The hybrid method of a fractal image compression and vector quantization

Dmitriy Vatolin

Moscow State University

dm@amc.ru

Introduction

New compression methods may be:

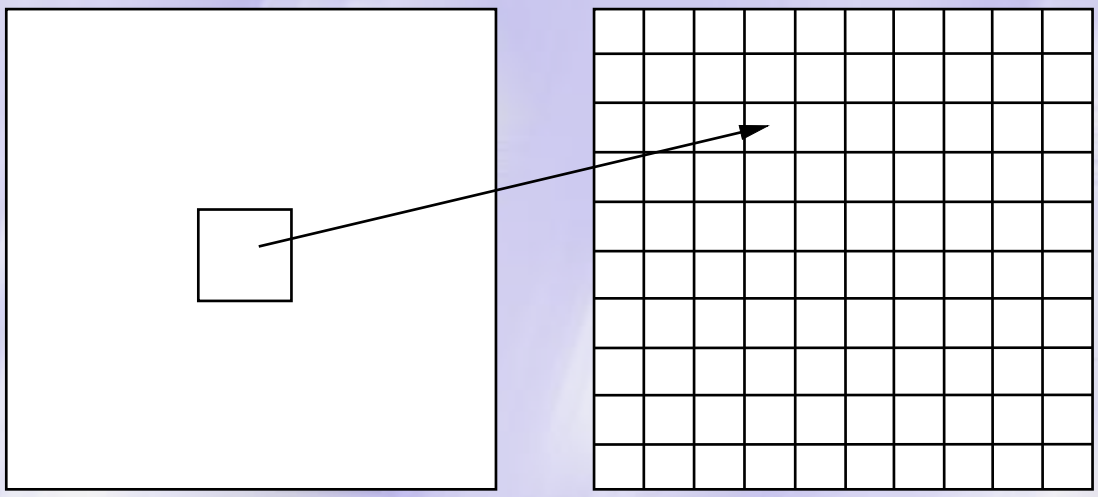
- With lost of information
- Without lost of information

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)



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

Affine transform is used for transformation .

Idea of the method (3)

$w_i(\tilde{o})$ — 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

Mathematical Background

$$[\min] R = \sum_{i=1}^{n} (s \times a_i + o - b_i)$$

Explicit statement for s and o:

$$s = \left[n^2 (\sum_{i=1}^{n} a_i b_i) - (\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} b_i) \right] / \left[n^2 \sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} b_i)^2 \right]$$

$$o = \left[\sum_{i=1}^{n} b_i - s \cdot \sum_{i=1}^{n} a_i \right] / n^2$$

$$R = \left[\sum_{i=1}^{n} b_i^2 + s \cdot (s \sum_{i=1}^{n} a_i^2 - 2(\sum_{i=1}^{n} a_i b_i) + 2o \sum_{i=1}^{n} a_i) + o(on^2 - 2 \sum_{i=1}^{n} b_i) \right] / n^2$$

Original method

Step 1. Pre-calculation

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

Step 2. Searching for a best blocks

```
for (all domain blocks) {
    for (all range blocks) {
          current= current coordinates;
          R=image->CopyBlock(current);
          current_dist=D_ij.L2dist (R);
          if(current_dist < dist_ij) {
                    dist_ij = current_dist;
                    best = current;
          }
          Save_Coefficients(best);
    }
}
```

Hybrid method: Idea

Main idea:

On the **last** level of qtree fractal affine transform was changed to **Vector Quantization** block.

Hybrid method: Compression

Changes in the algorithm:

Step 2. Searching for a best blocks

```
for (all domain blocks) {
    for (all range blocks) {
          current= current coordinates;
          R=image->CopyBlock(current);
          current_dist=Dij.L2dist (R);
          if(current_dist < distij) {
                    distij = current_dist;
                    best = current;
          }
          Save_Coefficients(best);
    }
}
```

This code changed to:

```
R=codebook->CopyBlock(current_number);
current_dist = Dij.L2_2x2dist(R);
```

Hybrid method: Decompression

Changes in the algorithm:

```
for (all iterations) {
    for (all domain blocks) {
        image1->CopyBlock(image2,coordinates);
    }
    image2=image1;
}
Save_Image(image1);
```

This code changed to:

```
if(!block_is_VQ)
        image1->CopyBlock(image2,coordinates);
else
        image1->MakeBlockFromCodebook(number);
```

Indexing for VQ blocks

C-code for index calculating:

```
index=((p[0] > p[1])?0: 1) |
      ((p[2] > p[3])?0: 2) |
      ((p[0] > p[2])?0: 4) |
      ((p[1] > p[3])?0: 8) |
      ((p[0] > p[3])?0:16) |
      ((p[1] > p[2])?0:32);
```

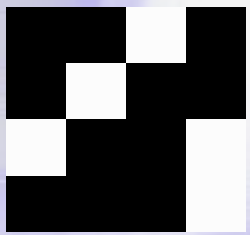
This code:

- Effective
- Simple
- Faster then DCT index for original algorithm

Changes: Quality & Scaling

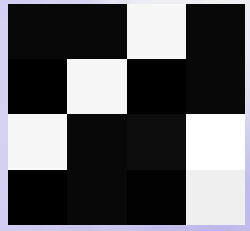
- We can compress small contrast images
- We can scale, but up to 4 times

Compression of special small 4x4 test image:

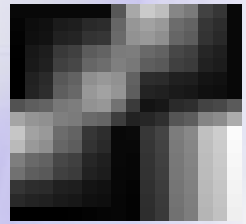


Original, size 4x4 pixels

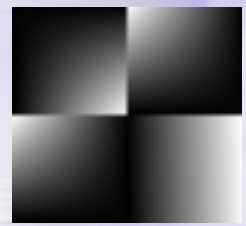
Original method



Hybrid method



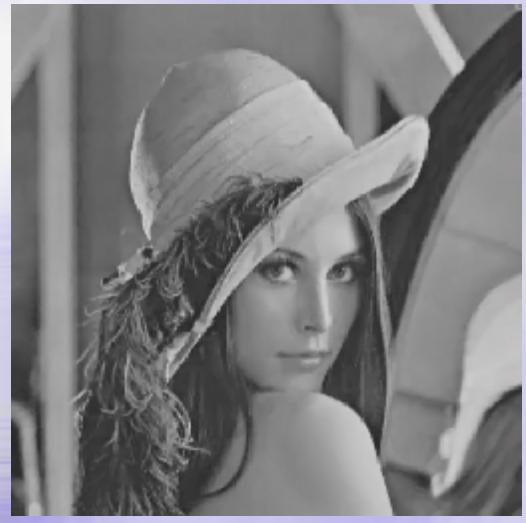
Scaling: Hybrid, 4 times



Scaling: Hybrid, 4 times

Example: Grayscale lena image

Original



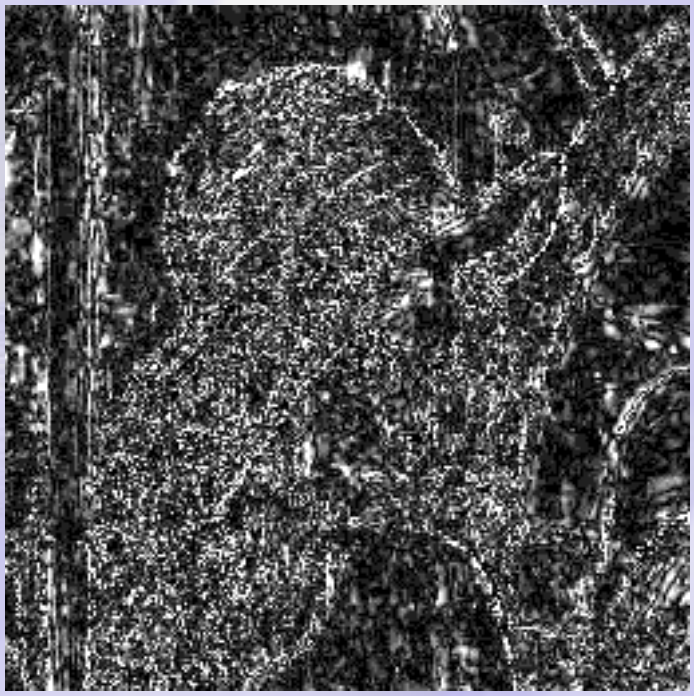
With rate 4.4 times

With rate 10.5 times

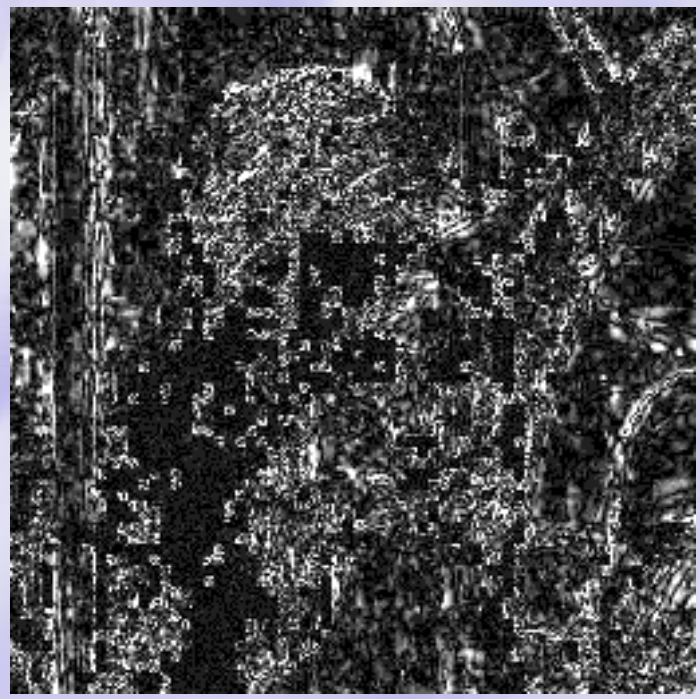
Results: Increased Quality (1)

Difference with original image

Original method

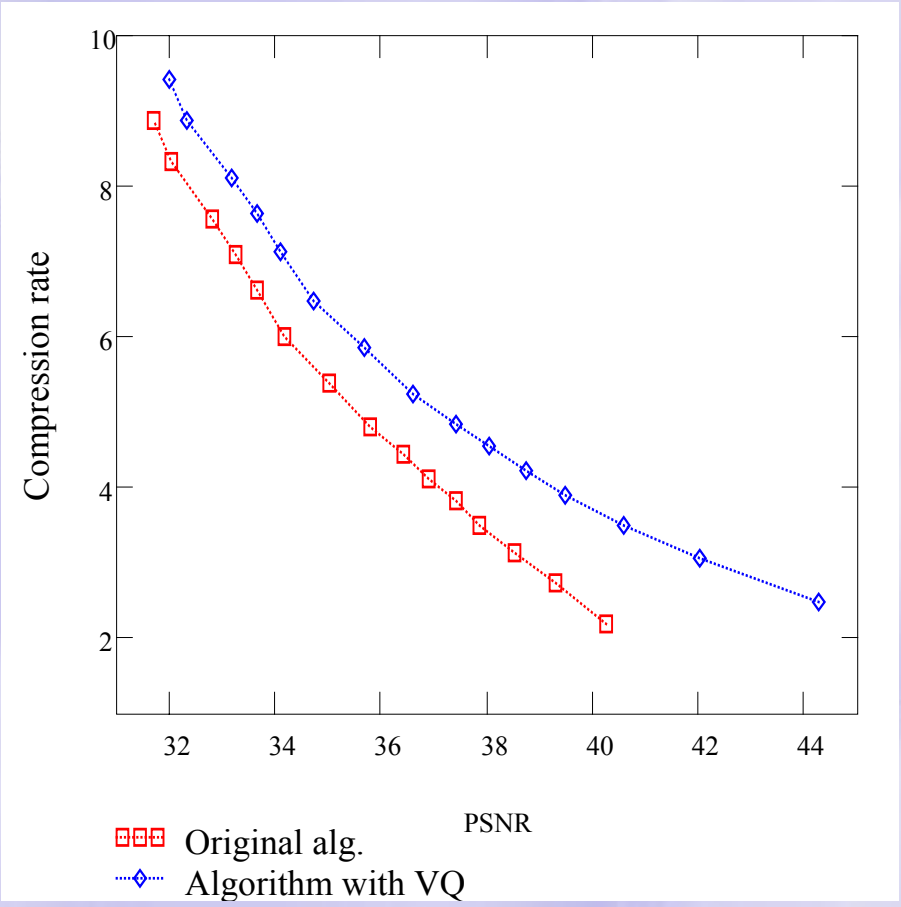


Hybrid method

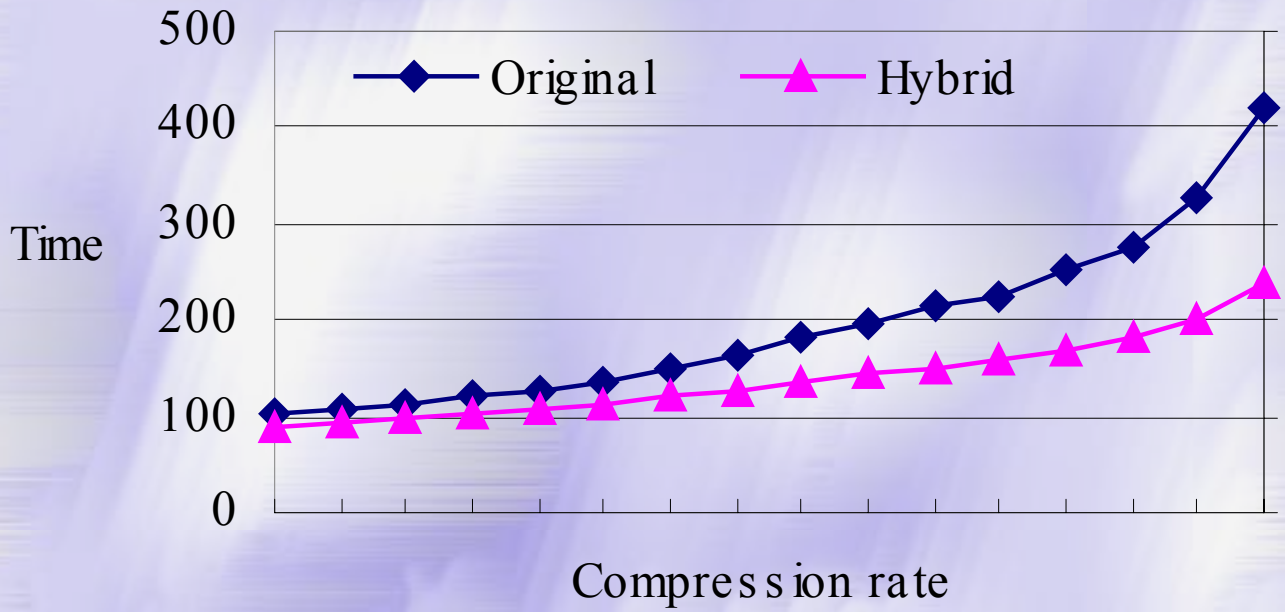


Results: Increased Quality (2)

Quality increased for low compression rate



Results: Faster compression



Compression speed increased up to 2 times

Results: Faster decompression(1)

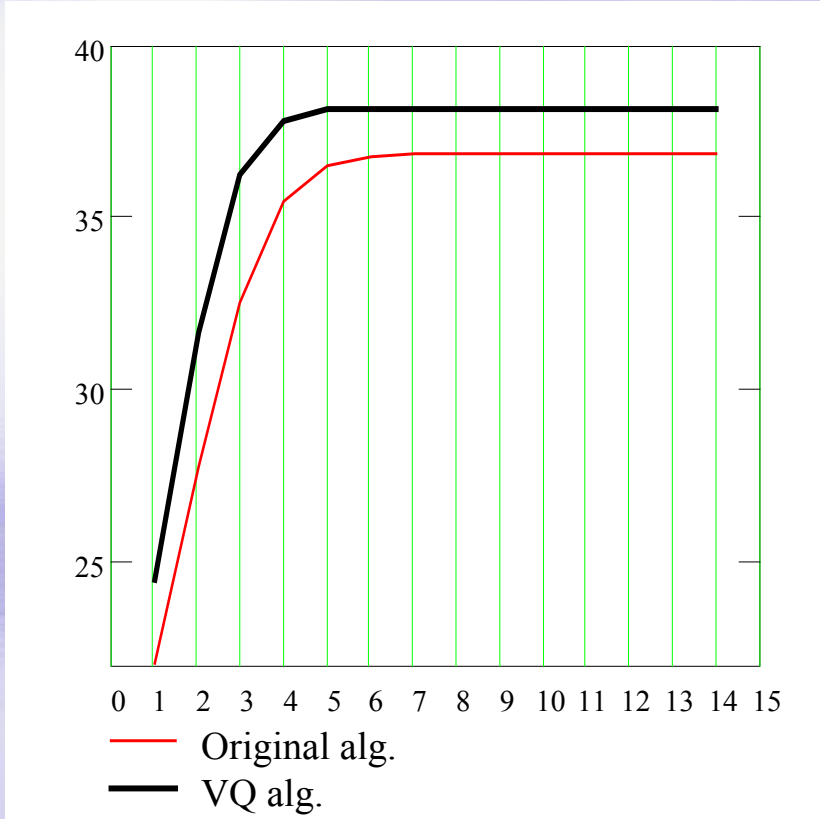
Image for 2-nd decompression iteration

Original method

Hybrid method

Results: Faster decompression(2)

Quality of image for number of iterations:

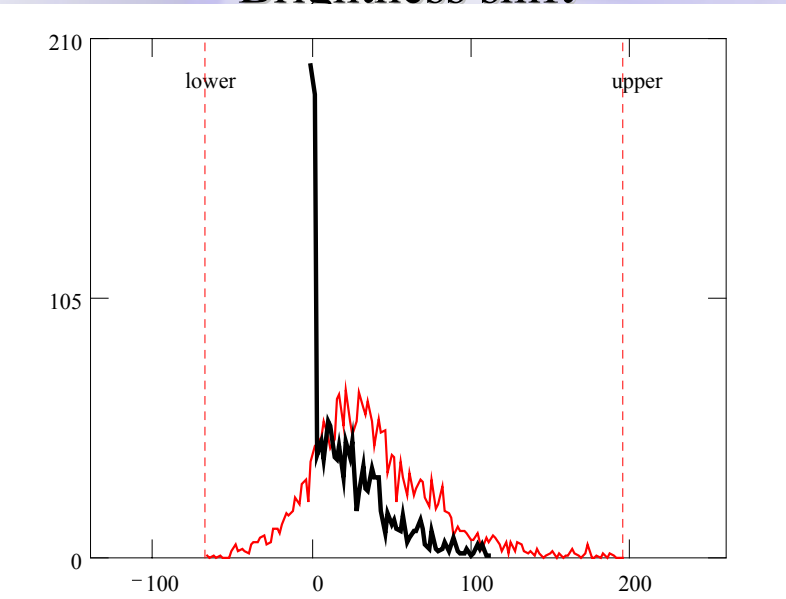


Results: Compact codepack

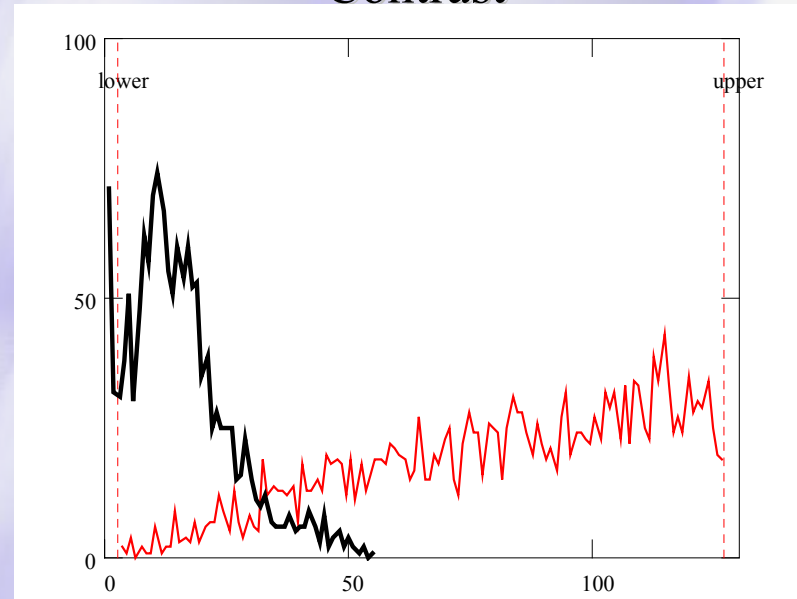
Histogram of distribution of the affine coefficients

Red line — original method, black — hybrid

Brightness shift



Contrast

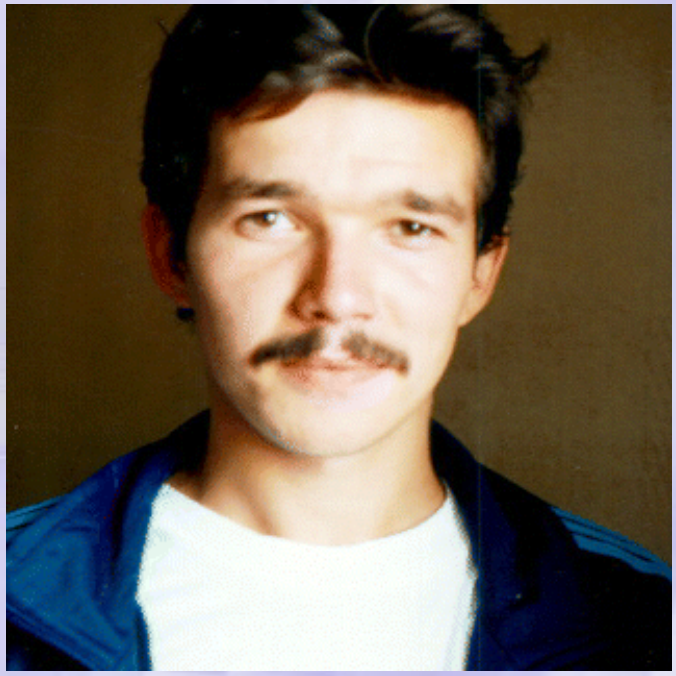


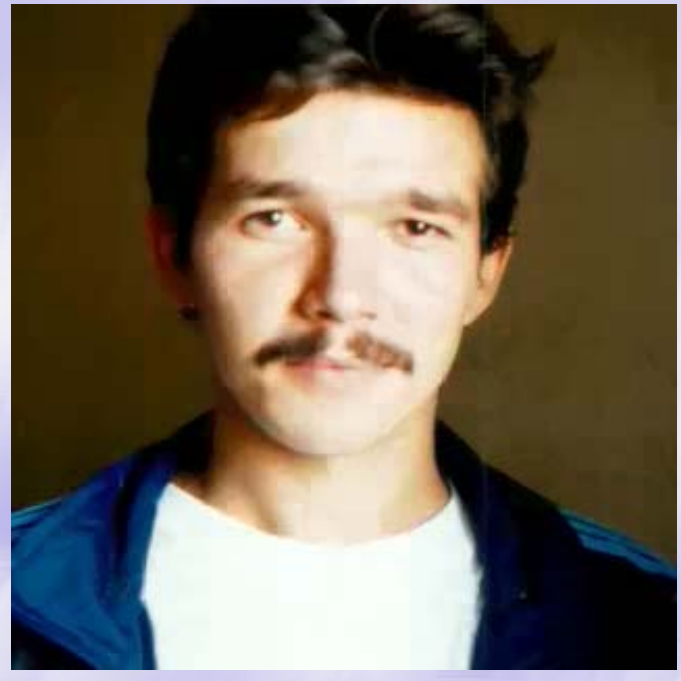
Conclusion

Hybrid method:

- Increase compression speed
- Increase decompression speed
- Increase quality & increase compression rate (compact codepack)

Example: Color photo dm (1)



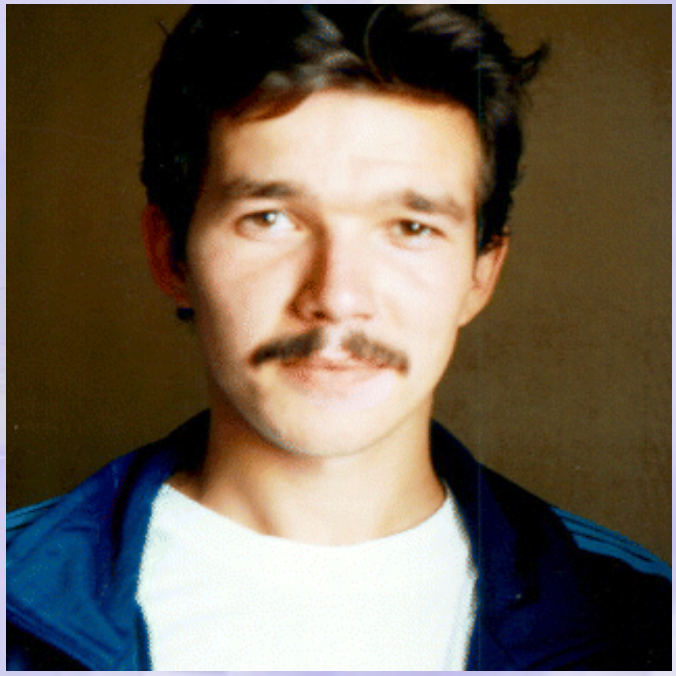


Original image (307Kb)

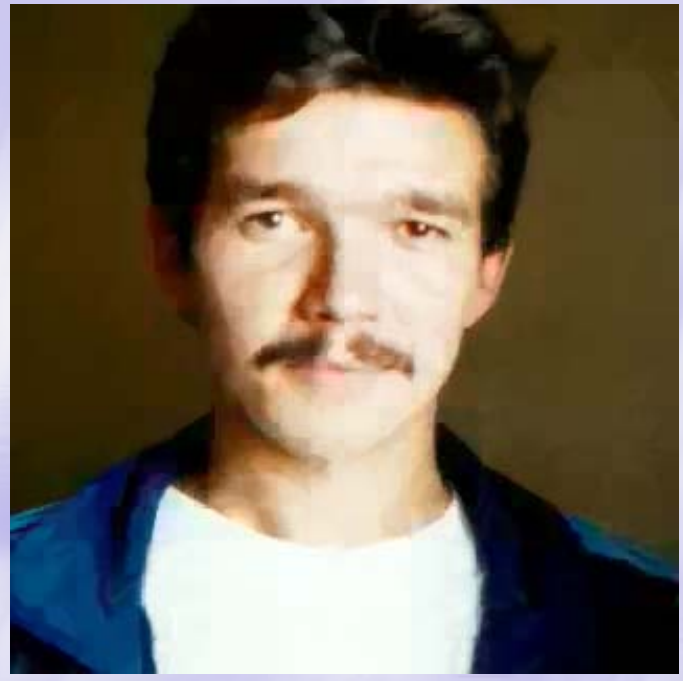
Rate 40 times (7.67)

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

Example: Color photo dm (2)

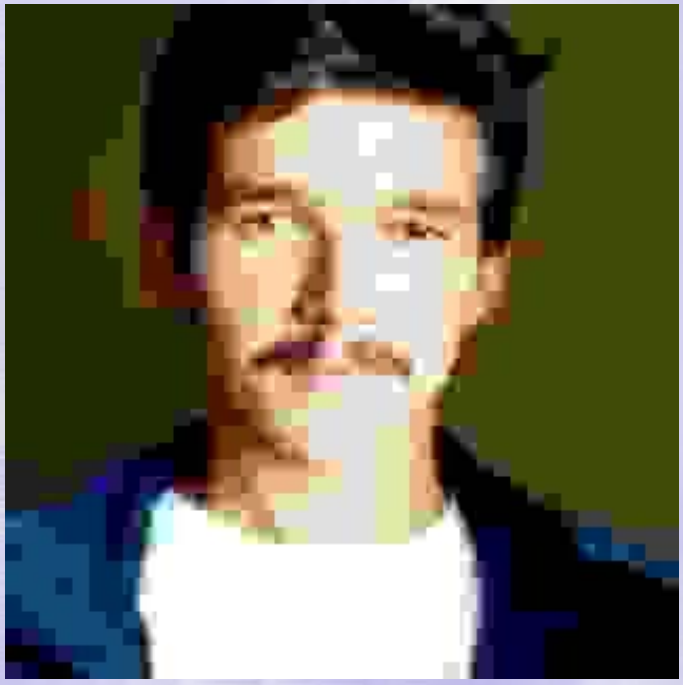


Original image (307Kb)



Rate 100 times (3.08Kb)

Compare with JPEG



JPEG: Rate 100 times (3.06Kb)

FIF: Rate 100 times (3.08Kb)

References

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

[2] Y. Fisher, "*Fractal image compression*", SigGraph 92.

[3] A.E. Jacquin, "*Fractal image coding*" A review Proceedings of the IEEE 81,10 (1993) 1451-1465.

[4] M. Gharavi-Alkhansari, T. Huang, "*Fractal based techniques for a generalised image coding method*" Proc. IEEE ICIP-94, Austin, Texas, Nov 1994.

[5] M. Barnsley, L. Hurd, "*Fractal Image Compression*" AK Peters. 1993

[6] S. Lepsoy, "*Block-based attractor coding: Potential and comparison to vector quantization*" Conference on Signals, systems and Computers, pp. 1504-1508, 1993

[7] T.Murakami K/Asai, E/Yamazaki. "*Vector quantiser of video signals*", Electronics Letters 7 (1982) 1005-1006

[8] R. Hamzaoui, M.Muller, D.Saupe "*VQ-enhanced fractal image compression*" IEEE ICIP'96

[9] Д.С. Ватолин. "*Тенденции развития алгоритмов архивации графики*" Открытые системы. N-4. Зима 1995

The end

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
Хаффмана	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

Тенденции развития алгоритмов

- Ориентация на фотореалистичные изображения
- Использование сжатия с потерями
- Использование избыточности в двух измерениях
- Появление существенно несимметричных алгоритмов
- Увеличивающаяся степень сжатия изображений