbf{1})\|_2^2.\end{aligned}$$

Optimal parameters in collage coding are solutions of the n_R independent minimization problems

$$\min_{(D(i),s(i),o(i))\in\mathcal{D}\times\mathcal{S}\times\mathcal{O}} \|\mathbf{x}_{f^*|R_i} - (s(i)M\mathbf{x}_{f^*|D(i)} + o(i)\mathbf{1})\|_2^2.$$

$f|R_i$ and $f|D(i)$ should be “similar”.

Solving collage coding with least squares

Let $D_p \in \mathcal{D}$. Denote $M\mathbf{x}_{f^*|D_p}$ by $\mathbf{x}_p$.

Let s and o denote the solutions of the least squares problem

$$\min_{s,o\in\mathbb{R}} \|\mathbf{x}_{f^*|R_i} - (s\mathbf{x}_p + o\mathbf{1})\|_2^2.$$

Quantize s and o in $\mathcal{S}$ and $\mathcal{O}$, yielding a scaling factor s^* and an offset o^*.

Find a domain $D(i)$ that minimizes the error

$$\|\mathbf{x}_{f^*|R_i} - (s^*\mathbf{x}_p + o^*\mathbf{1})\|_2^2.$$

Decoding complexity reduction

Let T be a fractal transform and $f_T = \lim_k f^{(k)}$ where $f^{(k+1)} = T(f^{(k)})$ and $f^{(0)}$ is an arbitrary initial image.

Suppose that

$$f^{(k+1)}(i,j) = \frac{s}{4}(f^{(k)}(l,m)+f^{(k)}(l+1,m)+f^{(k)}(l,m+1)+f^{(k)}(l+1,m+1))+o$$

where s is a scaling factor and o is an offset.

Suppose that the intensities of pixels $(l,m),(l+1,m),(l,m+1),(l+1,m+1)$ are computed before the pixel intensity of (i,j).

We can compute

$$\begin{aligned} f^{(k+1)}(i,j) &= \frac{s}{4}(f^{(k+1)}(l,m) + f^{(k+1)}(l+1,m) + f^{(k+1)}(l,m+1) \\ &\quad + f^{(k+1)}(l+1,m+1)) + o. \end{aligned}$$

We need only one image array in the decoding.

Using an ordering of the pixels, let $\mathbf{x}^{(k)}$ be the vector corresponding to image iterate $f^{(k)}$.

Then there exists a matrix $A = (a_{u,v})$ and a vector $\mathbf{b} = (b_u)$ such that for $u = 1, \ldots, N^2$

$$x_u^{(k+1)} = \sum_{v=1}^{N^2} a_{u,v} x_v^{(k)} + b_u.$$

New method:

For $u = 1, \ldots, N^2$

$$x_u^{(k+1)} = \sum_{v \leq u-1} a_{u,v} x_v^{(k+1)} + \sum_{v \geq u} a_{u,v} x_v^{(k)} + b_u$$

Let $A = L + U$ where

$$l_{u,v} = \begin{cases} 0, & \text{if } v \geq u; \\ a_{u,v}, & \text{otherwise.} \end{cases}$$

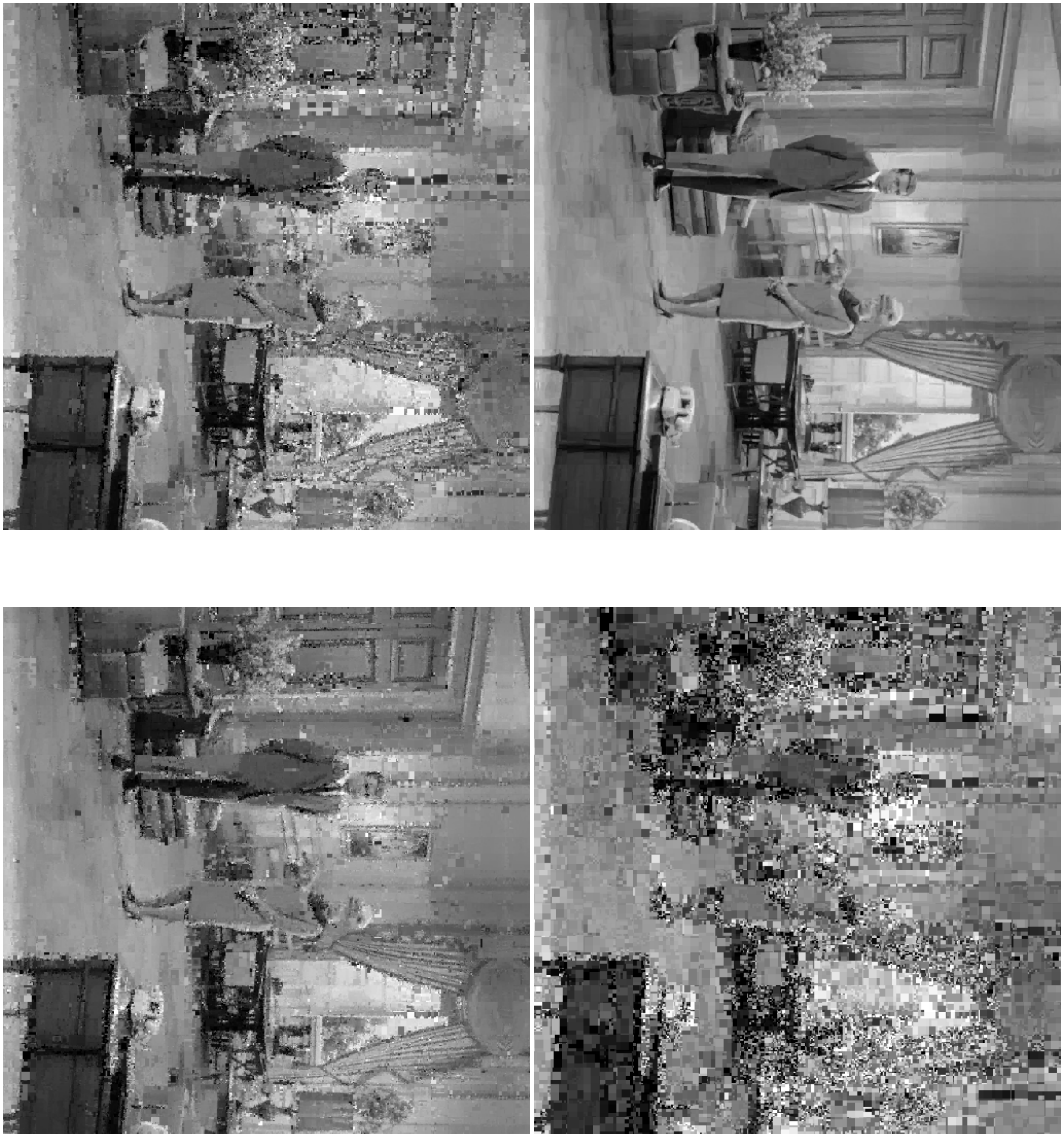
Then the proposed decoding corresponds to the iterative method

$$\mathbf{x}^{(k+1)} = (I - L)^{-1} U \mathbf{x}^{(k)} + (I - L)^{-1} \mathbf{b}.$$

Proposed decoding converges to the same fixed point as conventional decoding [Hamz97].

Convergence is faster.

If all scaling factors have the same sign, then the asymptotic rate of convergence of the new method is greater than or equal to that of the conventional method [Hamz97].



Improvement of collage coding

Start from the solution found by collage coding.

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

After quantization, the PSNR improvement over collage coding is insignificant.

Barthel [Barthel94] and Lu [Lu97] suggested to update all parameters.

Start from an original solution T_0 found by collage coding.

Construct a sequence of fractal transforms $T_1, T_2, \ldots,$ where at each step n, new parameters are found by solving the minimization problem

$$\min_{(D(i),s(i),o(i))\in\mathcal{D}\times\mathcal{S}\times\mathcal{O}} \|\mathbf{x}_{f^*|R_i} - (s(i)M\mathbf{x}_{f_{T_{n-1}}|D(i)} + o(i)\mathbf{1})\|^2$$

for all range blocks $R_i,\ i = 1, 2, \ldots, n_R$

Substantial PSNR improvements over collage coding.

No guarantee that the reconstruction error decreases after each step.

The procedure is time expensive because every step corresponds to a new encoding of the test image.

Combinatorial optimization

Given: f^* target image. $\mathcal{T}$ set of $(n_D n_s n_o)^{n_R}$ fractal transforms.

$$\min_{T \in \mathcal{T}} E(T) = \|f^* - f_T\|_2.$$

Problem is NP-hard [RuHa97].

Local search

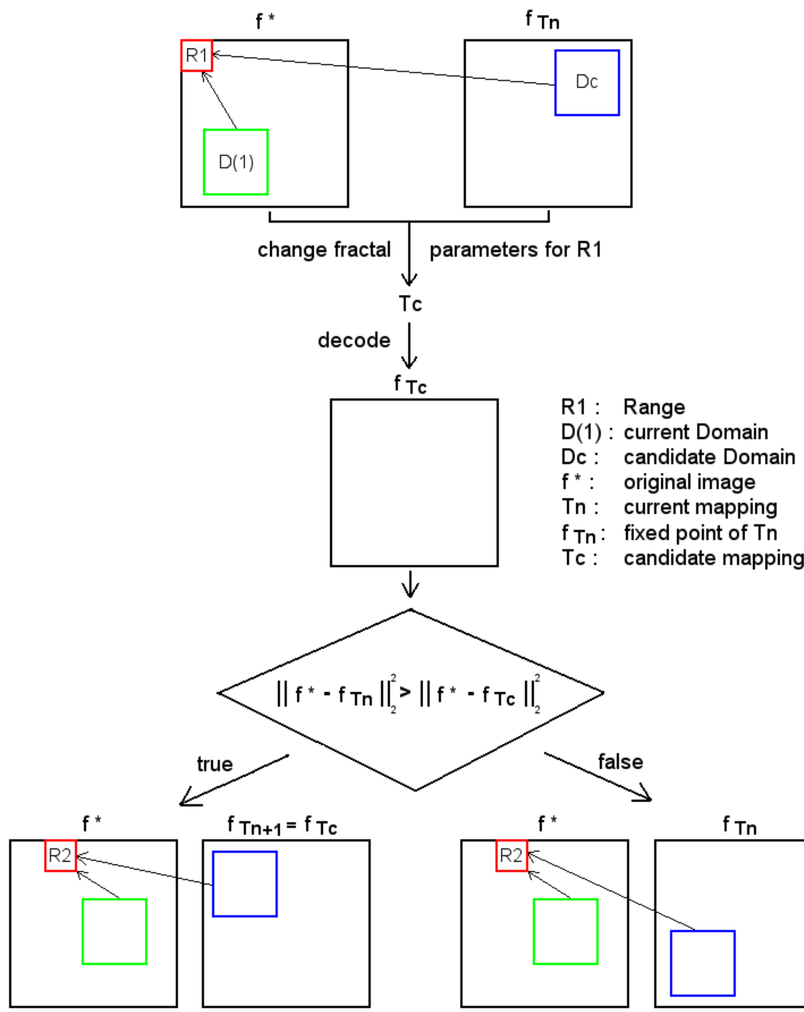
Define a neighborhood of a solution.

Start at some initial solution.

Search for a better solution in its neighborhood.

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

Local search algorithm



Performance

Improvement over collage coding can be up to 0.8 dB for standard 512×512 images.

Left: collage coding. Right: local search

Image	n_R	Compression ratio	Collage PSNR	Lu PSNR	Proposed PSNR
256×256 Lenna	1024	20.45:1	26.51	26.63	26.77
256×256 San Fran	1024	20.45:1	24.54	24.65	24.92
512×512 Boat	4096	18.96:1	29.74	29.87	30.01

Time complexity

Local search requires successive computations of the fixed points.

The candidate transformation is found by modifying the parameters of only one range.

Complexity reduction

Start the iterations from the current fixed point.

Use the pixel update decoding.

No computation of the fixed point if parameters are unchanged.

Sort ranges according to decreasing error $\|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{f_{T_0}|R_i}\|_2^2$.

Use the dependence graph.

Dependence graph

$f_{T_n} \to T_c(f_{T_n})$: only the pixel intensities in R_i have to be updated.

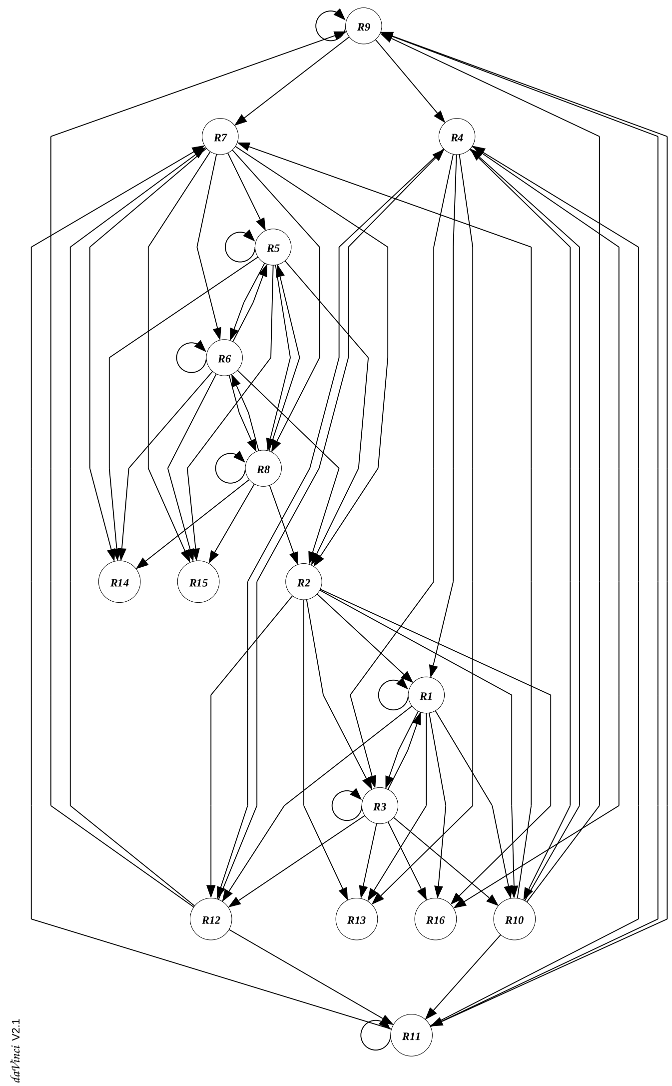
$T_c(f_{T_n}) \to T_c(T_c(f_{T_n}))$: only ranges whose domains overlap R_i have to be updated.

A range R_j is called a child of a range R_i if the domain that encodes R_j overlaps R_i.

A dependence graph of a code is a directed graph where vertices are the ranges and $R_i \to R_j$ if R_j is a child of R_i.

$f_{T_n} \to T_c(f_{T_n}) \to T_c(T_c(f_{T_n})) \to \cdots \to f_{T_c}$ can be implemented as a breadth-first traversal of the dependence graph of the code of T_c, starting from vertex R_i.

Example



Computation savings

For range R_i, least squares approach requires computation of sum of pixel intensities and sum of squared pixel intensities for all domain blocks.

For next range, recompute these sums only for domains that overlap the previously decoded ranges.

Combination with pixel update

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

This corresponds to the iteration scheme

$$f^{(k+1)} = T_{c,i_k}(f^{(k)}),\ f^{(0)} = f_{T_n}, \tag{1}$$

where T_{c,i_k} is a Gauss-Seidel like operator.

Iteration (1) converges to f_{T_c}.

Data structures

For each range a linked list is kept, which holds all its children.

When domain that encodes range R_i is changed, update the dependence graph:

1. Determine the current parents of vertex R_i.
2. Remove R_i from the linked lists of parents.
3. Determine the new parents of R_i.
4. Insert R_i in the linked list of each new parent.

Example

Test image: 8 bpp 512×512 Peppers.

Quadtree partition: 4×4 to 32×32. Total of $n_R = 2254$ ranges.

Domain blocks: 8×8 to 64×64.

Upper left pixel (i, j) satisfies $i \equiv 0 \pmod 4$ and $j \equiv 0 \pmod 4$.

$n_{D_1} = 12769$ domain blocks of size 64×64.

$n_{D_2} = 14641$ domain blocks of size 32×32.

$n_{D_3} = 15625$ domain blocks of size 16×16.

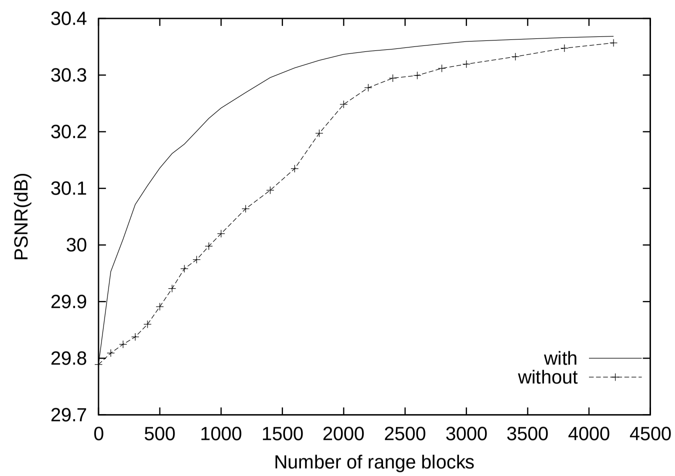
$n_{D_4} = 16129$ domain blocks of size 8×8.

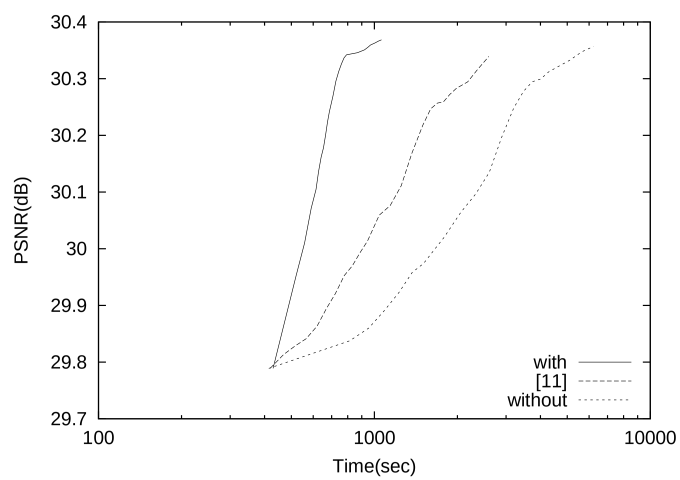
$n_s = 32$ and $n_o = 128$.

Memory requirements

Data structures	Bytes
two additional image arrays	524,288
ordering of range blocks	22,540
dependence graph	78,646
breadth-first search	6,762
two temporary arrays for parents of R_r	1,156
intensity sums for range blocks	18,032
unchanged domain blocks	59,164
Total	710,588

Experimental Results





th: best implementation, [11]: pixel update only, without: brute-force.

Computation savings

Number of range blocks	Number of range blocks with unchanged parameters	Average number of processed range blocks
$n_R = 2254$	489 (21.69 %)	11.52
$2n_R = 4508$	1943 (43.10 %)	13.63

Conclusion

Finding an optimal fractal image code is an intractable problem.

Local search improves collage coding by 0.2 to 0.8 dB.

Visual quality is better in many cases.

Local search is also useful for a state-of-the-art fractal scheme based on irregular partitions.

Local search algorithm is very fast.

How far is our local optimum from the global one?

Performance limit of fractal image compression remains an open question.

Future work

Select a better neighborhood.

Try strategies that escape local minima by accepting neighbors that increase the cost function.