Visual Quality Increasing using Image Parts Improvement for Fractal Image Compression

Dmitriy Vatolin

Moscow State University

dm@amc.ru

Introduction

New compression methods with loosing of information:

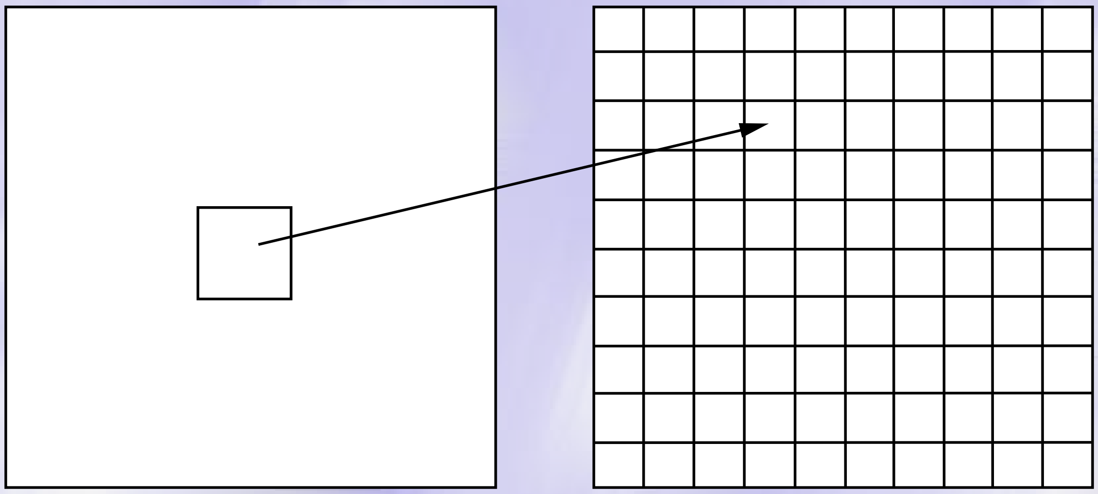
- JPEG based methods
- Wavelet compression
- Fractal image compression

Fractal image compression — method with lost of information, that appear in 1992.

Idea of the method (1)

Compression method — searching for similar regions with other size.

Idea of the method (2)



$$w_i(\tilde{\bar{o}}) = w_i\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & p \end{pmatrix} \bullet \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e \\ f \\ q \end{pmatrix}$$

Affine transform is used for transformation .

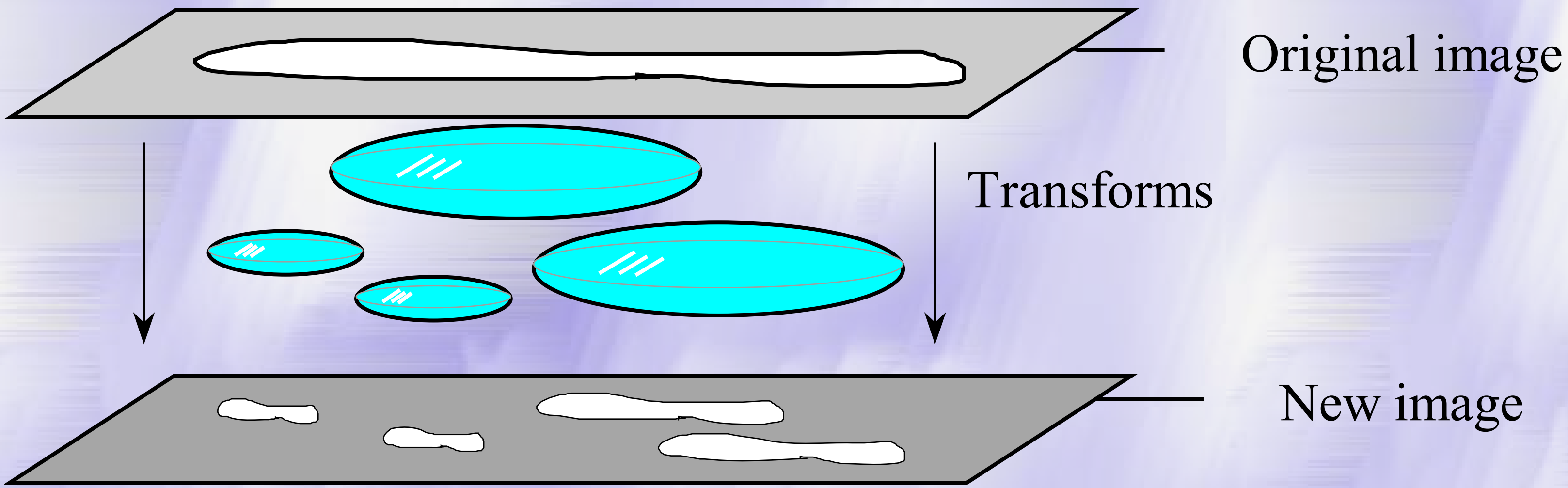
Idea of the method (3)

$w_i(\bar{o})$ — Iterated Function System (IFS)

- Collage Theorem theory tells that using only coefficients we can get the original image.
- Coefficients are saved to file
- If the size (in bytes) of coefficients is smaller than the size of the original image — we have a compression algorithm

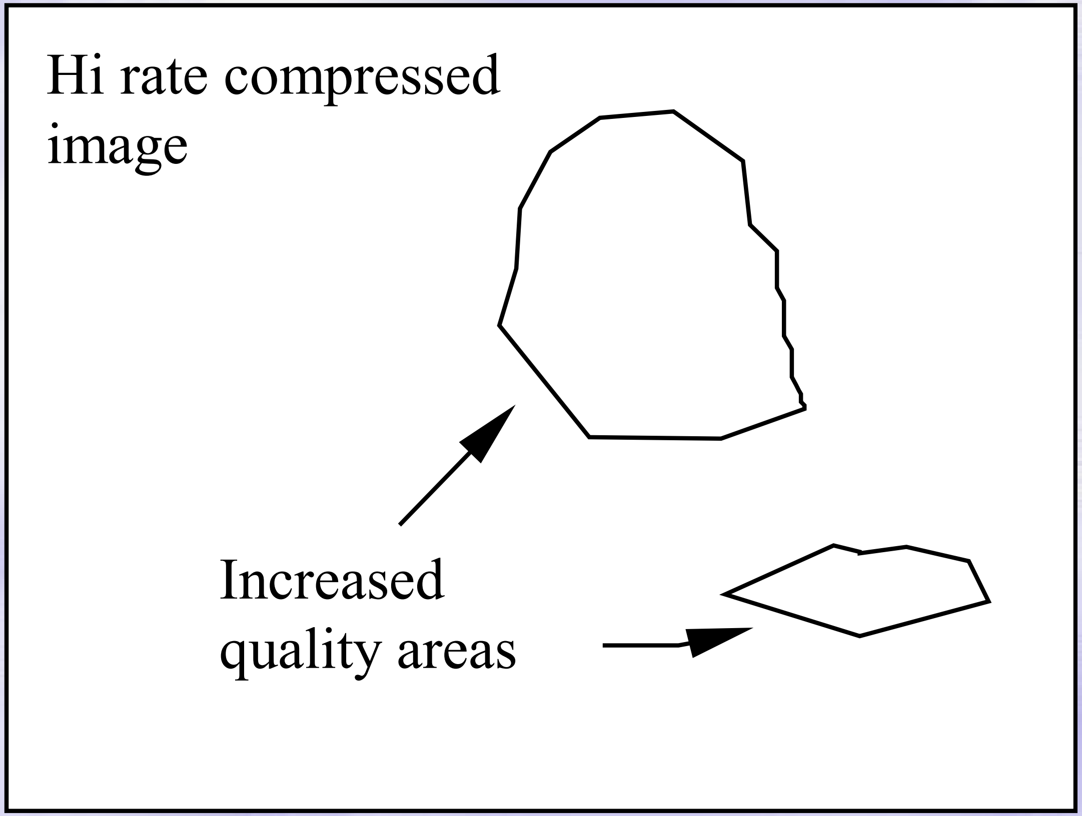
Decompression

One iteration:

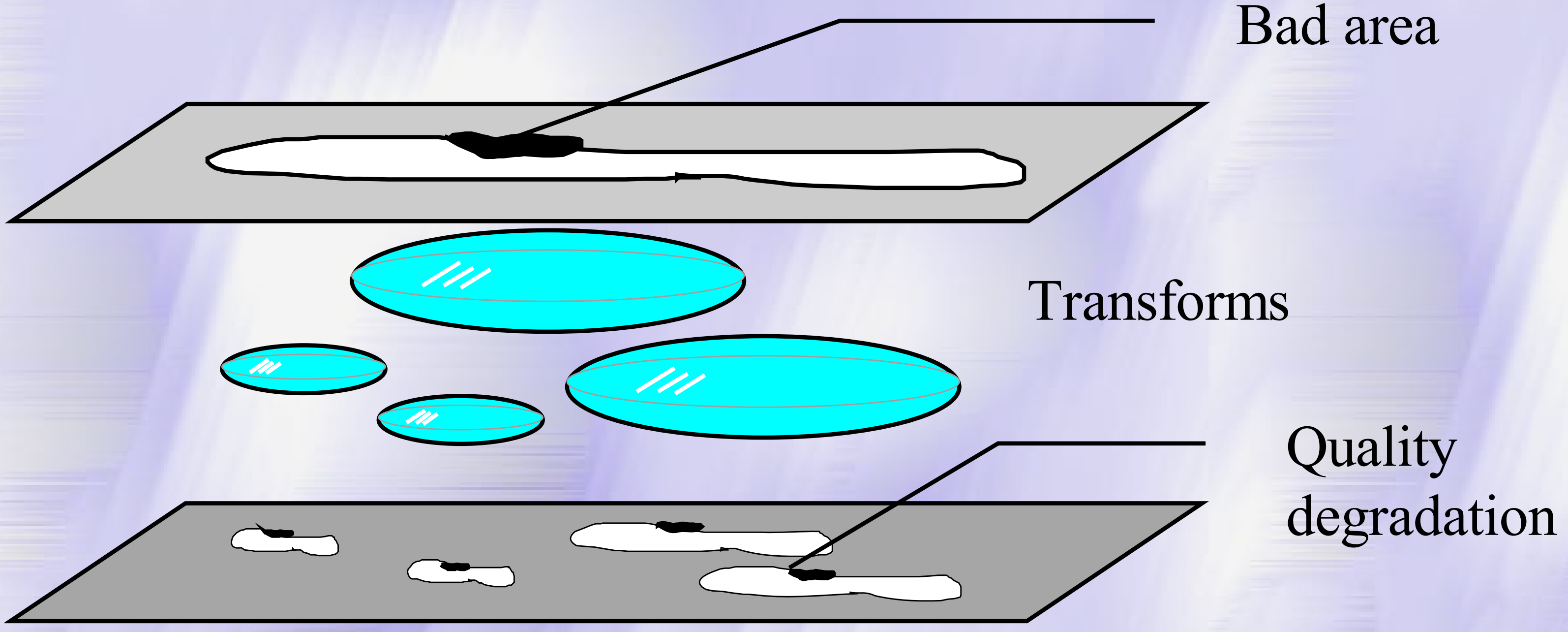


Selecting method: Idea

Main idea: Increase quality only in selected areas.



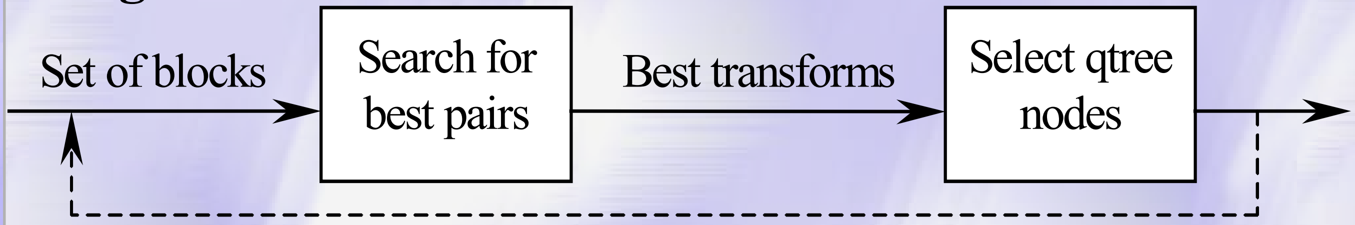
Selecting method: Problem



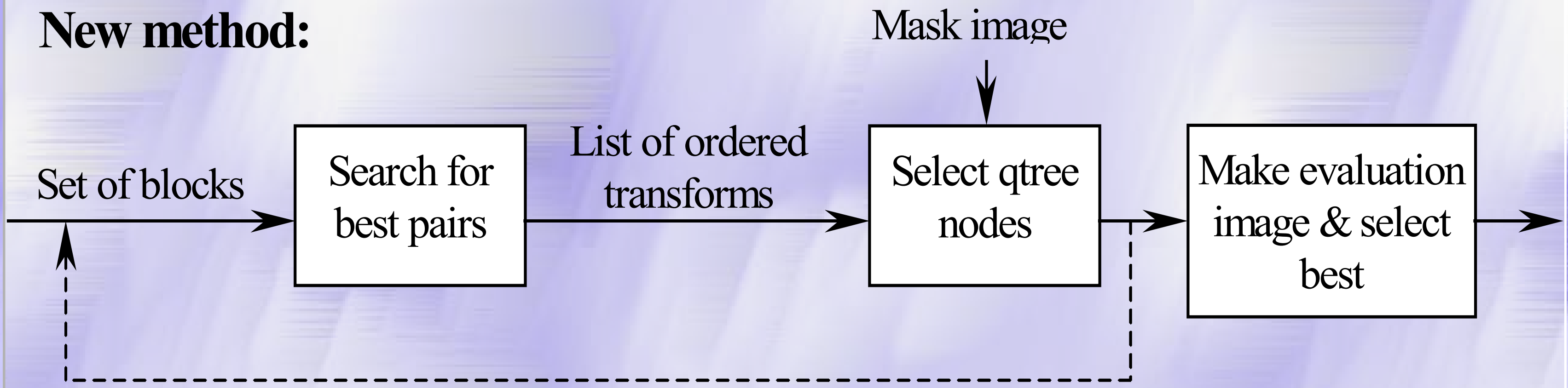
Quality degradation on the decompression stage

Selecting method: In short

Original method:



New method:



Original method

Step 1. Pre-calculation

```
for (all range blocks) {
    dist_ij = MaximumDistance;
    R_ij = image->CopyBlock(i,j);
}
```

Step 2. Searching for a best blocks

```
for (all range blocks) {
    for (all domain blocks) {
        current= current coordinates;
        D=image->CopyBlock(current);
        current_dist=R_ij.L2dist (D);
        if(current_dist < dist_ij) {
            dist_ij = current_dist;
            best_ij = current;
        }
    }
}
Save_All_Best_To_Fractal_File(best);
```

Selected method: Compression

Changes in the algorithm:

Step 2. Searching for a best blocks

```
for (all range blocks) {
   for (all domain blocks) {
        current= current coordinates;
        D=image->CopyBlock(current);
        current_dist=Rij.L2dist (D);
        Insert(bestij, current, current_dist);
        }
   }
}
// Prepare evaluation image.
for (all bestij) {
   Fill in evaluation image block with brightness,
   depends on best distance of this block
}
// Other levels of qtree
for (all bestij) {
   for (all transformations in bestij)
        select best transformation
}
Save_All_Best_To_Fractal_File(best);
```

Selected method: Decompression

Decompression does not changed in this approach!

Original images

Original method

Mask image

Decompressed images

Original method

New method

Creation of evaluation images

Two ways:

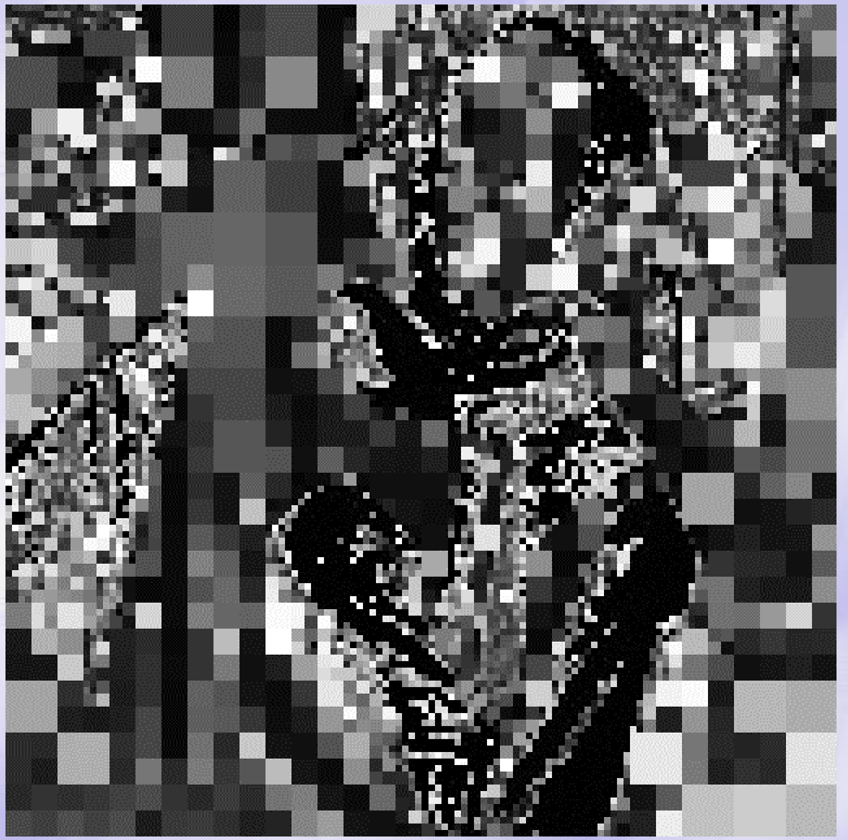
1) Faster:

Use block L_2 distances for average block as accuracy.

2) Better quality:

Use full image decompression and pixel differences.

Evaluation images (distance)

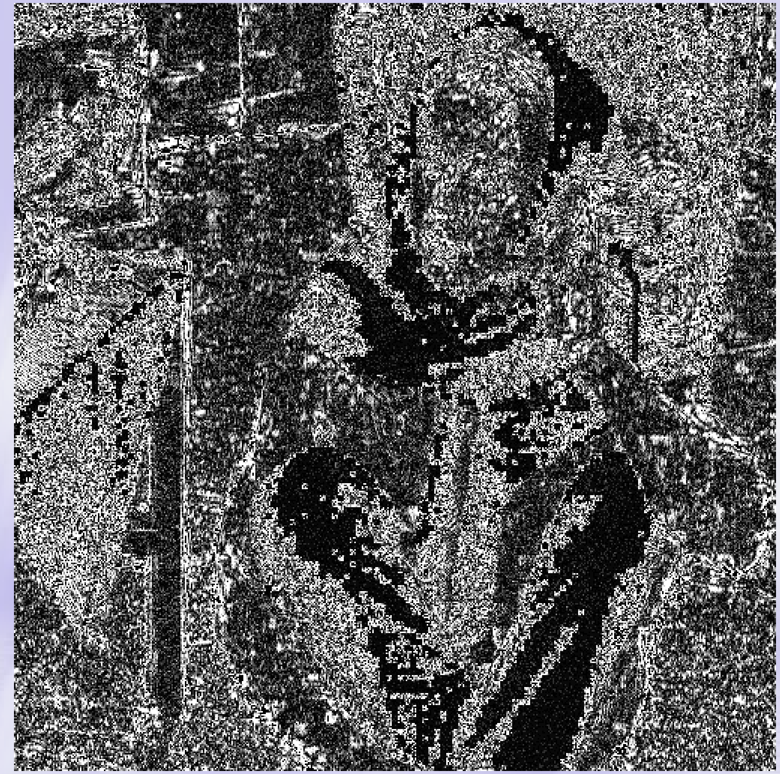


Original method

New method

Evaluation images (differences)

Original method



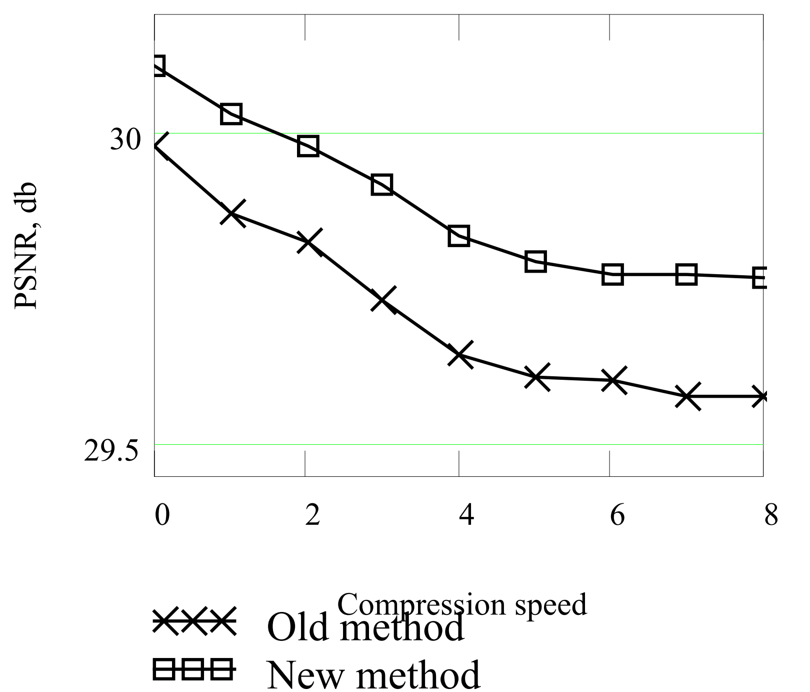
Our method

Zoomed images

Original method

Our method

Results: Quality increasing (1)



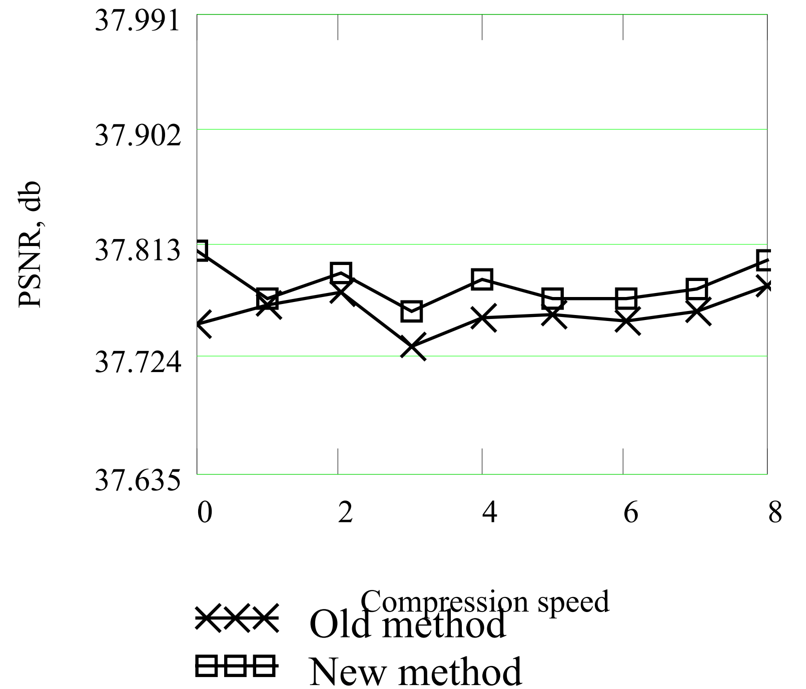


Image quality increasing for Barbara & Dmc images.

Results: Quality increasing (2)

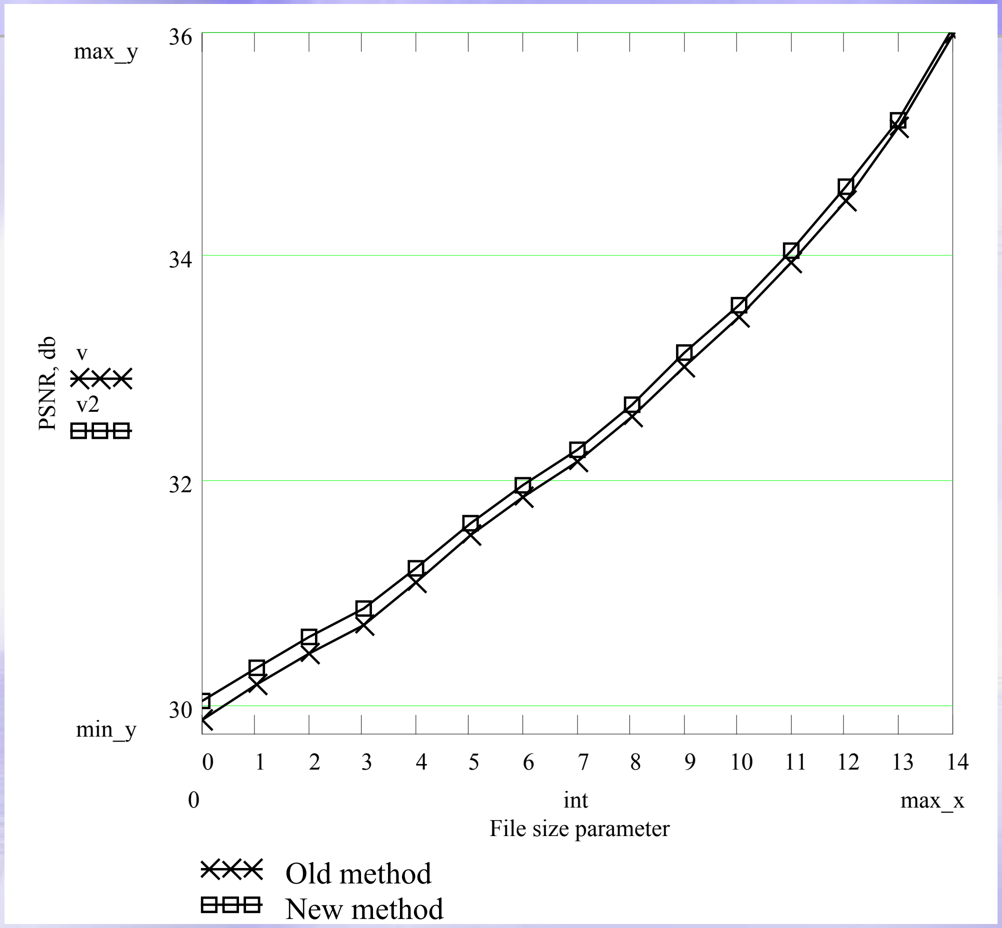
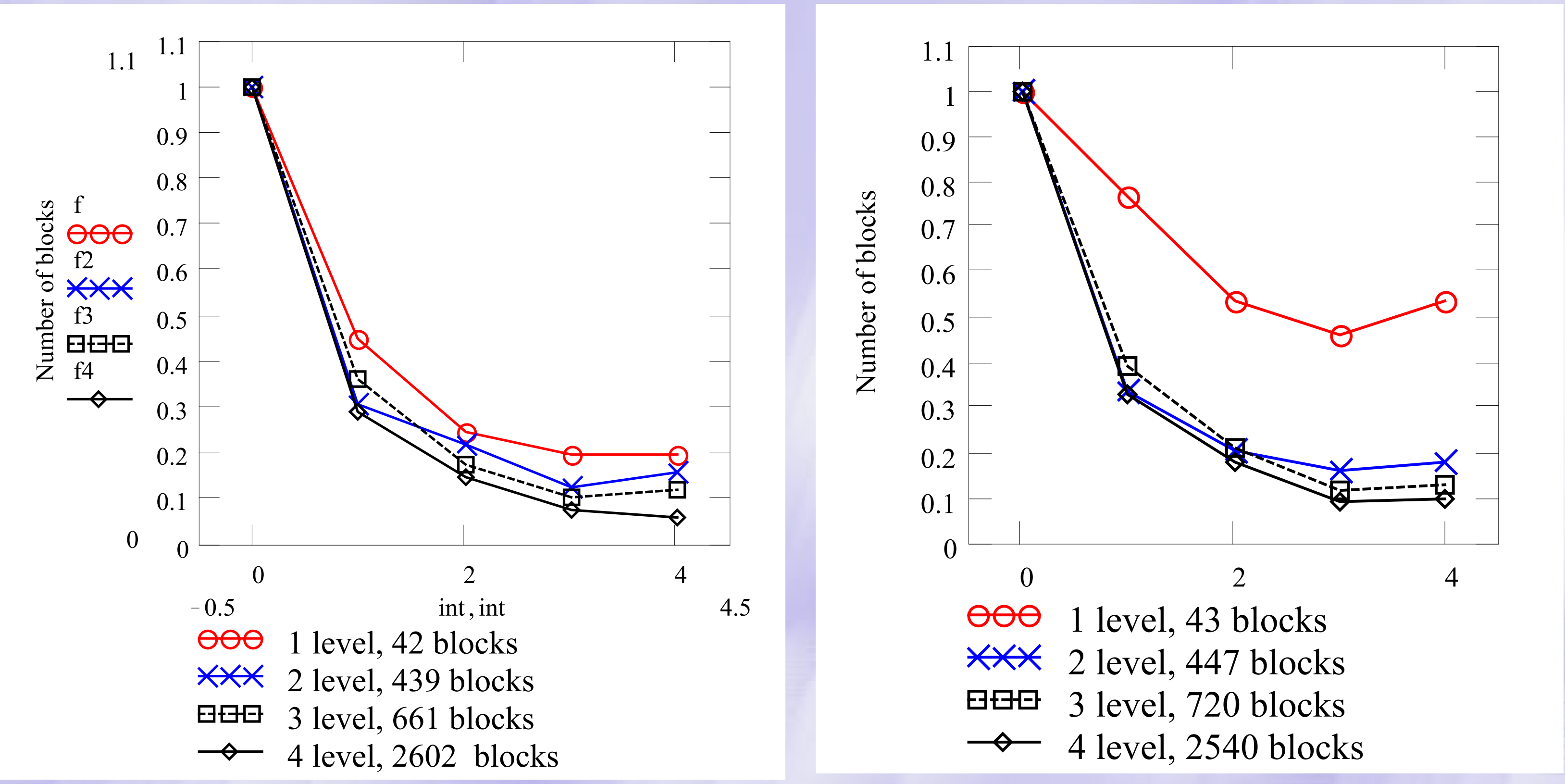


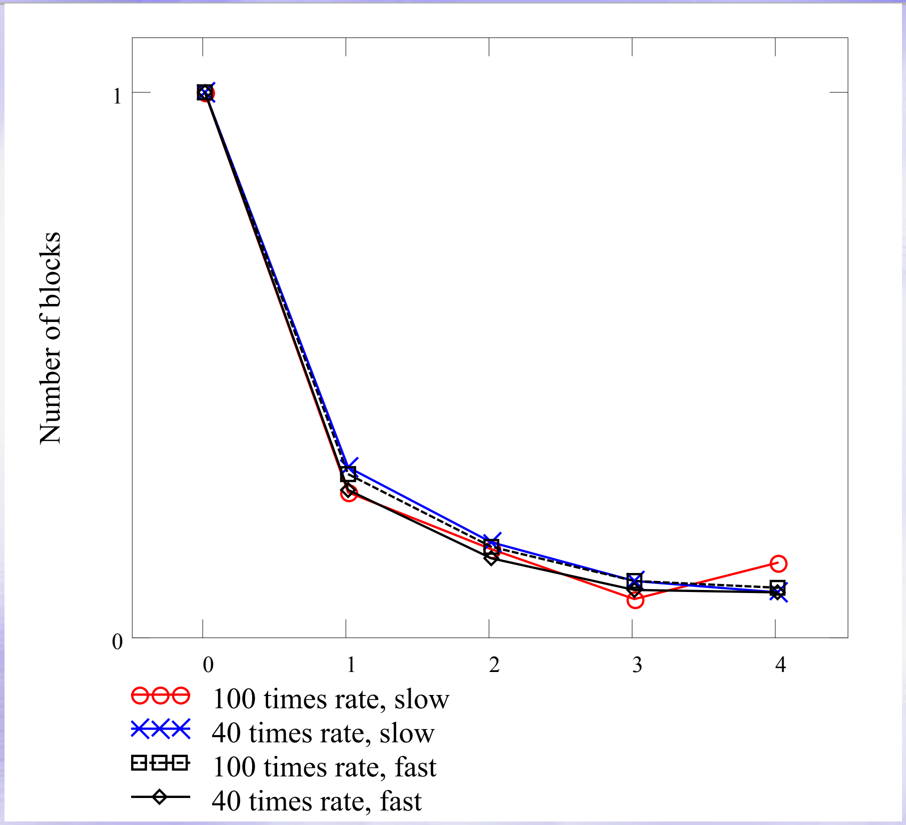
Image quality increasing for Barbara from compression rate

Results: Block selecting (1)



Frequency distribution of block number in ordered lists for fast and slow compression of Barbara image (7 times compression rate)

Results: Block selecting (2)



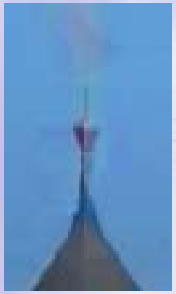
Frequency distribution of block number in ordered lists for fast and slow compression of Dmc image (40 & 100 times compression rate)

Example (1)

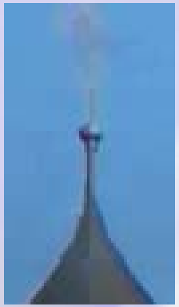
Very bad area for compression -- non sharp but important area

Peresl.bmp 320x512
491576 bytes

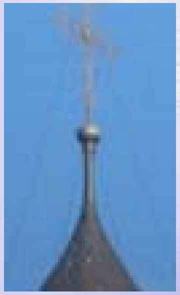
40 times compression
using areas



40 times old compression



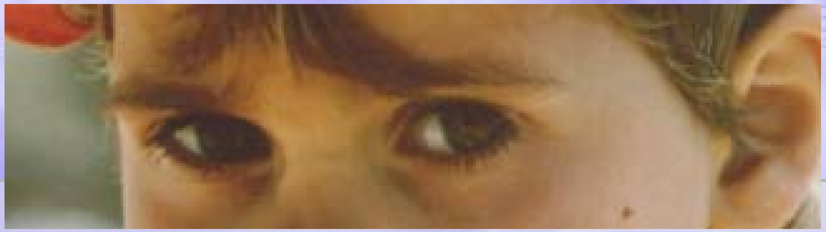
40 times compression using areas



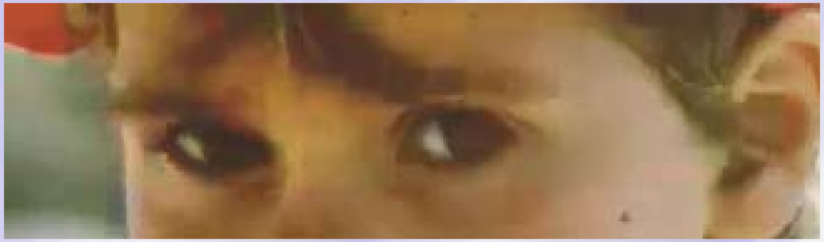
Original

Example (2)

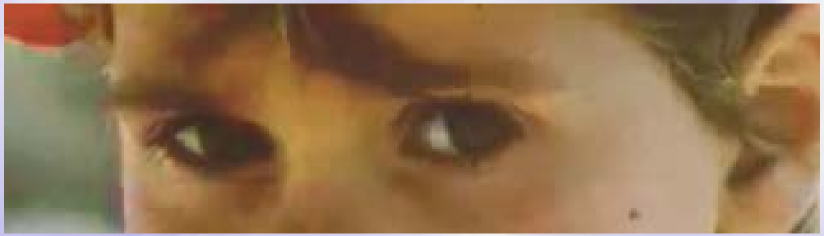
Florian.bmp 384x608
700472 bytes



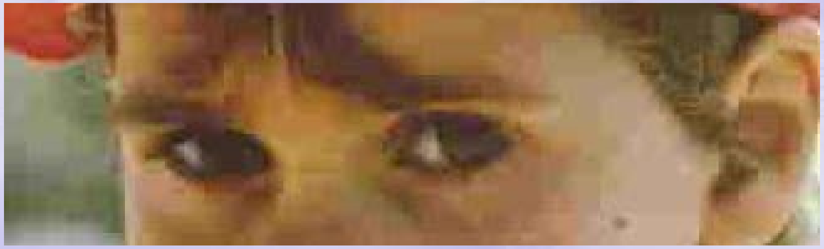
Original



63 times compression old

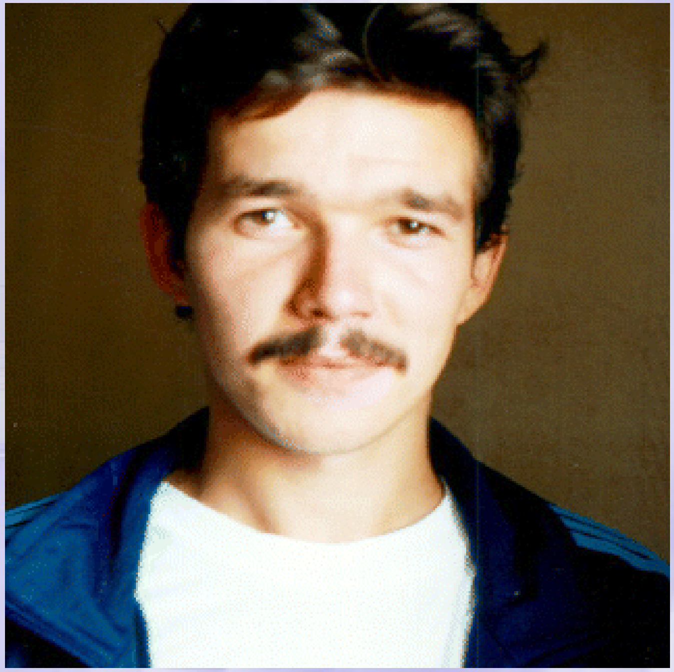


63 times compression using areas

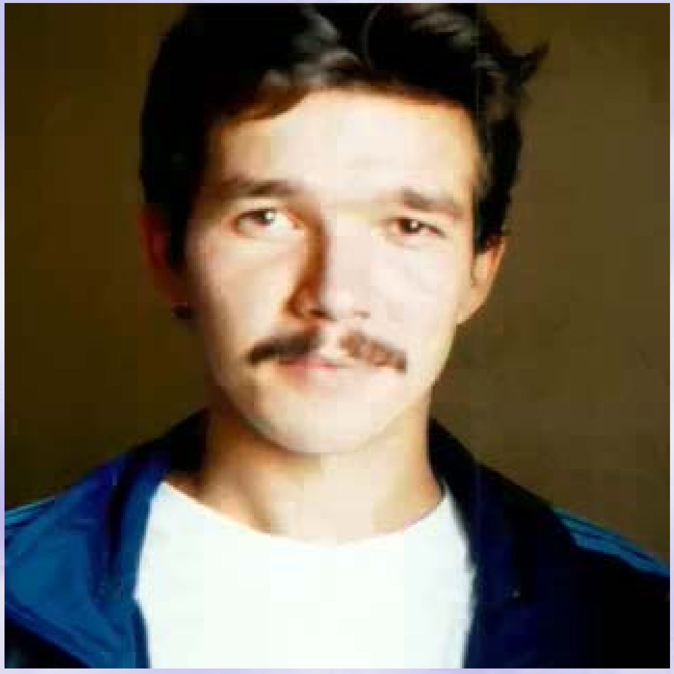


65 times compression JPEG

Example: Color photo dm (1)



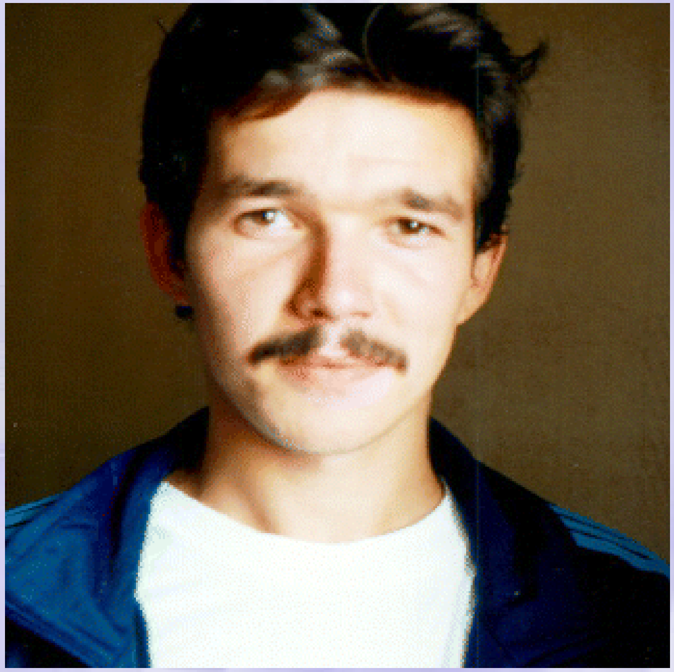
Original image (307Kb)



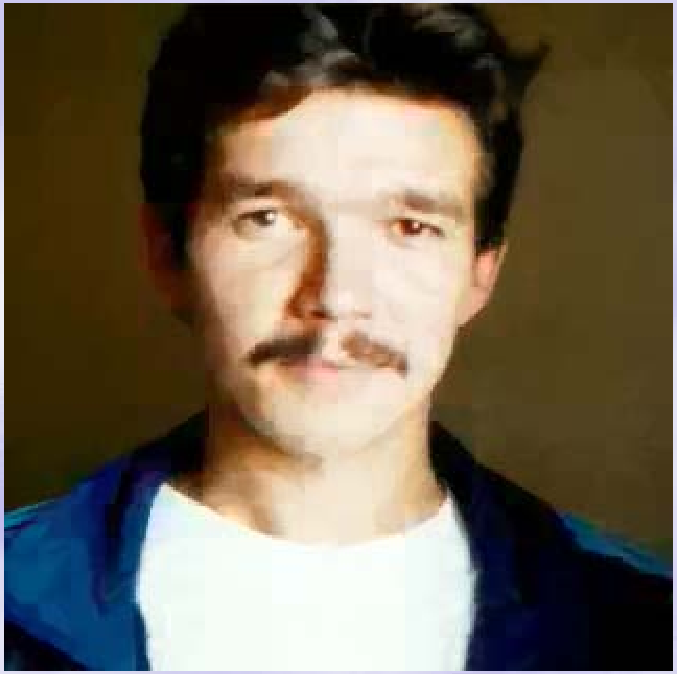
Rate 40 times (7.67)

Used 320x320 True Color (24 bits per pixel) image

Example: Color photo dm (2)



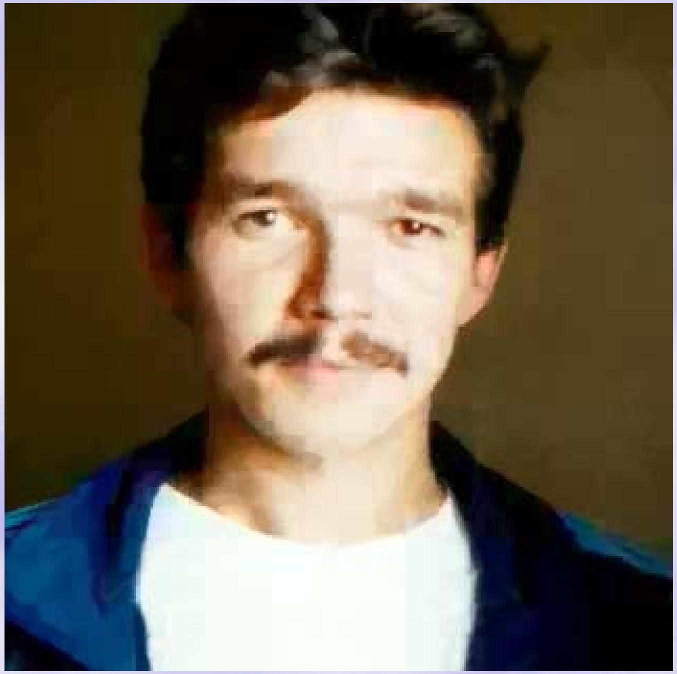
Original image (307Kb)



Rate 100 times (3.02Kb)

Compare with JPEG

JPEG: Rate 100 times (3.06Kb)



FIF: Rate 100 times (3.02Kb)

Conclusion & Future Work

- We increase image quality (without mask!)
- We increase average compression rate

Future work:

- Automatic mask construction
- Iterated block selection

References

[1] *A.E. Jacquin*. Image coding based on a fractal theory of iterated contractive image transformations. // IEEE Trans. Image Processing 1 18-30, 1992;

[2] *Y. Fisher*. Fractal image compression with quadtrees. In: Fractal Image Compression — Theory and Application to Digital Images. Y. Fisher (ed.). Springer-Verlag, New York 1994;

[3] E. Reusens, Partitioning complexity issue for Iterated Function Systems based image coding, in Proc. of VII European Signal Processing Conference, Vol. 1, Edinburg, U.K., September 1994, pp. 171-174

[4] *Y. Fisher, S. Menlove*, Fractal encoding with HV partitions, in [2];

[5] *F. Davoine, M. Antonini, J.-M. Chassery, M. Barlaud,* Fractal image compression based on Delaunay triangulation and vector quantization, in Proc. of IEEE Transaction on Image Procession, Vol. 5, No. 2, February 1996;

[6] *F. Davoine, J. Svensson, J.-M. Chassery,* A mixed triangular and quadrilateral partition for fractal image coding, in Proc. of IEEE International Conference on Image Processing, Washington, D.C, 1995;

[7] *L. Thomas, F. Deravi,* **Region-based fractal image compression using heuristic search**, in Proc. of IEEE Transaction on Image Procession, Vol. 4, No. 6, June 1995, pp. 832-838;

[8] *D. Saupe, H. Hartenstein,* Evolutionary fractal image compression, in Proc. of IEEE International Conference on Image Processing, Laussane, September 1996, pp. 129-132;

[9] *M. Ruhl, H. Hartenstein, D. Saupe,* **Adaptive partitioning for fractal image compression**, in Proc. of IEEE International Conference on Image Processing, Santa Barbara, October 1997;

[10] *D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, D. Marini,* **Optimal hierarchical partitions for fractal image compression**, in Proc. of IEEE International Conference on Image Processing, Chicago, October 1998;

[11] *Д.С. Ватолин*. Гибридная схема фрактальной компрессии и квантования векторов для малых блоков. // Материалы Graphicon-98, 1998;

[12] *Д.С. Ватолин.* Использование ДКП для ускорения фрактального сжатия изображений. // Программирование, N 4 1999.

Сравнение с другими алгоритмами (1)

Алгоритм	За счет чего происходит сжатие
RLE	2 2 2 2 2 2 2 15 15 15 — Подряд идущие цвета
LZW	2 3 15 40 2 3 15 40 — Одинаковые подцепочки
Хаффмана	2 2 3 2 2 4 3 2 2 2 4 — Разная частота появления цвета
Wavelet	Плавные переходы цветов и отсутствие резких границ
JPEG	Отсутствие резких границ
Фрактальный	Подобие между элементами изображения

Сравнение с другими алгоритмами (2)

Алгоритм	К-ты сжатия	Сим-метрич-ность	На что ориенти-рован	Поте-ри	Раз-мер-ность
RLE	1/32 1/2 2/1	1	3-8 bit	Нет	1D
LZW	1/1000 1/4 7/5	1.2-3	1-8 bit	Нет	1D
Haffman	1/8 2/3 1/1	1-1.5	5-8 bit	Нет	1D
JBIG	1.5	~1	1-bit.	Нет	2D
Lossless JPEG	2	~1	24-bit greyscale	Нет	2D
JPEG	2-20	~1	24-bits greyscale	Да	2D
Wavelet	2-200	1.5	24-bits greyscale	Да	2D
Fractal	2-2000	1000-10000	24-bits greyscale	Да	2D