Synergy: Cross-Disciplinary
Journal of Digital Investigation
(ISSN 2995-4827) VOLUME 02 ISSUE 4, 2024

# IMAGE COMPRESSION METHODS, FRACTAL COMPRESSION

***Khurramova Rano Ibragimovna***
*Associate Professor of the Department of Convergence of Digital Technologies of the Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Ph.D.*

***Egamberdiyeva Nodirabegim Abdunamid qizi, Almatova Nodira Azimjon qizi***
*Master of Tashkent University of Information Technologies named after Muhammad al- Khwarizmi*

**Abstract:**

This article describes image compression, its methods, advantages and disadvantages of each method. Also, the method of fractal compression of images is studied more widely and compared with other methods.

***Keywords:*** *image, image compression, compression methods, lossless and lossy compression, fractal, fractal compression, fractal coding, fractal decoding, jpeg, WebP, compression ratio, tile symmetry*.

It is easy to calculate that an uncompressed full-color image with a size of 2000 * 1000 pixels will have a size of about 6 megabytes. If we talk about images obtained from professional cameras or high-resolution scanners, then their size can be even larger. Despite the rapid growth in the capacity of storage devices, various image compression algorithms are still very relevant.

All existing algorithms can be divided into two large classes:

- ✓ Lossless compression methods;
- ✓ Lossy compression methods.

When we talk about lossless compression, we mean that there is an inverse algorithm to the compression algorithm that allows you to accurately restore the original image. There is no inverse algorithm for lossy compression algorithms. There is an algorithm that restores an image that does not necessarily exactly match the original one. Compression and recovery algorithms are selected to achieve a high compression ratio while maintaining the visual quality of the image.

This article discusses lossless compression methods

**JPEG** is one of the newest and quite powerful algorithms. In practice, it is the de facto standard for full-color images [1]. The algorithm operates on 8x8 areas in which brightness and color change relatively smoothly. As a result, when decomposing the matrix of such an area into a double series in cosines (see formulas below), only the first coefficients are significant. Thus, JPEG compression is achieved by smoothly changing colors in the image.

In general, the algorithm is based on a discrete cosine transform (DCT) applied to the image matrix to obtain a new coefficient matrix. An inverse transformation is applied to obtain the original image.

The JPEG compression method is performed in the following sequence:

1. Translate the image from the RGB color space, with the component, the red (Red), green (Green) and blue (Blue) light source, and the color space YCrCb (called YUV in the mirror);
2. Splitting image into 8x8 matrics;
3. Apply DCT (Dual-Clutch Transmission) to each matrix.
4. Perform quantization.
5. Converting an 8x8 matrix into a 64-element vector using zigzag scanning;
6. Collapsing the vector using the group encoding algorithm.
7. Collapsing the resulting pairs using Huffman coding with a fixed table.

The significant positive aspects of the algorithm:

- Sets the compression ratio.
- The output color image can be 24 bits per dot.
- The negative aspects of the algorithm:
- As the compression level increases, the image breaks up into individual squares (8x8). This is due to the fact that large losses occur at low frequencies during quantization, and it becomes impossible to restore the original data.
- The Gibbs effect appears - halos along the boundaries of sharp color transitions.

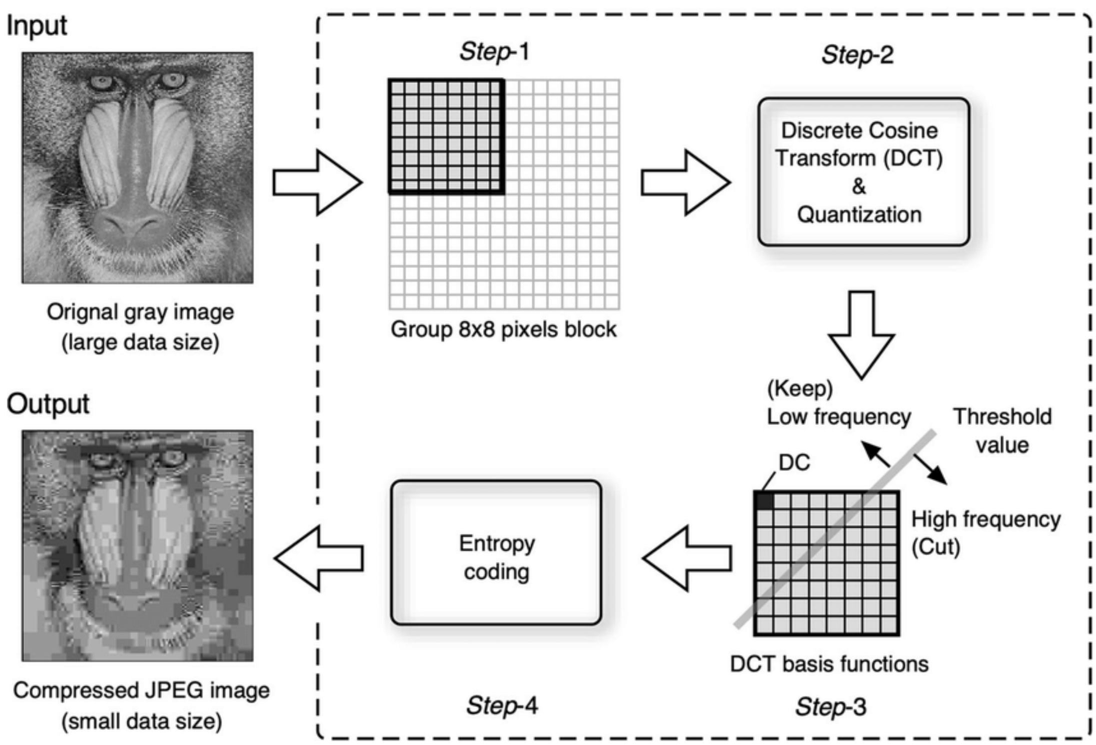


**Figure 1. JPEG image compression method steps [1]**

*JPEG algorithm characteristics:*

- Compression ratios: 2-200 (user-defined).
- Image class: Full-color 24-bit images or grayscale images without sharp color transitions (photographs).
- Symmetry: 1
- Characteristic features: In some cases, the algorithm creates Gibbs effect. In addition, with a high degree of compression, the image is divided into blocks of 8x8 pixels.

**WebP** - (WEB Pictures) image compression format, both lossy and lossless, proposed by Google Inc. in 2010 year. Based on the still image (key frame) compression algorithm from the VP8 video codec. Uses the RIFF container.

Compression consists of two stages. On the first, an attempt is made to "predict" the contents of some blocks based on already decoded ones (three blocks above the current one and one block to the left of it), on the second, the prediction error is encoded. Blocks are drawn in order from left to right and top to bottom [3].

The DCT is used to compress prediction errors and subblocks that were not predicted. Both transformations operate on 4x4 pixel subblocks. The implementation of the transformations is carried out on a fixed-precision representation of numbers in order to reduce rounding errors [5]. The DCT and WHT coefficients are packed using an entropy codec.

WebP does not work in the RGB color space; before encoding, the image is converted to YUV with a depth of 8 bits and a 4:2:0 format. Translation is carried out according to the ITU-R BT.601 standard

Method advantages:

- ✓ Smaller file size
- ✓ Improved compression algorithm
- ✓ Better color transitions
- ✓ Alpha Channel Mask

Method disadvantages:

- ✓ Poor browser support
- ✓ Distortion has a plastic look
- ✓ Inconvenient export interface

*WEBP algorithm characteristics [4]:*

- ✓ Compression ratios: 2-300 (user-defined).
- ✓ Image class: 32-bit images
- ✓ Symmetry: 1.5

**Fractal compression** is based on the fact that we represent the image in a more compact form - using the coefficients of the Iterated Function System (hereinafter referred to as IFS). Before considering the archiving process itself, let's look at how IFS builds an image decompression process.

Strictly speaking, IFS is a set of three-dimensional affine transformations, in our case transforming one image into another. Points in three-dimensional space (x_coordinate, y_coordinate, brightness) undergo transformation.

The two most famous images obtained using IFS are the “Sierpinski triangle” and the “Barnsley fern”. The “Sierpinski triangle” is defined by three, and the “Barnsley fern” by four affine transformations (or, in our terminology, “lenses”). Each transformation is encoded in literally a few bytes, while the image constructed using them can occupy several megabytes.

In fact, fractal compression is a search for self-similar areas in an image and determination of affine transformation parameters for them (figure 2).

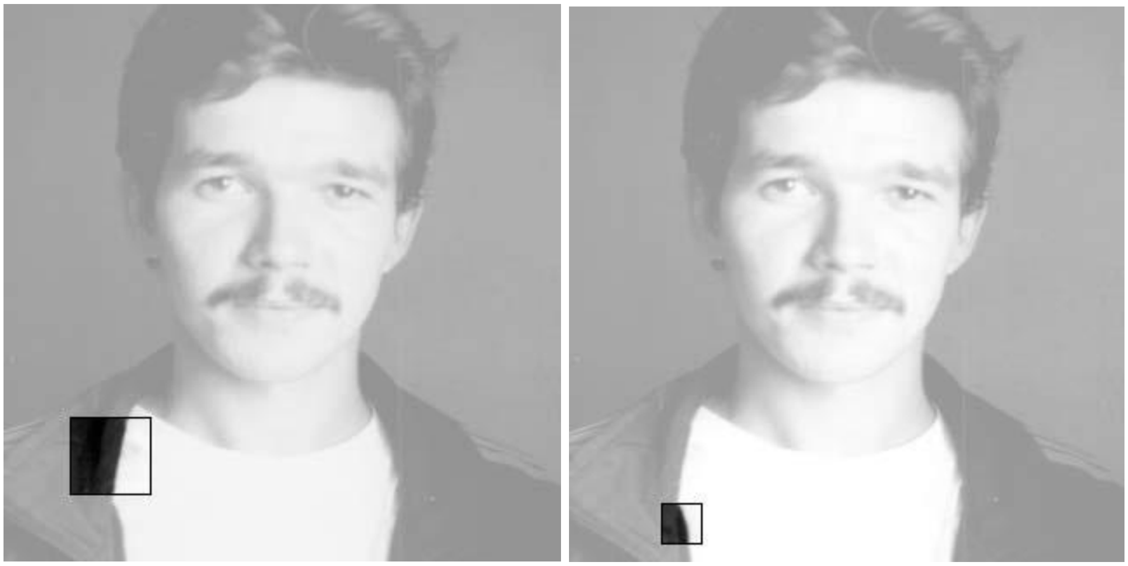

**Figure 2. Searching for self-similar areas.**

In the worst case, if no optimizing algorithm is used, it will be necessary to enumerate and compare all possible image fragments of different sizes. Even for small images, taking discreteness into account, we get an astronomical number of options to be sorted out. Moreover, even a sharp narrowing of transformation classes, for example, due to scaling only a certain number of times, does not provide a noticeable gain in time. In addition, image quality is lost. The vast majority of research in the field of fractal compression is now aimed at reducing the archiving time required to obtain a high-quality image.

As has already become obvious from the above, the main task when compressing with a fractal algorithm is to find the corresponding affine transformations. In the most general case, we can translate image areas of any size and shape, but in this case we get an astronomical number of iterations of different fragments, which cannot currently be processed even on a supercomputer.

In the training version of the algorithm outlined below, the following restrictions are made on the areas:

- All areas are squares with sides parallel to the sides of the image. This limitation is quite strict. In fact, we are going to approximate the entire variety of geometric shapes with just squares.
- When converting a domain area to a rank area, the size is reduced exactly by half. This greatly simplifies both the compressor and the decompressor, because the task of scaling small areas is non-trivial.
- All domain blocks are squares and have a fixed size. The image is divided into a set of domain blocks using a uniform grid.
- Domain areas are taken “through a point” both in X and Y, which immediately reduces the search by 4 times.

- When converting a domain area to a rank area, rotation of the cube is only possible by 00, 900, 1800 or 2700. Mirror reflection is also allowed. The total number of possible transformations (counting empty ones) is 8.
- Scaling (compression) vertically (brightness) is carried out a fixed number of times - 0.75.

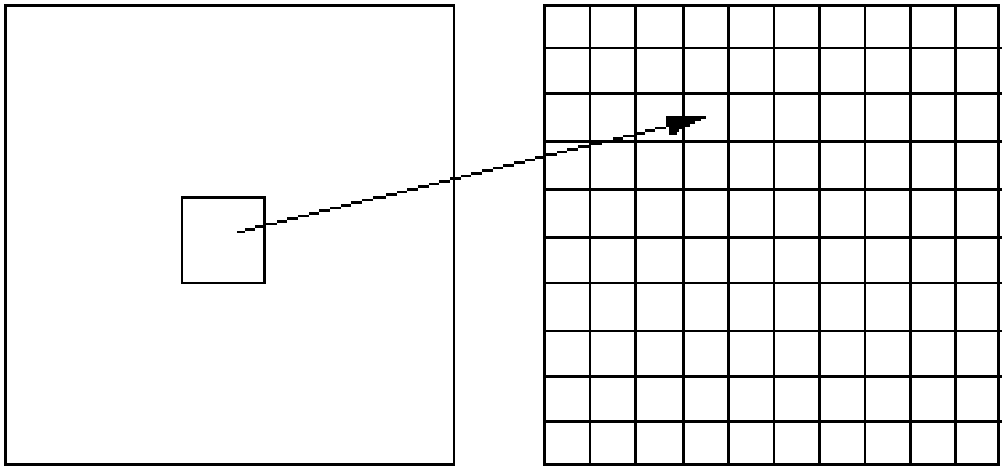

**Figure 3. comparing domain and range matrics**

The packing algorithm itself boils down to enumerating all domain blocks and selecting a corresponding rank block for each. Below is a diagram of this algorithm:

```
for (all range blocks) {
        min_distance = MaximumDistance;
        Rij = image->CopyBlock(i,j);
        for (all domain blocks) {
                current=Координаты тек. преобразования;
                D=image->CopyBlock(current);
                current_distance = Rij.L2distance(D);
                if(current_distance < min_distance) {
                        min_distance = current_distance;
                        best = current;
                }
        }
        Save_Coefficients_to_file(best);
}
```

*Characteristics of the fractal algorithm:*

- Compression ratios: 2-2000 (user-defined).
- Image class: Full-color 24-bit images or grayscale images without sharp color transitions (photographs). It is desirable that areas of greater significance (for perception) be more contrasting and sharp, and areas of lesser significance should be low-contrast and blurred.
- Symmetry: 100-100000

- Characteristic features: Can freely scale the image when unzipping, enlarging it 2-4 times without the appearance of a “staircase effect”. As the degree of compression increases, a “blocky” effect appears at the boundaries of blocks in the image.

**Table 1. Comparison of 3 image compression methods**

| N | Image compression method | Compression ratio | Image class | Time simmetry |
|---|---|---|---|---|
| 1 | JPEG | 2-200 | 24-bit | 1 |
| 2 | WEBP | 2-300 | 32-bit | 1.5 |
| 3 | Fractal | 2-2000 | 24-bit | 100-100000 |

It can be seen from Table 1 that the highest image compression ratio was observed in fractal compression. From the Time Symmetry column, we can see that the encoding and decoding time of the method is highest.

**REFERENCES**


1. Shien Ri “Dynamic Deformation Measurement by the Sampling Moiré Method from Video Recording and its Application to Bridge Engineering”, Experimental Techniques, 25.07.2019
2. Inside WebM Technology: VP8 Intra and Inter Prediction — The WebM Open Media Project Blog, 3 октября 2010.
3. VideoBits.org — Prediction, 3 октября 2010.
4. https://developers.google.com
5. https://ru.wikipedia.org/wiki/WebP