Color Image Compression using Wavelets

A dissertation submitted in partial
Fulfillment of the requirements of M.Tech in Computer
Science degree of Indian Statistical Institute, Kolkata

By

Barindra Nath Dutta

Under the supervision of

Prof. C. A. Murthy

11th July 2003

Indian Statistical Institute
203, Barrackpore Trunk Road,
Kolkata-700 108.

Indian Statistical Institute
203, Barrackpore Trunk Road,
Kolkata-700108

Certificate of Approval

This is to certify that this thesis titled "Color Image Compression using Wavelets" submitted by Barindra Nath Dutta towards partial fulfillment of requirements for the degree of M. Tech in Computer Science at Indian Statistical Institute, Kolkata embodies the work done under my supervision.

10.7.03

Prof. C. A. Murthy
Machine Intelligence Unit
Indian Statistical Institute, Kolkata-700 108.

Acknowledgements

I like to express my deep sense of gratitude and thanks to my supervisor, Prof. C.A. Murthy, Machine Intelligence Unit, Indian Statistical Institute, Kolkata for his invaluable guidance, suggestions and encouragement throughout the project work.

I also like to thank my classmates for their constant help and encouragement. Last but not the least, I have to mention, the support and care I got from my parents and family members.

Date - 10.07.03

Barindra Nath Dutta

Barindra Nath Dutta

Contents

Chapter 1

Introduction

1.1 Image Compression

Image Compression refers to a process in which the amount of data used to represent image is reduced to meet a bit rate requirement, while the quality of the reconstructed image satisfies a requirement for a certain application and the complexity of computation involved is affordable for the application. Over the past few years the world has witnessed a growing demand for visual based information and communications applications. As the demand for new applications and higher quality for existing applications continues to rise, the transmission and storage of the visual information becomes a critical issue. The reason is that the higher image or video quality requires larger volume of information. How ever transmission media have a finite and limited bandwidth. To illustrate this problem consider a typical (512x512) gray level (8-bit) image. This image has 20,97,152 bits. By using a 64 Kbit/s communication channel, it will take about 33 seconds to transmit the image. Where as this might be acceptable for a one-time transmission of a single image, it would definitely not be acceptable for tele-conference applications, where some form of continuous motion is required. To store digital version of a 90 minute black and white movie, at 30 frames/sec, with each frame having (512x512x8) bits, would require 3.397386e+11, over 42 Gbytes. So efficient image compression algorithms are appreciable.

1.2 Principles behind compression.

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the image. Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System.

In general, two types of redundancy can be identified:

* Spatial Redundancy: within an image there exists significant correlation among neighboring pixels.

* Spectral Redundancy: For color Image there exists significant correlation among color planes.

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible.

1.3 Different classes of compression techniques

Two ways of classifying compression techniques are mentioned here.

(a) Lossless vs. Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually loss less).

(b) Predictive vs. Transform coding: In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computation.

1.4 Organization of the thesis.

The thesis is organized as follows.

Chapter 2 surveys the compression algorithms currently in use and different color spaces.
Chapter 3 provides an overview of wavelets and Image Compression

Chapter 4 provides the Proposed Algorithms and Implementation Details

Chapter 5 shows the Results and Comparison.

Chapter 6 provides conclusions, discussions and scope for further work.

Chapter 2
Literature Survey

In this chapter we will discuss about the different color spaces and common Image Compression techniques.

2.1 Color Space and Human Perception

A digital color image can be viewed as a three valued (channels) positive function I=I (x, y) defined onto a plane. Its algebraic representation is obtained through an N by M by 3 matrix A. Thus, each entry of A is a three-component integer vector (pixel color) expressing an intensity value at discrete location (x, y) with a precision p (for instance, one bit for each channel). Each component or layer of the image can be viewed as a single channel image, which, under particular conditions, can be analyzed independently from the others. This is not the case for RGB space, because, if two channels are fixed, human visual perception is very sensitive to small changes of the value of the remaining channel. Thus, even though RGB is the most common storage format for images, other formats may be better for compression.

The key step in lossy data compression in which data cannot be recovered exactly is the quantization phase, which exploits a data reduction based on their low information content. This is not optimal for RGB images. Nevertheless, for three layers, this can lead to the elimination of some low coefficients in a channel in a certain spatial location, even though the corresponding coefficients in the other layers are not eliminated because they carry high information content. When reconstructing the image at that location, a high visual distortion is introduced. The assumption of analyzing the three layers separately is valid only if they are not correlated with respect the visual appearance.

RGB Space

RGB is perhaps the simplest and most commonly used color space .It uses proportions of red, green, and blue that are scaled to a minimum and maximum values for each component (for example, 0 through 255). Most colors in the visible spectrum can be recreated, although not completely. This scheme is based on the additive properties of color. [10].

YUV Space

YUV was originally used for PAL (European standard) analog video. To convert from RGB to YUV spaces, the following equations can be used:

$$Y = 0.299\,R + 0.587\,G + 0.114\,B$$

$$U = 0.492\,(B - Y)$$

$$V = 0.877\,(R - Y)$$

Any errors in the resolution of the luminance (Y) are more important than the errors in the chrominance (U, V) values. The luminance information can be coded using higher bandwidth than the chrominance information [10].

YIQ Space

YIQ is used in the U.S. television standard, NTSC (National Television System Committee). It is similar to YUV, except that its color space is rotated 33 degrees clockwise, so that I is the orange-blue axis, and Q is the purple-green axis. The equations to convert from RGB to YIQ are [10]

$$Y = 0.299\ R + 0.587\ G + 0.114\ B$$

$$I = 0.74\ (R - Y) - 0.27\ (B - Y) = 0.596\ R - 0.275\ G - 0.321\ B$$

$$Q = 0.48\ (R - Y) + 0.41\ (B - Y) = 0.212\ R - 0.523\ G + 0.311\ B$$

YCrCb Space

The YCrCb color space was developed as part of Recommendation ITU-RBT.601 (formerly CCIR 601) during the development of a worldwide digital component video standard. Y is defined to have a range of 16 to 235; Cb and Cr are defined to have a range of 16 to 260 with 128 equal to zero. The linear transform from RGB to YCrCb generates one luminance space Y and two chrominance Cr and Cb spaces and is given by [3]

$$Y = (77/256)\ R + (150/256)\ G + (29/256)\ B$$

$$Cr = (131/256)\ R - (110/256)\ G - (21/256)\ B + 128$$

$$Cb = -(44/256)\ R - (87/256)\ G + (131/256)\ B + 128$$

YCrCb to RGB conversion

$$R = Y + 1.371(Cr-128)$$

$$G = Y - 0.698(Cr-128) - 0.336(Cb-128)$$

$$B = Y + 1.732(Cb-128)$$

Comparison of Color Spaces

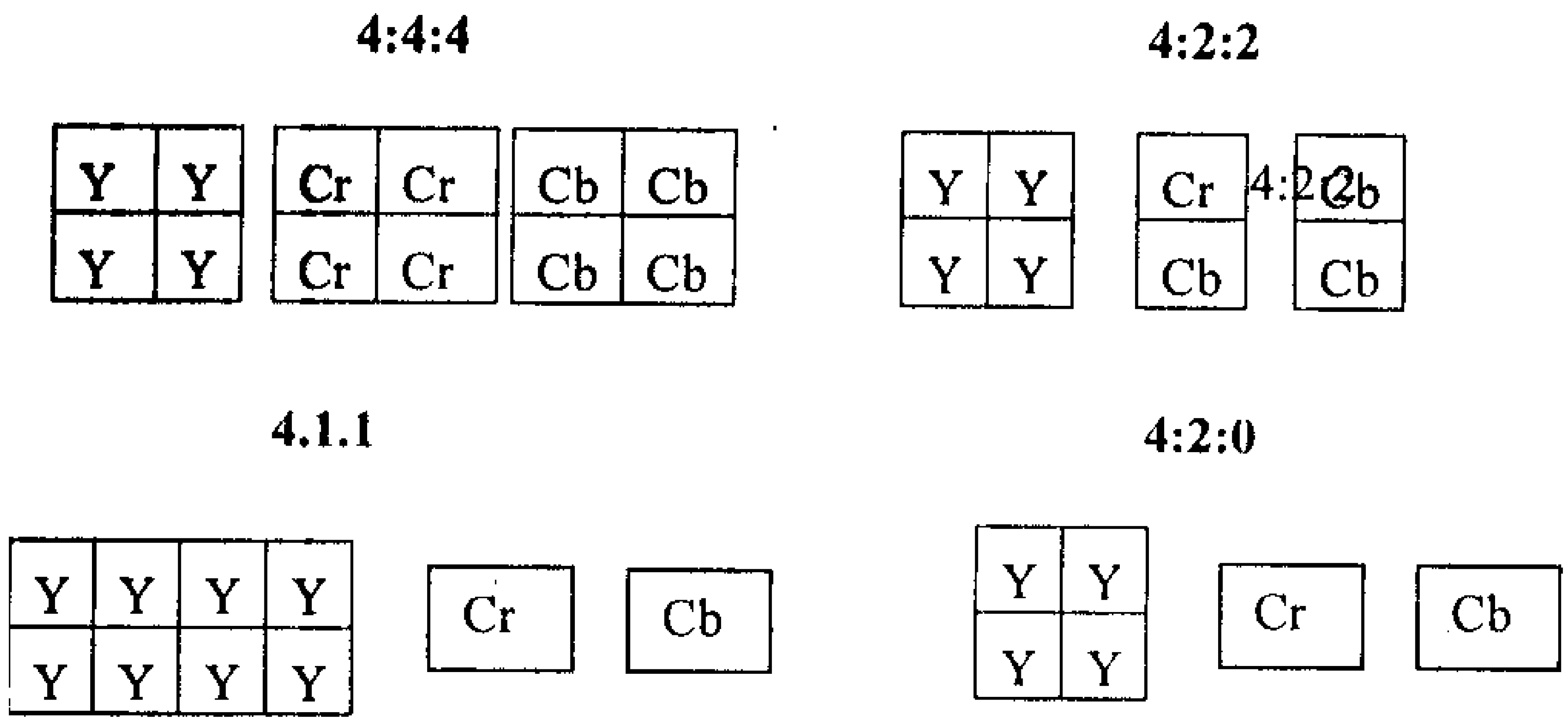
Among these three-color spaces, only YCrCb is the proper one used in the digital image processing because

1) There is no correlation between the spaces of YCrCb, so each space can be analyzed separately.

2) Human eye is more sensitive to the change of brightness than of color, so Cr and Cb Chrominance Space can be compressed more heavily than Luminance Space Y to get better compression ratio.

Color Sub Sampling

Since Human Visual System is more sensitive to the change of intensity than the color and as Luminance is controlled by the intensity of the signal and Chrominance is the color signal, so sub sampling of the chrominance space can be done to achieve better compression. Sample patterns 4:4:4, 4:2:2, 4:1:1 and 4:2:0 are all commonly used. 4:4:4 has no color sub sampling. 4:2:2 sub samples both C_B and C_R by two in the horizontal direction. This representation is used in Rec. 601 DTV, though the other sample patterns are possible. 4:1:1 is by 4-sub sampling of the color difference signals in the horizontal direction, it is not used extensively. 4:2:0 sub samples both color signals by 2 in both the horizontal and vertical dimensions, it is used extensively, especially in low bandwidth representations (which have marginal pictures anyhow) such as CDROM (e.g., MPEG1) "multimedia" video.[11]



2.2 Compression Techniques

Compression takes an input f (x, y) and generates a representation $f^C(x, y)$ that hopefully requires fewer bits. There is a reconstruction algorithm that operates on the compressed representation $f^C(x, y)$ to generate the reconstruction g (x,y).

Based on the requirements of reconstruction, Image compression schemes can be divided into two broad classes. One is loss less compression, in which g (x,y) is identical to f(x,y). Examples of loss less methods are Run Length coding, Huffman coding, Lempel/Ziv algorithms, and Arithmetic coding. The other is lossy compression, which generally provides much higher compression than loss less compression but allows g(x,y) to be different from f(x,y).

Fractal Image Compression:

The application of fractals in image compression started with M.F. Barnsley and A.Jacquin [6]. Fractal image compression is a process to find a small set of mathematical equations that can describe the image. By sending the parameters of these equations to the decoder, the original image can be reconstructed.

In general, the theory of fractal compression is based on the contraction-mapping theorem in the mathematics of metric spaces. The Partitioned Iterated Function System (PIFS), which is essentially a set of contraction mappings, is formed by analyzing the image. Those mappings can exploit the redundancy that is commonly present in most images. This redundancy is related to the similarity of an image with itself, that is, part A of a certain image is similar to another part B of the image, by doing an arbitrary number of contractive transformations that can bring A and B together. These contractive transformations are actually common geometrical operations such as rotation, scaling, skewing and shifting. By applying the resulting PIFS on an initially blank image iteratively, we can completely regenerate the original image at the decoder. Since the PIFS often consists of a small number of parameters, a huge compression ratio (e.g. 500 to 1000 times) can be achieved by representing the original image using these parameters.

However, fractal image compression has its disadvantages. Because fractal image compression usually involves a large amount of matching and geometric operations, it is time consuming. The coding process is so asymmetrical that encoding of an image takes much longer time than decoding.

Segmented Image Coding

Segmented Image Coding [7] is a region-oriented coding, which considers images to be composed of regions of slowly varying image intensity, separated by image edges. The texture inside each region is approximated by a bi-variate polynomial. The coefficients of these polynomials, and the location of the contours separating the image regions constitute the contour-texture image model. The visually extremely important image contours are always well retained by this technique, which guarantees a reasonable image quality, even at very low bit rates. In contrast to some conventional compression techniques, region oriented techniques are very well suited for progressive image transmission: in the reconstruction phase, the image is gradually built up. This is important in, e.g., database applications, where the gradual image build-up allows the user to quickly decide whether or not the transmitted image is the correct one required

Transform Based Image Compression

The basic encoding method for transform based compression works as follows:

1. Image transform

Divide the source image into blocks and apply the transformations to the blocks.

2. Parameter quantization

The data generated by the transformation are quantized to reduce the amount of information. This step represents the information within the new domain by reducing the amount of data. Quantization is in most cases not a reversible operation because of its lossy property.

3. Encoding

Encode the results of the quantization. This last step can be error free by using Run Length encoding or Huffman coding. It can also be lossy if it optimizes the representation of the information to further reduce the bit rate.

Transform based compression is one of the most useful applications. Combined with other compression techniques, this technique allows the efficient transmission, storage, and display of images that otherwise would be impractical.

The Discrete Cosine Transform (DCT) [9] was first applied to image compression in the work by Ahmed, Natarajan, and Rao. It is a popular transform used by the JPEG (Joint Photographic Experts Group) image compression standard for lossy compression of images. Since it is used so frequently, DCT is often referred to in the literature as JPEG-DCT, DCT used in JPEG.

JPEG-DCT is a transform coding method comprising four steps. The source image is first partitioned into sub-blocks of size 8x8 pixels in dimension. Then each block is transformed from spatial domain to frequency domain using a 2-D DCT basis function. The resulting frequency coefficients are quantized and finally output to a loss less entropy coder. DCT is an efficient image compression method since it can decorrelate pixels in the image (since the cosine basis is orthogonal) and compact most image energy to a few transformed coefficients. Moreover, DCT coefficients can be lossyly quantized according to some human visual characteristics. Therefore, the JPEG image file format is very efficient. This makes it very popular, especially in the World Wide Web.

Still Image Compression techniques using wavelets - JPEG 2000

JPEG has been designed as a compression method for continuous-tone images. The word JPEG is an acronym that stands for Joint Photographic Experts Group. The JPEG compression is based on discrete cosine transform (DCT) and it has proved to be successful and has become widely used for image compression, especially in web pages. The use of DCT on 8x8 blocks of pixels results sometimes in a reconstructed image that has a blocky appearance. That is why the JPEG committee decided, as early as 1995, to develop a new, wavelet-based standard for the compression of still images, to be known as JPEG 2000 or JPEG Y2K. The following paragraph is a short summary of how the JPEG 2000 algorithm works [8].

Given an image it is partitioned into rectangular, no overlapping regions called tiles, which are compressed individually. If the image being compressed is in color, it is divided into three components.

A tile is compressed in four main steps.

1. Compute a wavelet transform that results in sub bands of wavelet coefficients. Two such transformation an integer and a floating point, are specified by the standard. There is L+1 resolution levels of sub bands, where L is a parameter determined by the encoder.

2. The wavelet coefficients are quantized. This is done if the user specifies a target bit rate. The lower the bit rate, the coarser the wavelet coefficients have to be quantized.

3. Use MQ coder to arithmetically encode the wavelet coefficients. The EBCOT algorithm has been adopted for the encoding step. The principle of EBCOT is to divide each sub band into blocks, called code-blocks, which are coded individually. The bits resulting from coding several code-blocks become a packet and the packets are the components of the bit stream.

4. The last step is to construct the bit stream. The markers can be used by the decoder to skip certain areas of the bit stream and to reach certain points quickly. Using markers, the decoder can for e.g., decode certain code-blocks before others, thereby displaying certain regions of the image before other regions. The bit stream is organized in layers, where each layer contains higher-resolution image information. Thus decoding the image layer by layer is a natural way to achieve progressive image transmission and decompression.

The details about these steps can be found in [12]

Current experiments indicate that JPEG 2000 performs better than the original JPEG, especially for images where very low bit rates or very high image quality are required. For lossless or near-loseless compression, JPEG 2000 offers only modest improvements over JPEG.

Chapter 3

In this chapter, we discuss wavelets. For a better understanding the need of wavelet transform, we also briefly discuss the Fourier transform, which is most popularly used for signal processing.

3.1 Fourier Transform

In 19th century, the French mathematician, J. Fourier, showed that any periodic function could be expressed as an infinite sum of periodic complex exponential functions. Many years after this remarkable property of periodic functions was discovered, the ideas were generalized to non-periodic functions, and then to periodic or non-periodic discrete time signals. After this, Fourier transform became a very famous tool for computer calculations.

The equation $F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$ is generally called the Fourier Transform. The equation $f(t) = \int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega$ is called the inverse Fourier Transform.

Note that in the Fourier Transform equation, the integration is from minus infinity to plus infinity over time. So, no matter when the component with frequency ω appears in time, it will affect the result of the integration equally as well. The lack of time information is one serious weakness of Fourier Transform. That is why Fourier transform is not suitable if the signal has time varying frequency, i.e., the signal is non-stationary.

3.2 Windowed Fourier Transform

To solve the above problem, the Windowed Fourier Transform is used. The basic idea is to divide the signal into small enough segments, where these segments can be assumed to be stationary. The width of this window must be equal to the segment of the signal where this assumption is valid.

The Windowed Fourier Transform has several problems. If we use a window of infinite length, we get the Fourier Transform, which gives perfect frequency resolution, but no time information. On the other hand, in order to obtain a stationary sample, we must have a small enough window in which the signal is stationary. The narrower we make the window, the better the time resolution, and better the assumption of stationary, but poorer the frequency resolution. However, the Wavelet transform solves the dilemma of resolution to a certain extent, as we will see in the next part.

3.3 Wavelets

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology. During the last few years wavelets is being used in data compression. The wavelets transformation is used to analyze the time and frequency content of an image and remove redundancy as well as areas that cannot be perceived by human eye. The human eye cannot perceive high frequency, as human ear cannot hear high frequency sounds.

3.3.1 Mother Wavelet

Wavelets constitute a family of functions derived from one single function and indexed by two labels one for position and one for frequency. This function is called the mother wavelet, denoted by ψ. Wavelets are defined as

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{3.1}$$

Where s≠ 0 is the scale factor, τ is the translation factor.

A Mother wavelet ψ should satisfy the following conditions:

$$\psi \in L^2(R) \tag{3.2}$$

$$\int_{-\infty}^{\infty} \frac{|\phi(\omega)|^2}{|\omega|} d\omega < +\infty \tag{3.3}$$

Where ϕ is the Fourier transformation of ψ defined by

$$\phi(\omega) = \int_{-\infty}^{\infty} \psi(t)e^{-j\omega t} dt \tag{3.4}$$

Equation (3.2) implies that the mean value of the wavelet ψ should be zero i.e.

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{3.5}$$

and hence ψ must be oscillatory i.e. ψ must be wave. Thus a wavelet is a small wave which has its energy concentrated in time. It has the ability to allow simultaneous time and frequency analysis.

3.3.2 Wavelet Transform

The basis idea of wavelet transform is to represent an arbitrary function f (x) as a linear combination of a set of wavelets or basis functions. These basis functions are obtained from a single prototype wavelet called Mother wavelet by dilations (scaling) and translations (shifts). The purpose of the wavelet transformation is to change the data from time-space domain to time-frequency domain, which makes better compression results.

There are two types of wavelet transform, Continuous wavelet transform and discrete wavelet transform.

3.3.2.1 Continuous wavelets transform

An important method for analyzing the time and frequency contents of a function f (t) by means of a wavelet is continuous wavelet transform. The transform is done by calculating the integral of the product of wavelet and f(t). The continuous wavelet transform of a function $f \in L^2(R)$ can be defined as

$$F(s,\tau) = \int f(t)\psi^*{}_{s,\tau}(t)dt \qquad (3.6)$$

Where ψ^* is the complex conjugate of ψ.

The inverse wavelet transform is defined by

$$f(t) = \int\int F(s,\tau)\psi_{s,\tau}(t)d\tau ds \qquad (3.7)$$

The continuous wavelet transform defined as above cannot be used directly due to the following reasons

1.The wavelet transform is calculated by continuously shifting a scalable function over a signal and calculating the correlation between the two. The obtained wavelet coefficients are highly redundant as these scaled functions will be nowhere near an orthogonal basis. This redundancy is not agreeable for practical applications.

2. The number of wavelets used in the transform is infinite and this number should be reduced.

3. For most functions the wavelet transform have no analytical solutions and they can be calculated only numerically or by an optical analog computer.

3.3.2.2 Discrete wavelets transform

To overcome the problems in the continuous wavelet transform, discrete wavelets have been introduced. Discrete wavelets are not continuously scalable and translatable but can only be scaled and translated in discrete steps.

Discrete wavelets are obtained by modifying equation (3.1) to

$$\psi_{j,k}(t) = s^{-\frac{j}{2}}\psi(s^{\frac{j}{2}}t - k\tau) \qquad (3.8)$$

where $s \neq 0$ is the scale factor, τ is the translation factor. The effect of discretizing the wavelet is that the time-scale space is now sampled at discrete intervals. We usually choose $s = 2$ and $\tau = 1$ so that the sampling of the frequency axis as well as time axis corresponds to dyadic sampling. Thus equation (3.8) becomes

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{\frac{j}{2}}t - k\tau) \qquad (3.9)$$

The generation of the discrete wavelets and the calculation of the discrete wavelet transform is well matched to the digital computer. The discrete wavelet transform is the discrete version of the continuous wavelet transform.

A wavelet is represented by means of several filter coefficients and the transform is carried out by matrix multiplication. Any function f (t) can be decomposed into a set of wavelets as

$$f(t) = \sum_{j,k} a_{j,k} \psi_{j,k}(t) \tag{3.10}$$

where the coefficients $a_{j,k}$ are calculated using the following formula.

$$a_{j,k} = \langle \psi_{j,k}, f(t) \rangle = \int f(t) \psi^*_{j,k}(t) dt \tag{3.11}$$

Equation (3.11) is the discrete wavelet transform of f(t). When discrete wavelets are used to transform a continuous signal, the result will be a series of wavelet coefficients. Equation (3.10) is the inverse discrete wavelets transform which shows that any arbitrary signal can be reconstructed by summing the orthogonal wavelet basis functions, weighted by the wavelet transform coefficients. The mother wavelet ψ should be so chosen such that the wavelets $\{\psi_{j,k}\}$ constitute an orthonormal basis of $L^2(R)$.

3.3.3 Father wavelet

Prior to the construction of ψ, one constructs a function ϕ such that the functions $\{\phi(t-k)\}, k \in Z$ constitute an orthonormal system. This orthonormal system can be supplemented to a full orthonormal basis of $L^2(R)$ with the functions $2^{-\frac{j}{2}} \psi(2^{\frac{j}{2}} t - k\tau), k \in Z, t \in Z_+$ for some mother wavelet ψ. This function ϕ is called the father of the wavelets or the scaling function.

3.3.4 Multi resolution analysis

A multiresolution analysis of $L^2(R)$ is a sequence of closed subspaces $\ldots, V_{-2}, V_{-1}, V_0, V_1, V_2, \ldots$ such that

1. $V_n \subset V_{n-1}, n \in Z.$
2. $\bigcup_{n=-\infty}^{\infty} V_n$ is dense in L^2 and $\bigcap_{n=-\infty}^{\infty} V_n = \{0\}$
3. $f(x) \in V_n \Leftrightarrow f(2x) \in V_{n-1}$
4. $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0$ For all $k \in Z$
5. $\exists$ g $\in V_0$ Such that the collection g (, -k), $k \in Z$ is a Riesz basis for V_0

This means that if a set of signals can be represented by a weighted sum of **g (t- k)** then a larger set including the original can be represented by a weighted sum of **g (2t - k)** .In other words, if the basic expansion signals are made half as wide and translated in steps half as wide, they will represent a larger class of signals exactly or give a better approximation of any signal.

3.3.5 How to construct Wavelets

In this section we will see how wavelets can be constructed from a scaling function ϕ. We define a set of scaling functions in terms of integer translates of ϕ by

$$\phi_{0n}(x) = \phi(x-n) \text{ for } x \in Z \;\&\; \phi \in L^2 \tag{3.12}$$

In wavelet theory it will be assumed that V_0 is generated by $\{\phi_{0n}\}$.

Since $\phi \in V_0$ and $V_0 \subset V_{-1}$ we have $\phi \in V_{-1}$. But property (3) of multiresolution analysis implies that $\phi(2^{-1}) \in V_0$ thus

$$\phi(\frac{x}{2}) = \sum_{n=-\infty}^{\infty} C_n \phi(x-n), x \in R \tag{3.13}$$

for some coefficients $\{C_n\}$. This is called the dilation equation. It can be rewritten as

$$\phi(x) = \sqrt{2} \sum_{n=-\infty}^{\infty} h_n \phi(2x-n) \tag{3.14}$$

The coefficients h_n are called the filter coefficients of the function ϕ, which satisfy the following prosperities

$$\sum_{n=-\infty}^{\infty} h_n = \sqrt{2} \tag{3.15}$$

$$\sum_{n=-\infty}^{\infty} (-1)^n h_n = 0 \tag{3.16}$$

The associated mother wavelet ψ is then generated through ϕ by the definition

$$\psi(x) = \sqrt{2} \sum_{n=-\infty}^{\infty} g_n \phi(2x-n), g_n = (-1)^n h_{1-n} \tag{3.17}$$

and the associated wavelets are

$$\psi_{m,n}(x) = 2^{\frac{-m}{2}} \psi(2^{-m}x-n), m,n \in Z \tag{3.18}$$

Wavelets are generally expressed by means of their filter coefficients h_n.

3.4 Examples of Wavelets

1. Haar Wavelets

$$h_0 = \frac{1}{\sqrt{2}}$$

$$h_1 = \frac{1}{\sqrt{2}}$$

2. Daubechies Wavelets

D4:

$$h_0 = \frac{\sqrt{2}}{8}(1+\sqrt{3})$$

$$h_1 = \frac{\sqrt{2}}{8}(3+\sqrt{3})$$

$$h_2 = \frac{\sqrt{2}}{8}(3-\sqrt{3})$$

$$h_3 = \frac{\sqrt{2}}{8}(1-\sqrt{3})$$

D6:

$$h_0 = \frac{\sqrt{2}}{32}(b+c)$$

$$h_1 = \frac{\sqrt{2}}{32}(2a+3b+3c)$$

$$h_2 = \frac{\sqrt{2}}{32}(6a+4b+2c)$$

$$h_3 = \frac{\sqrt{2}}{32}(6a+4b-2c)$$

$$h_4 = \frac{\sqrt{2}}{32}(2a+3b-3c)$$

$$h_5 = \frac{\sqrt{2}}{32}(b-c)$$

Where $a = 1-\sqrt{10}, b = 1+\sqrt{10}, c = \sqrt{5+2\sqrt{10}}$

More Daubechies wavelets can be found in [4].

More information about wavelets can be found in [2].

3.5 Wavelets Chosen for this Project

The ideal wavelet would be compact, orthogonal, symmetric and continuous. The first sym-metrical and orthogonal wavelet discovered was a simple square wave, which is not continuous, and it is known as the Haar wavelet.

Scaling function for Haar Wavelet

$$\phi(x) = \begin{cases} 1 & if\ 0 \le x \le 1 \\ 0 & otherwise \end{cases}$$

Wavelet function for Haar Wavelet

$$\psi(x) = \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Wavelet Coefficients are $h_0 = h_1 = \frac{1}{\sqrt{2}}$

3.6 Why wavelets for Compression

Digital data, i.e., data from speech, images, video and graphics, are highly correlated, and contain redundancy. Wavelet exploits this structure by providing a system by which this type of digital data can be represented accurately with a few parameters. Moreover, the computation involved in obtaining the representation is fast and coefficients, usually linear in complexity. Because of this property, wavelet has been used in image compression, video compression, datà compression, data transmission, geometric modeling, and as well as in numerical computations.

Chapter 4

4.1 Proposed Algorithm for Gray Level Image Compression

Algorithm G: Compression Algorithm for Grey Level Image

1. Decompose the Image into N-levels using Discrete Wavelets Transformation, where N is the no of times the Discrete Wavelets Transform is to be used.
2. Remove the $\mathbf{HH_1}$ portion.
3. Compress the $\mathbf{LL}_N$ portions losslessly.
4. Compress the other portions Lossyly.

Implementation Detail:

Step 1: Here Haar wavelet is used for Discrete Wavelet Transform (DWT).

$$\text{Analysis Filter: } h_0 = h_1 = \frac{1}{\sqrt{2}}$$

$$\text{Synthesis Filter: } g_0 = \frac{1}{\sqrt{2}}, g_1 = -\frac{1}{\sqrt{2}}$$

Let me first explain the 1D DWT of a Signal.

E.g.: The 1D DWTof the signal {162,162,159,161,162,159,159,157} three times.

1st Level:

$$\frac{(162+162)}{\sqrt{2}}, \frac{(159+161)}{\sqrt{2}}, \frac{(162+159)}{\sqrt{2}}, \frac{(159+157)}{\sqrt{2}}, \frac{(162-162)}{\sqrt{2}}, \frac{(159-161)}{\sqrt{2}}, \frac{(162-159)}{\sqrt{2}}, \frac{(159-157)}{\sqrt{2}}$$

i.e. 229.10,226.27,226.98,223.45,0.00,-1.41,2.12,1.41

2nd Level:

$$\frac{(229.10+226.27)}{\sqrt{2}}, \frac{(226.98+223.45)}{\sqrt{2}}, \frac{(229.10-226.27)}{\sqrt{2}}, \frac{(226.98-223.45)}{\sqrt{2}}, 0.00, -1.41, 2.12, 1.41$$

i.e. 322.00,318.50,2.00,2.50,0.00,-1.41,2.12,1.41

3rd Level: $\frac{(322.00+318.50)}{\sqrt{2}}, \frac{(322.00-318.50)}{\sqrt{2}}, 2.00, 2.50, 0.00, -1.41, 2.12, 1.41$

i.e. 452.90,2.47,2.00,2.50,0.00,-1.41,2.12,1.41

First of all apply the 1D Discrete Wavelet Transform to each row of the image. Then apply the 1D Discrete Wavelet Transform to each column of the image. Which completes the one-level DWT of the Image

One-level Discrete Wavelet Transform decomposes an image into four-part LL, LH, HL, and HH.

LL	HL
LH	HH

1. LL: horizontal low-pass/vertical low-pass

LL portion gives the coarse scale information.

2. HL: horizontal high-pass/vertical low-pass

HL portion gives information about the vertical edges.

3. LH: horizontal low-pass/vertical high-pass

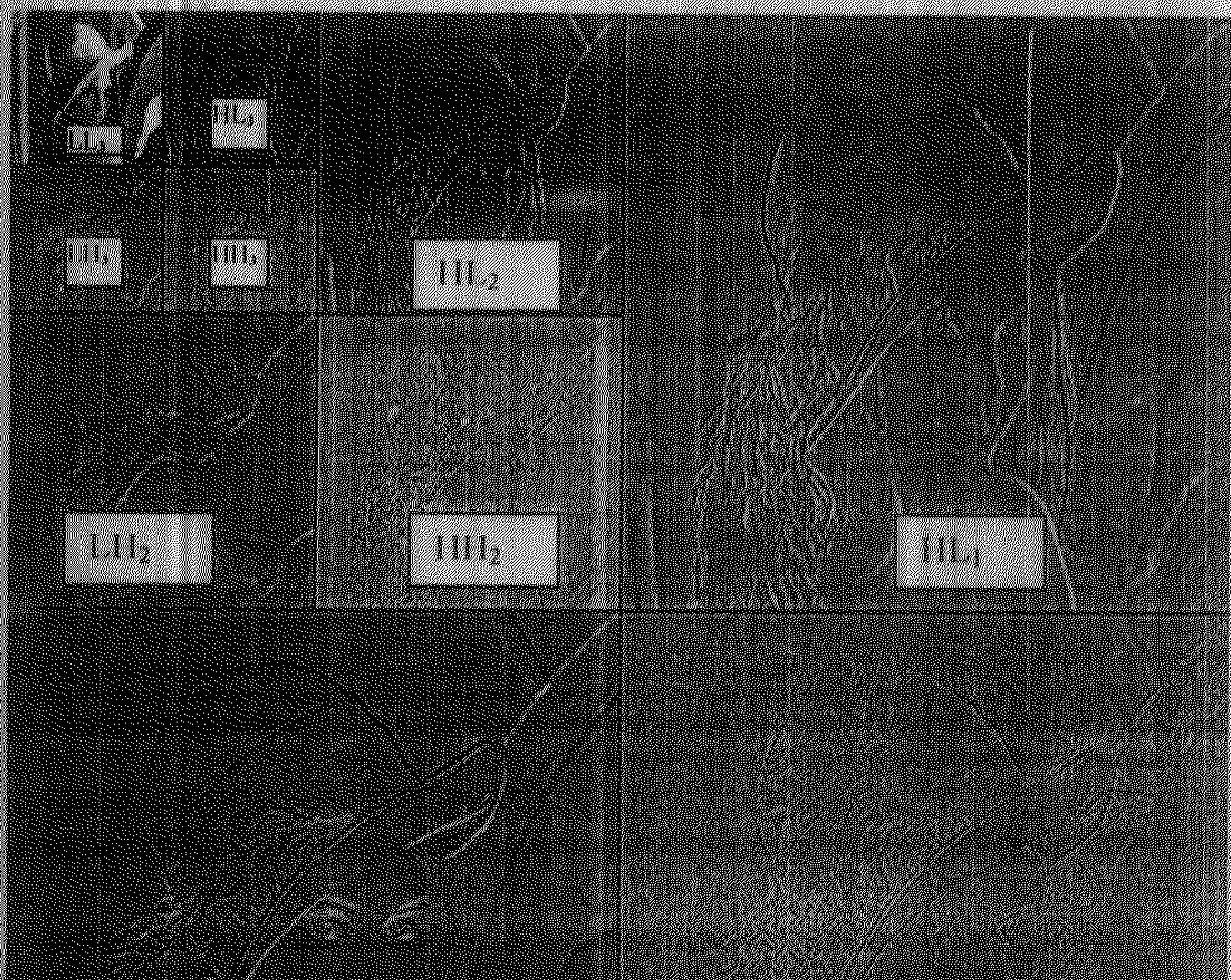
LH portion gives information about the horizontal edges.

4. HH: horizontal high-pass/vertical high-pass

HH portion gives information about the corner points.

Then apply Discrete Wavelet Transform to the LL portion of the decomposed image. Repeat these process N times, where N is the no of times you want to decompose the Image.

E.g.: Three-level decomposition of Gray Level Lena Image. (N=3)



Step 2: The HH_1 portion of the transformed image contains information about the corner points and the diagonal edges. Majority of the information is present in the other portions. So, generally the loss incurred by removing HH_1 portion is negligible and not discernable to the naked eye. Thus HH_1 portion of the image is removed.

Step 3: The left topmost LL_N portion is coded losslessly. For Lossless compression first of all use Modified Huffman Coding and then the Modified Run Length Coding.

Modified Huffman Coding:

Modified Huffman Coding is a standard variable length coding technique where the input data is encoded by a data structure having two fields (range, level). Range x denotes that the input data is an element of the set S containing 2^x elements and level *l* indicates the position of an element of the set. For this the following Code Book is used.

Range	Value	Code for Range
0	0	0
1	-1, 1	10
2	-3, -2, 2, 3	110
3	-7, -6, -5, -4,4,5,6,7	1110
4	-15, -14...-8,8,..., 15	11110
5	-31, ..., -16,16,..., 31	111110
6	-63,..., -32,32...63	
:	:	
15	-32767,..., -16384,16384,..., 32767	1111111111111110
16	32768	1111111111111111

To find the code of a value V first find the range r of V then find the level *l* of V.

Code of V = Code of r followed by binary representation of *l*.

E.G: Suppose the encoder finds −14. We can see that range r of −14 is 4 and −14 is the 1st element (Considering −15 as the 0th element). So level *l* of −14 is 1. Again the number of bit required to code level of an element in the range r is r. So binary representation of level of −14 i.e. 1 using 4 bits is 0001 and the code of range r=4 is 11110.Hence the code of −14 is 111100001. This type of coding is dependent on the fact that the probability of getting a coefficient in the range 0,1,2 is very high. In this way the total number of bits is reduced for the whole image.

Modified Run Length Coding:

Modified run length coding is performed on the Huffman coding data. It is implemented using two counts one count and zero count. One count is used to count the run of ones and zero count is used to count the run of zeroes. This counting will stop if a number, different from the number in run, is found. This run count is sent in binary form, using the relevant number of bits required. It is done using the following table.

Range of run count	Number of Bits
1	1
2-5	2
6-13	3
14-29	4
30-61	5
Else	6

E.g.: Let the run count be 16. From the table we can see that it is third element in the range 14-31. Hence it is run length coded using 4 bits i.e. 0010. Similarly 61 is coded as 11111.

Step 4: For each of the remaining portions of the decomposed image apply Lossy Compression. Step of Lossy Compression are

1. Divide each portion into 8x8 non-overlapping blocks.
2. Quantize each element of the blocks.
3. Apply Modified Huffman Coding to the each of the quantized elements.
4. Apply Modified Run Length Coding to the Huffman Coded data.

Quantization:

This is the step where the information loss occurs, but allows for large compression ratios. Each element of the block is divided by the value of a quantization matrix and the result is rounded to the nearest integer. Here the quantization matrix is taken as

$$\left[Q_{i,j}\right] = 1 + (i + j) * P$$ Where **P** is a constant.

There is a tradeoff between image quality and degree of quantization. A large quantization step size can produce unacceptable large image distortion. After quantization most of the values of the block becomes zero. Then the values of the block are coded using Modified Huffman Coding and after that Modified Run Length Coding is applied to the Huffman coded data.

4.2 Proposed Algorithm for Color Image Compression

Algorithm C: Compression Algorithm for Color Image

1. Convert the Color Image from RGB Color Space to YCrCb Color Space.
2. Sub sample Cr and Cb Space.
3. Compress each of the components Y, Cr & Cb using Algorithm G independently.

Step 1: Here the following transformation equations are used for the conversion from RGB color space to YCrCb color space.

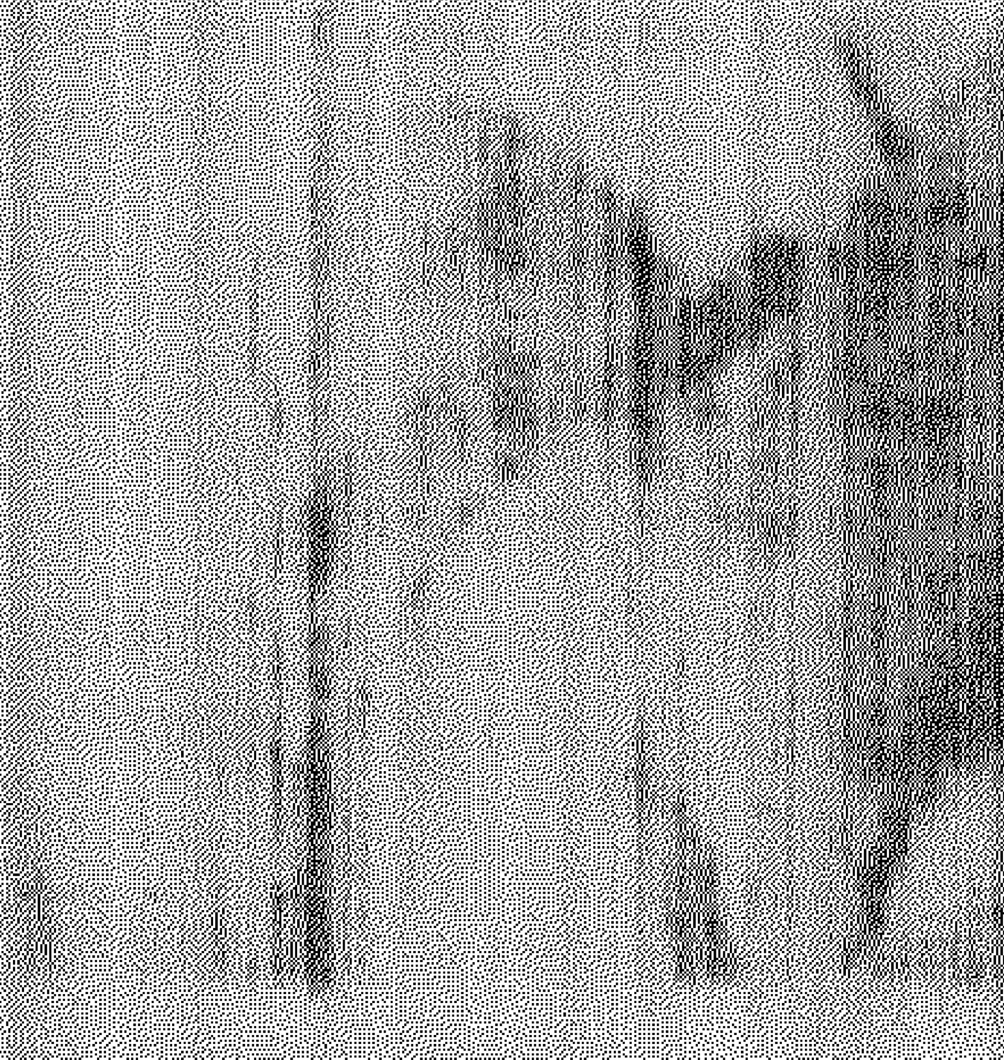
$$Y = (77/256)\,R + (150/256)\,G + (29/256)\,B$$

$$Cr = (131/256)\,R - (110/256)\,G - (21/256)\,B + 128$$

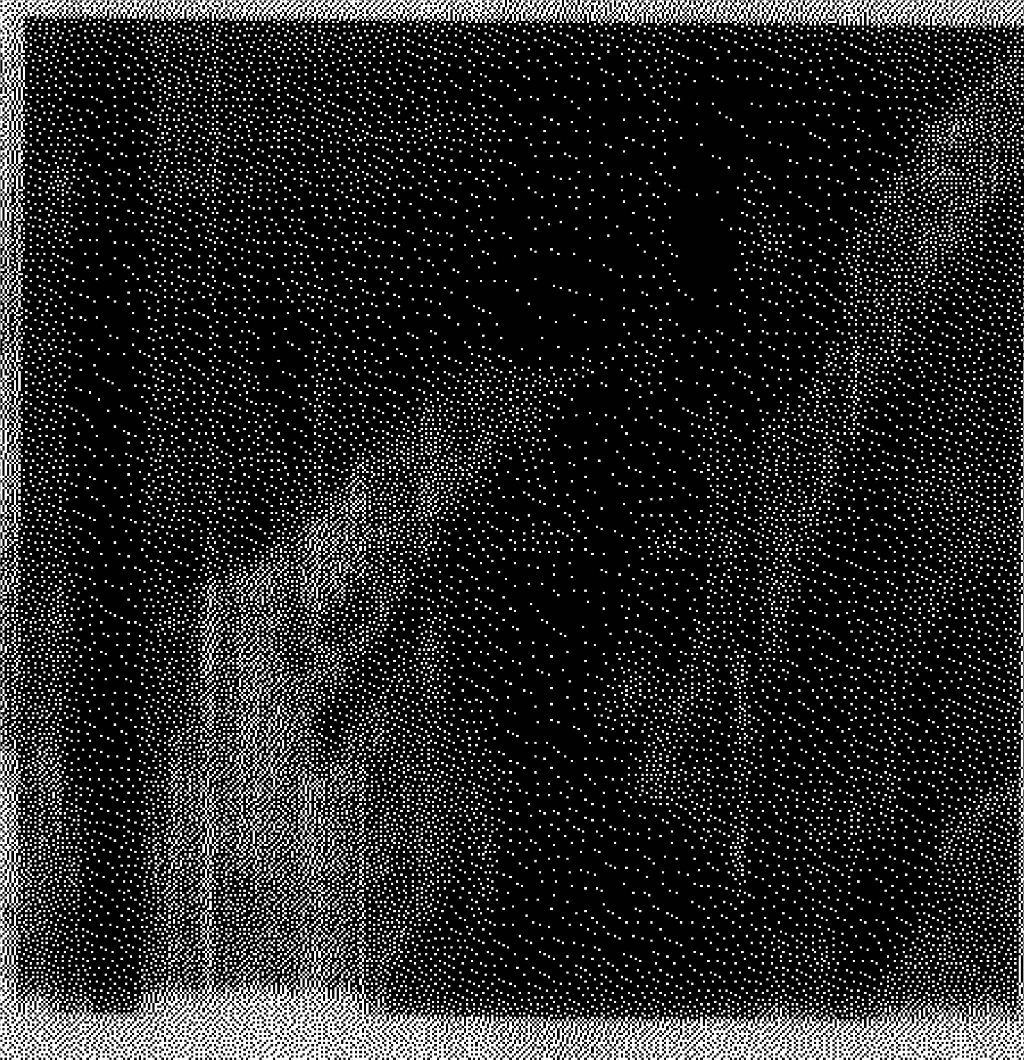
$$Cb = -(44/256)\,R - (87/256)\,G + (131/256)\,B + 128$$

Original Color Image

Y Component



Cr Component



Cb Component

Step 2: Since Human Visual System is more sensitive to the change of intensity than the color and as Luminance is controlled by the intensity of the signal and Chrominance is the color signal, so sub sampling of the chrominance spaces can be done to achieve better compression. Here 4:2:0 sample pattern is used.

4:2:0

Y	Y		Cr		Cb
Y	Y				

Step 3: Since there is no correlation among the spaces of YCrCb, So each of Y, Cr and Cb component is compressed independently. Apply **Algorithm G** to compress each of them.

Compression Ratio

Compression Ratio is defined as the ratio of the size of the Image before compression and the size of the image after compression. If image size is N then
In case of Gray Level Image Compression Ratio is defined as

$$CR_{GRAY} = \frac{NxNx8}{Total\ No\ of\ Bits\ After\ Compression}$$

In case of Color Image Compression Ratio is defined as

$$CR_{COLOR} = \frac{NxNx3x8}{Total\ No\ of\ Bits\ After\ Compression}$$

PSNR

PSNR is the acronym of **peak signal to noise ration.** PSNR is defined as
In Case of Gray Level Image of size N

$$PSNR_{GRAY} = 10\log_{10}\frac{255x255}{MSE}$$

Where
$$MSE = \frac{1}{NxN}\sum_{i,j=1}^{N}[f(i,j) - f^{C}(i,j)]^2$$

Where f is the original image and f^{C} is the reconstructed image.
In case of Color Image of size N

$$PSNR_{COLOR} = 10\log_{10}\frac{3x255x255}{MSE(R) + MSE(G) + MSE(B)}$$

The unit of measurement of PSNR is ***decibels*** (dB)

Chapter 5
Results

The Proposed Algorithm C is tested for Lena Color Image (512x512) and Fruits Color Image (512x512) and Pepper color Image (512x512) using Six-Level Decomposition. The results are given below.

Original Lena Image (512x512)

Result for Lena Image (512x512)

Level of Decomposition	P (Quantization Constant)	PSNR	Compression Ratio
6	1	41.84	28.55:1
6	2	39.96	45.59:1
6	3	38.71	58.76:1
6	5	37.25	77.59:1
6	7	36.39	88.27:1
6	9	35.90	95.47:1
6	11	35.51	99.58:1
6	13	35.20	102.10:1

PSNR=41.48 CR=28.55:1

PSNR=39.96 CR=45.59:1

PSNR=38.71 CR=58.76:1

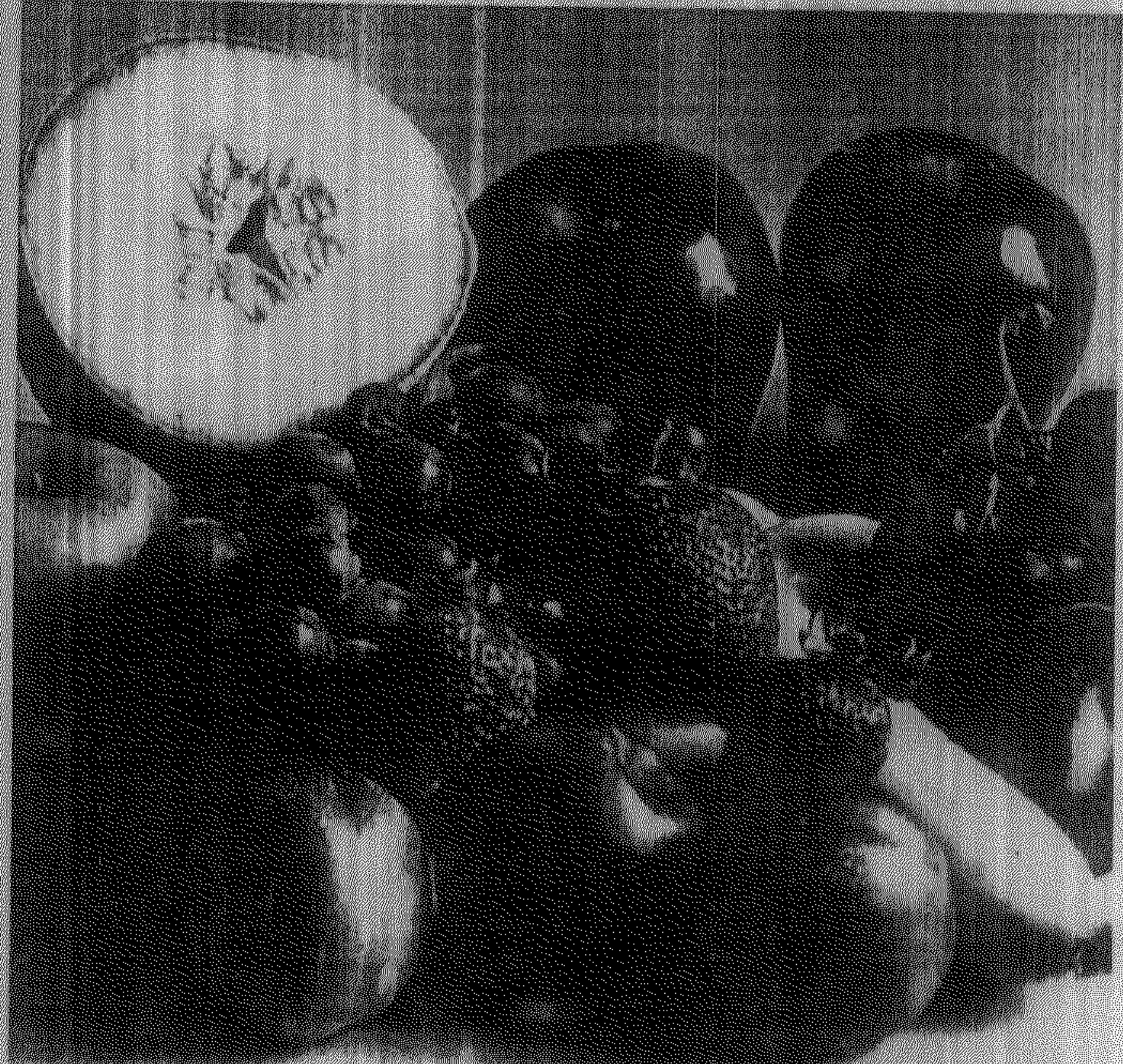
PSNR=37.25 CR=77.59:1

PSNR=36.39 CR=88.27:1

PSNR=35.90 CR=95.47:1

PSNR=35.51 CR=99.58:1

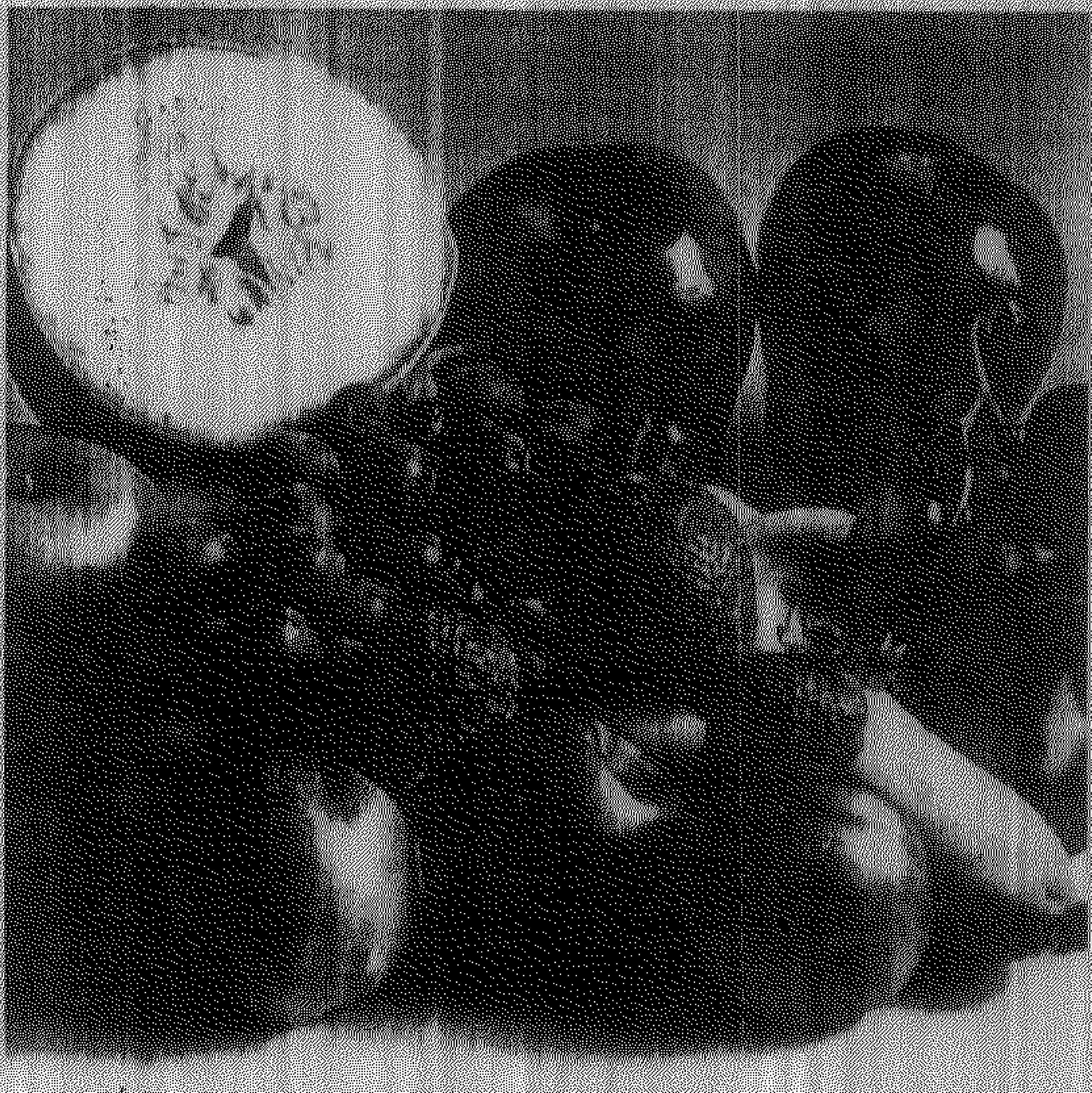
PSNR=35.20 CR=102.10:1



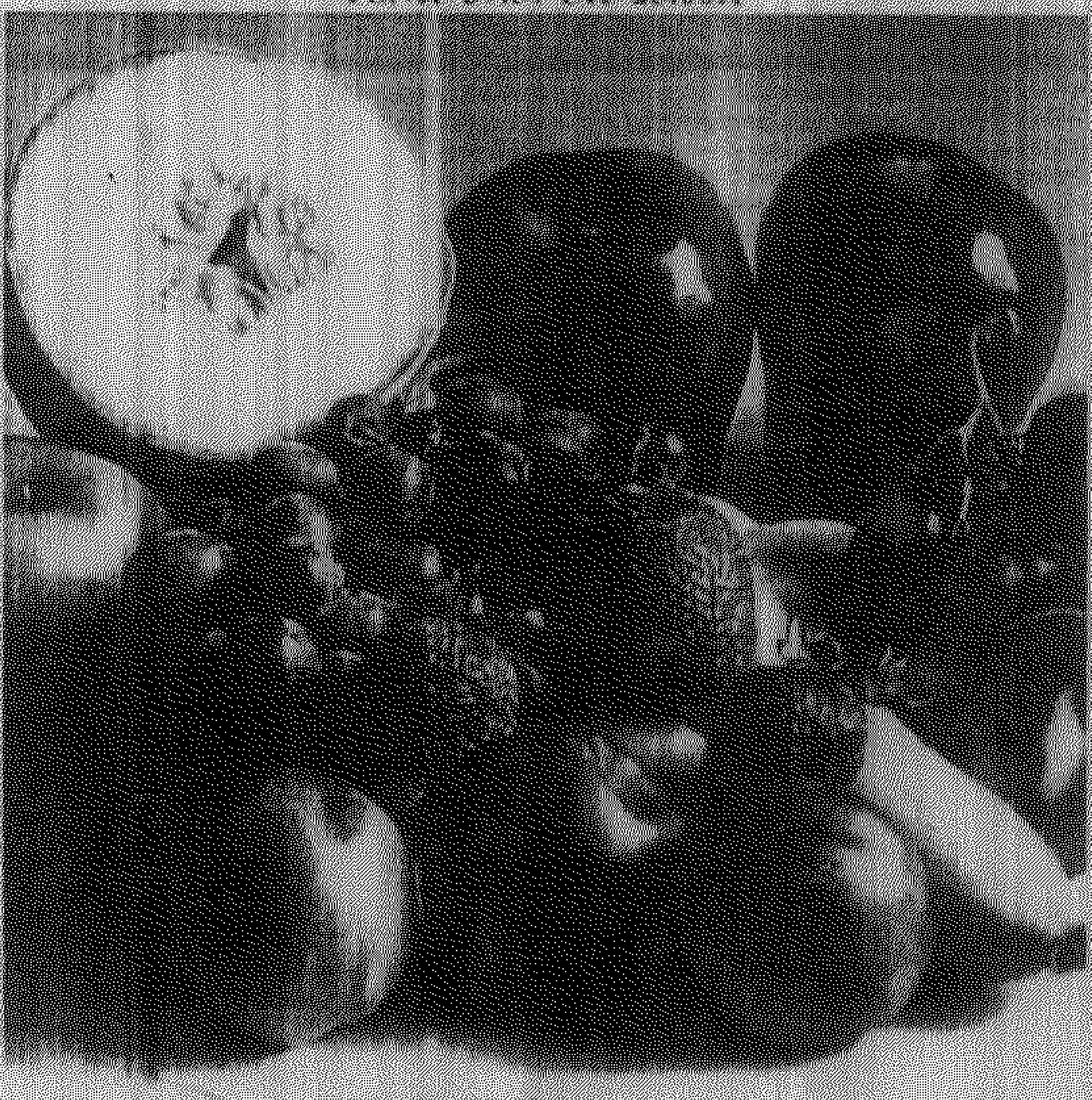
Original Fruits Image (512x512)

Result for Fruits Image (512x512)

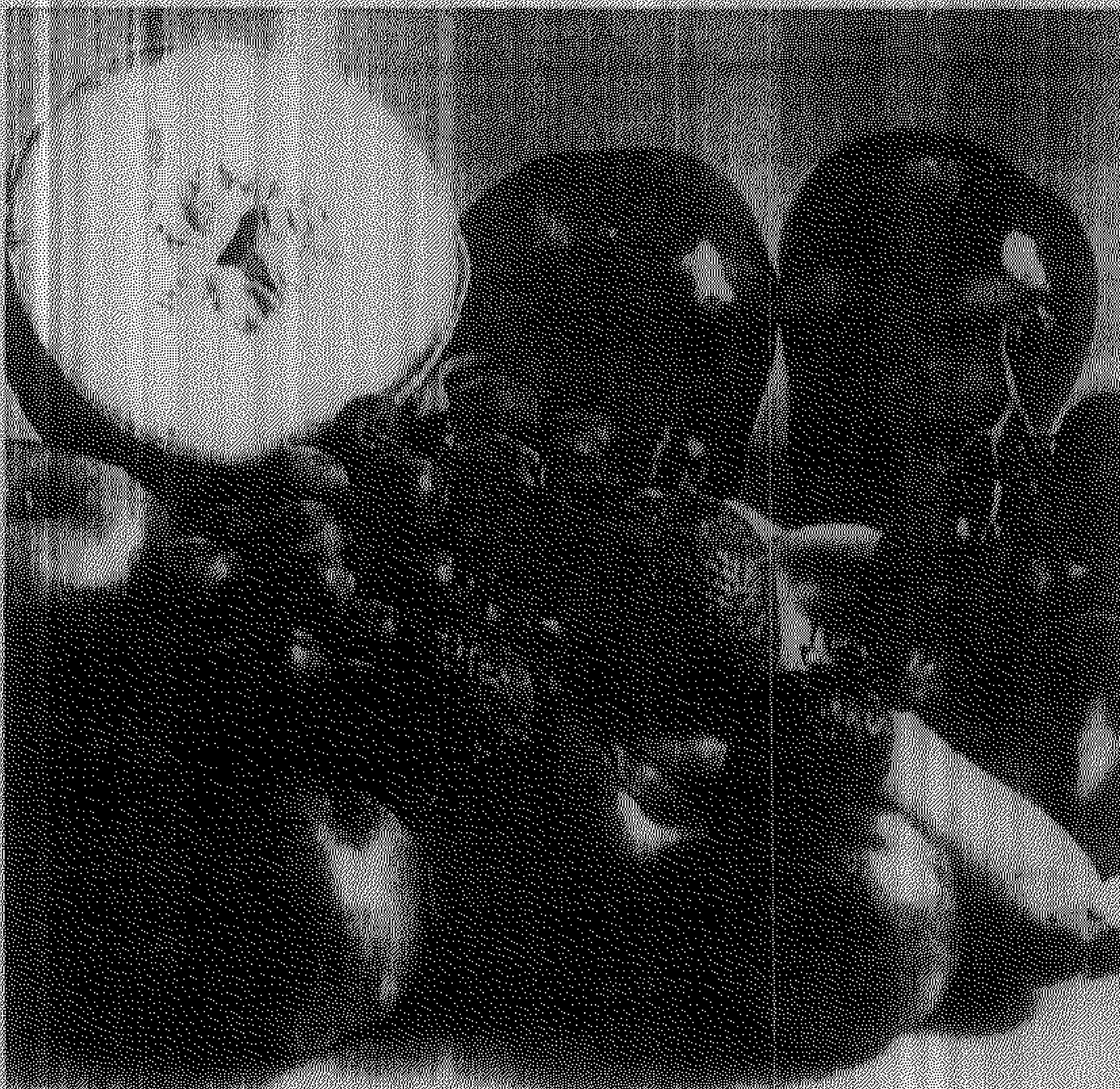
Level of Decomposition	P (Quantization Constant)	PSNR	Compression Ratio
6	1	39.57	25.00:1
6	2	38.05	41.22:1
6	3	37.11	54.40:1
6	5	36.08	73.26:1
6	7	35.43	85.02:1
6	9	34.93	92.48:1
6	11	34.58	97.06:1
6	13	34.39	99.90:1



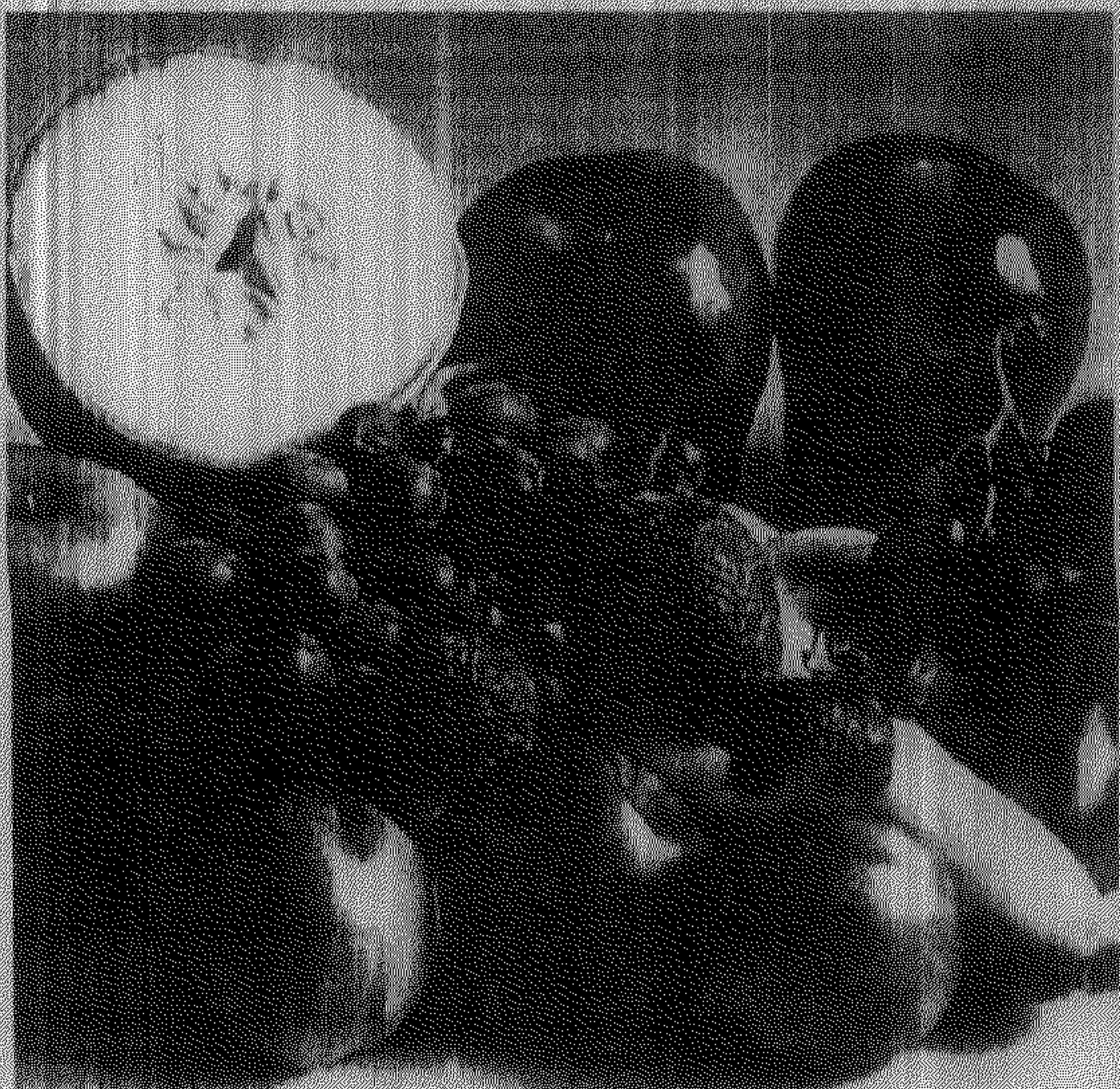
PSNR=39.57 CR=25.00:1



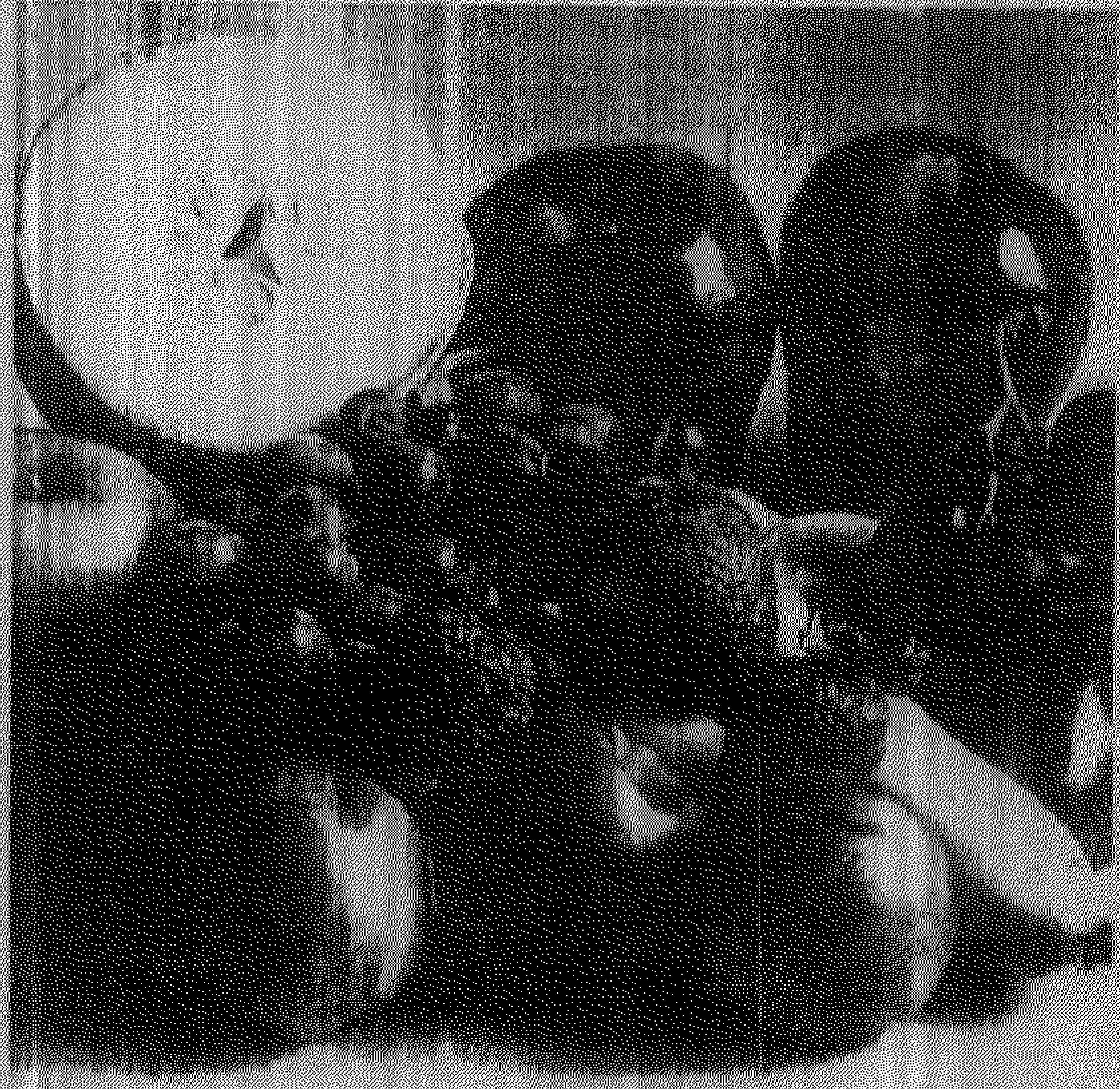
PSNR=38.05 CR=41.22:1



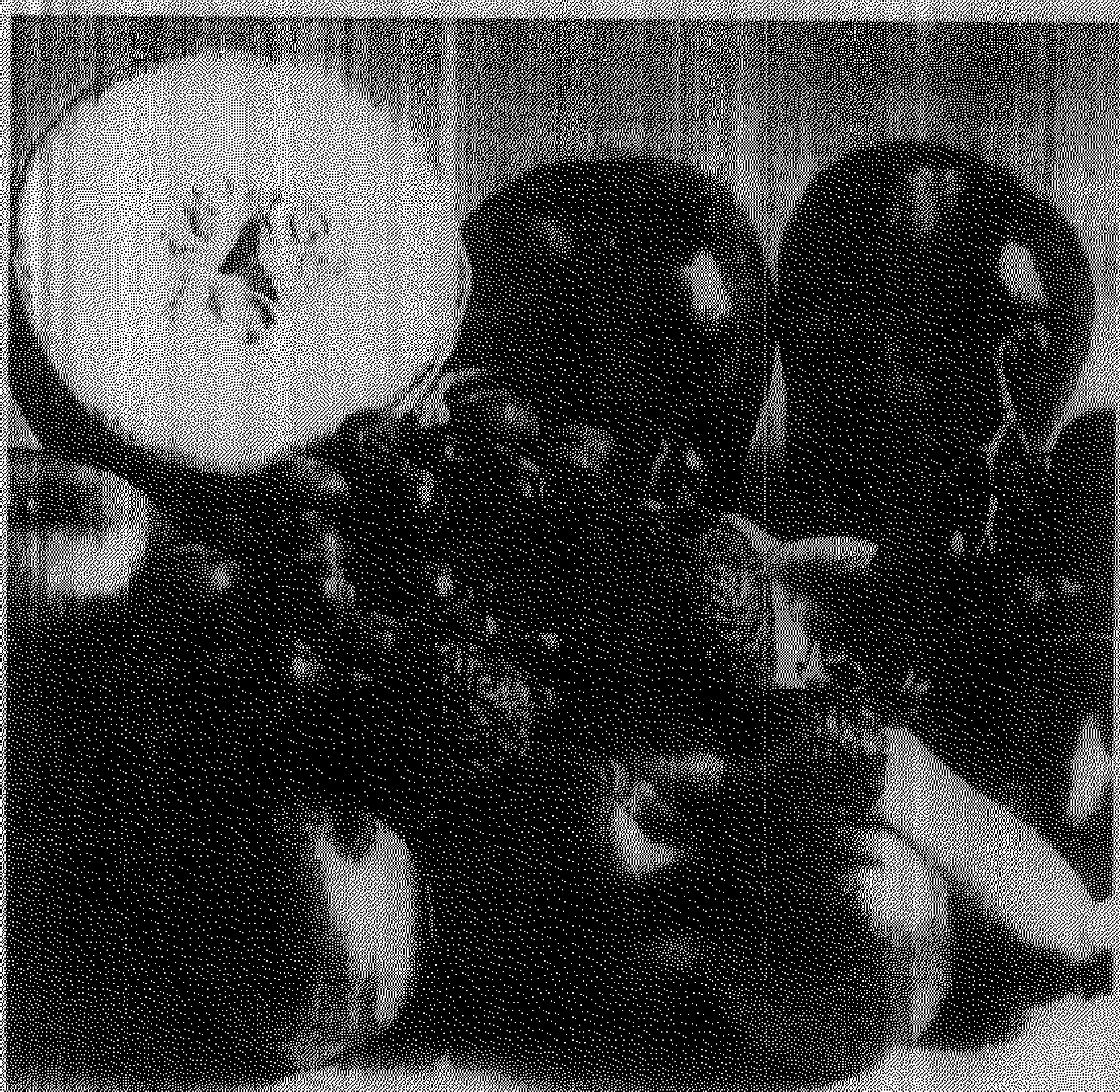
PSNR=37.11 CR=54.40:1



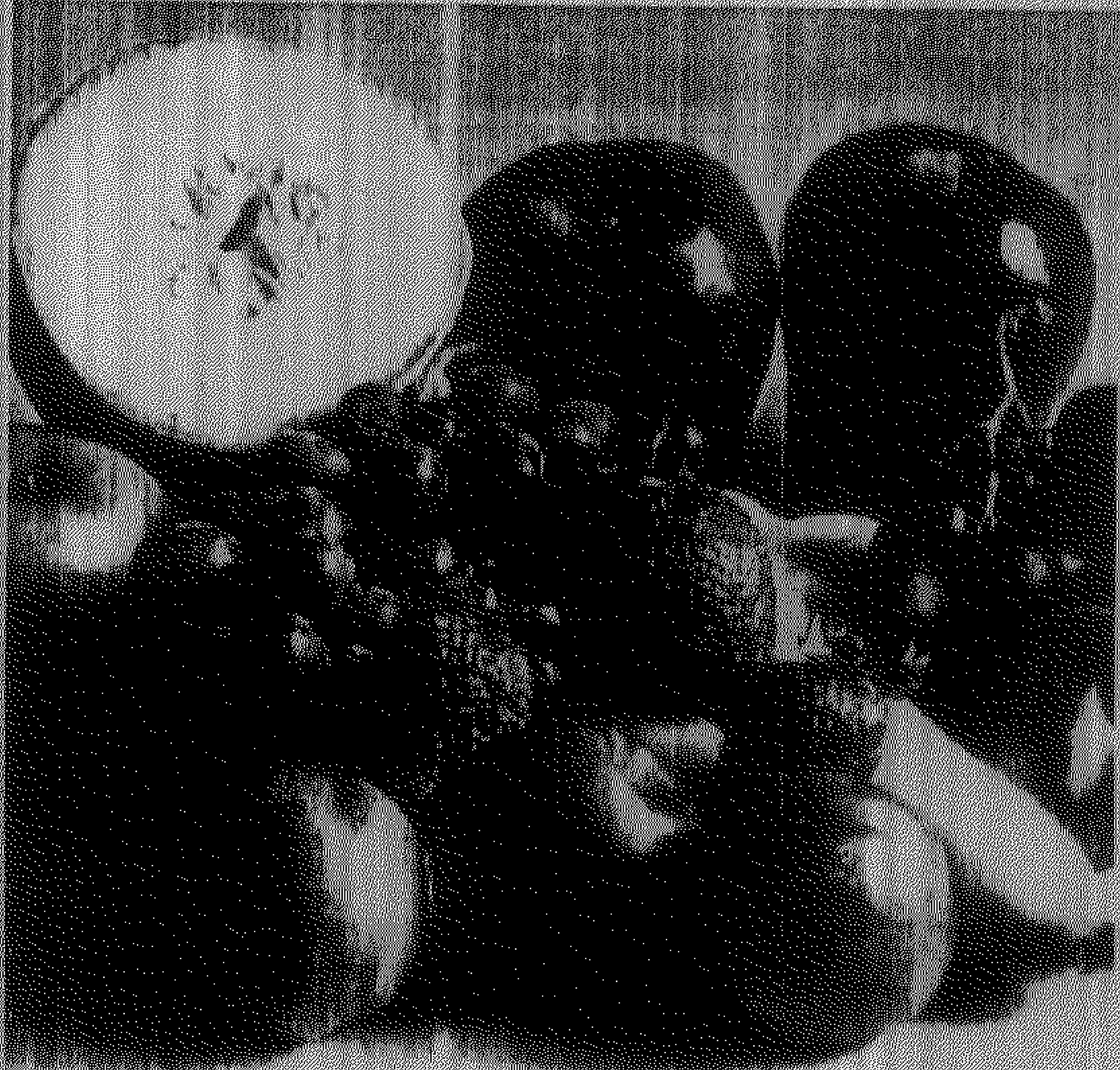
PSNR=36.08 CR=73.26:1



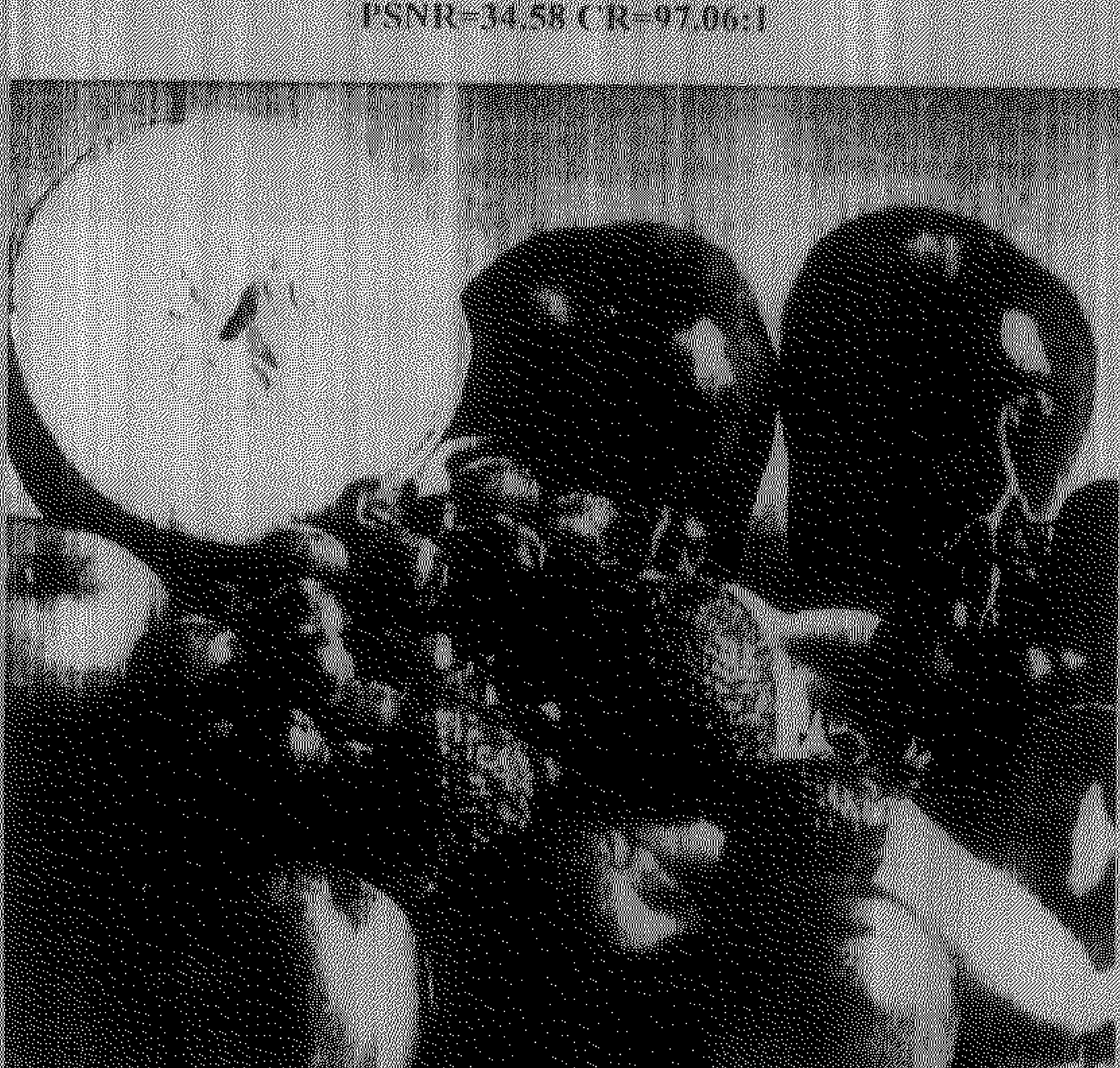
PSNR=35.43 CR=85.02:1



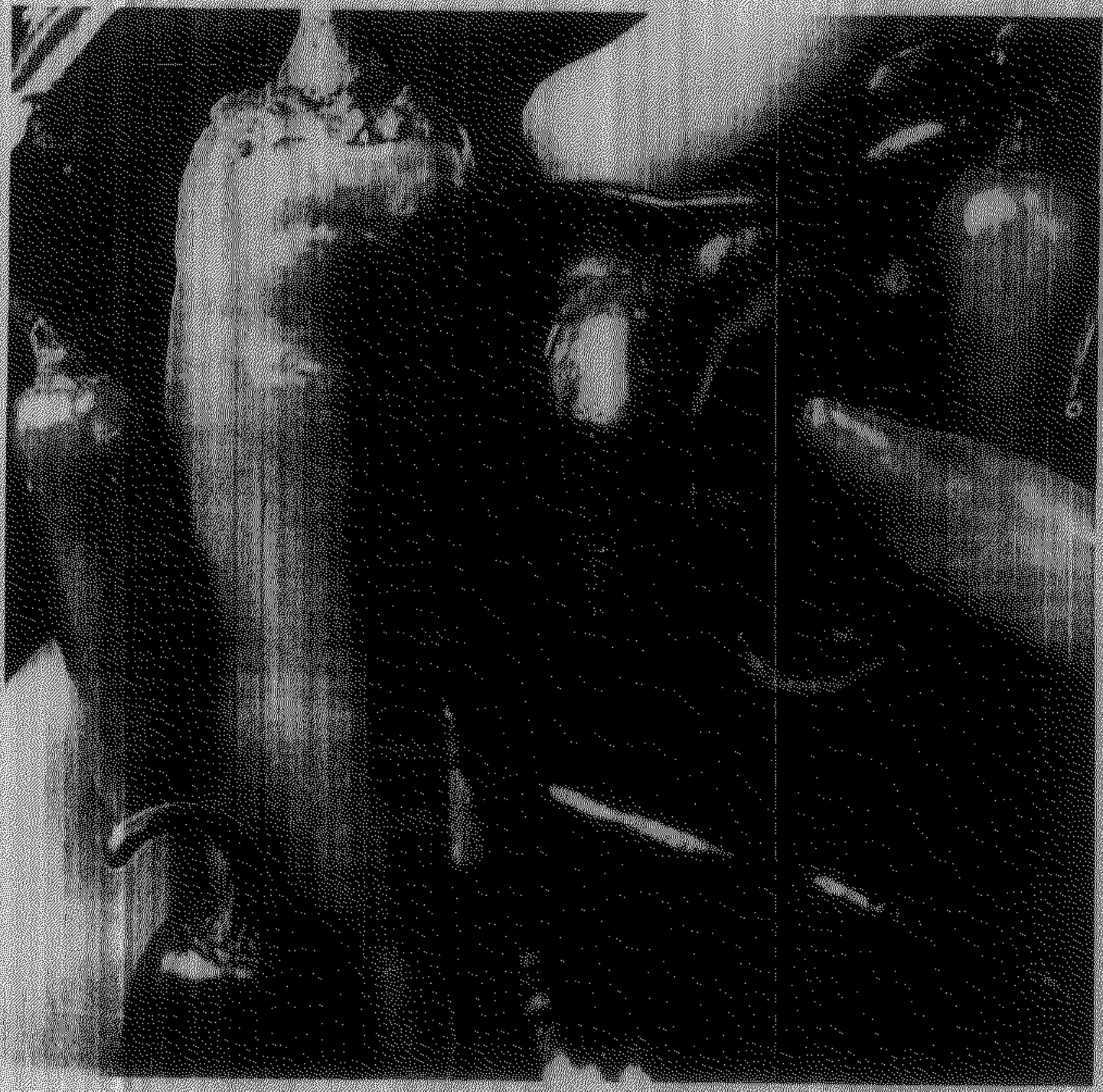
PSNR=34.93 CR=92.48:1



PSNR=34.58 CR=97.06:1



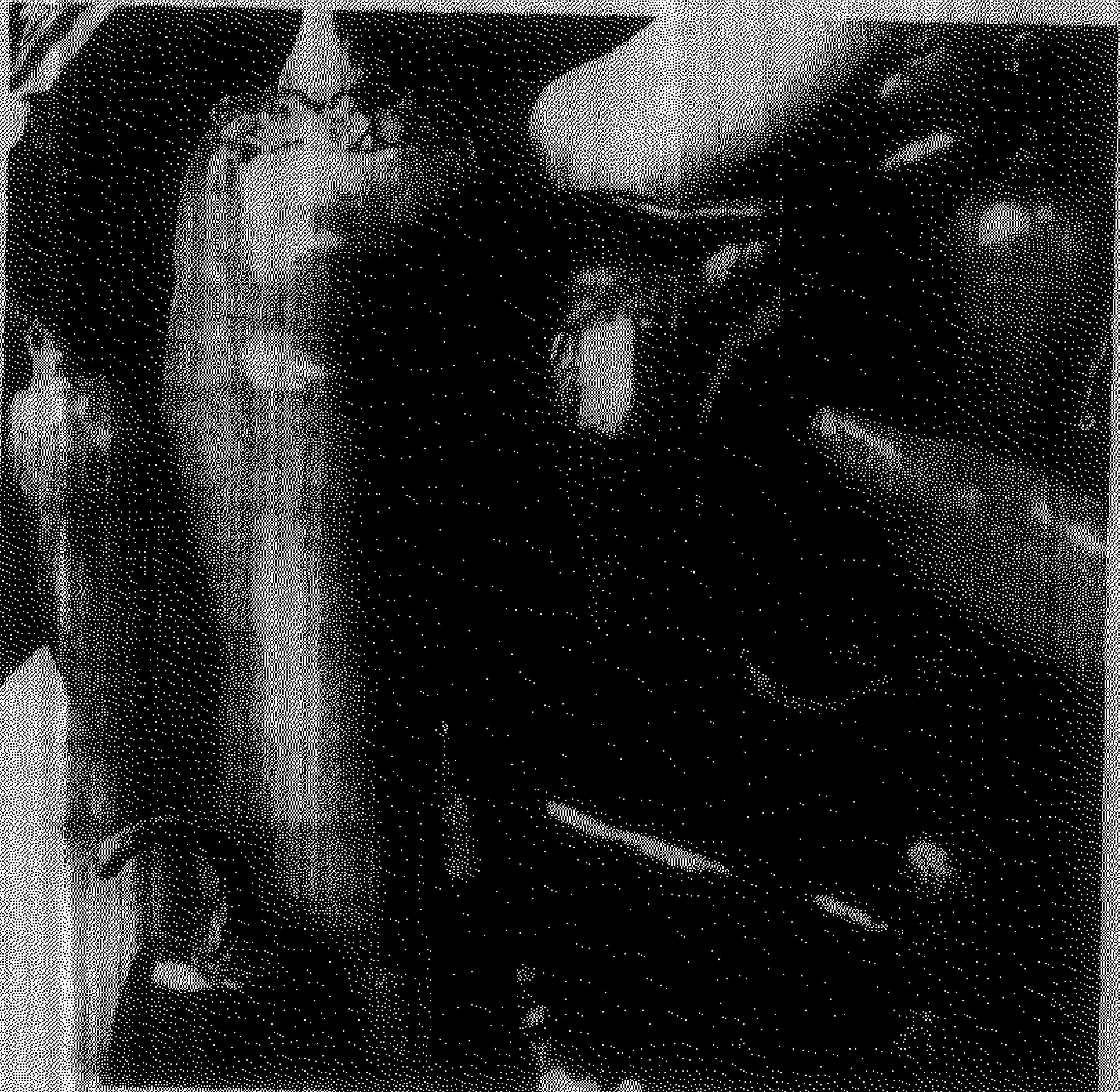
PSNR=35.51 CR=99.90:1



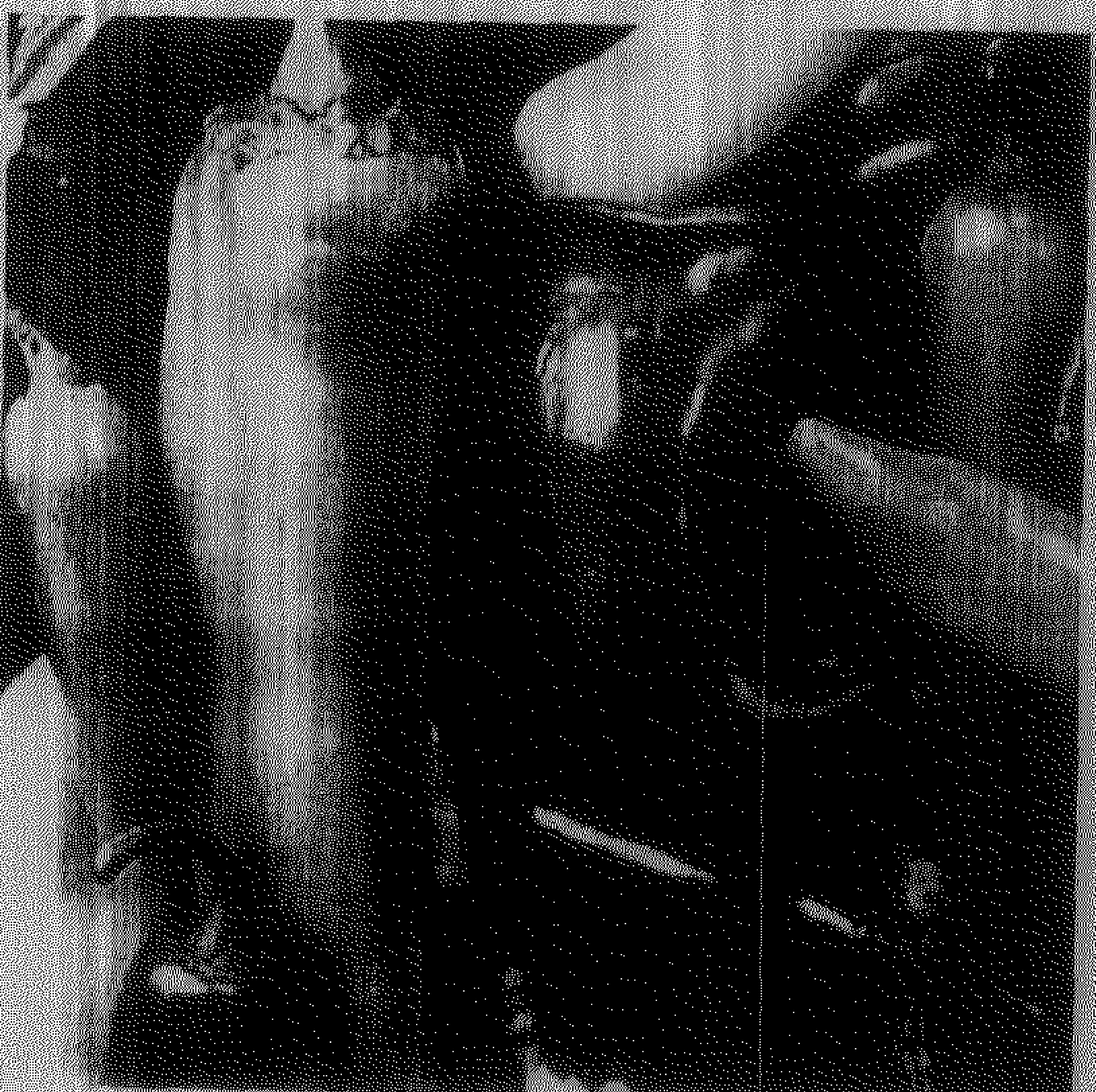
Original Pepper Image (512x512)

Result for Pepper Image (512x512)

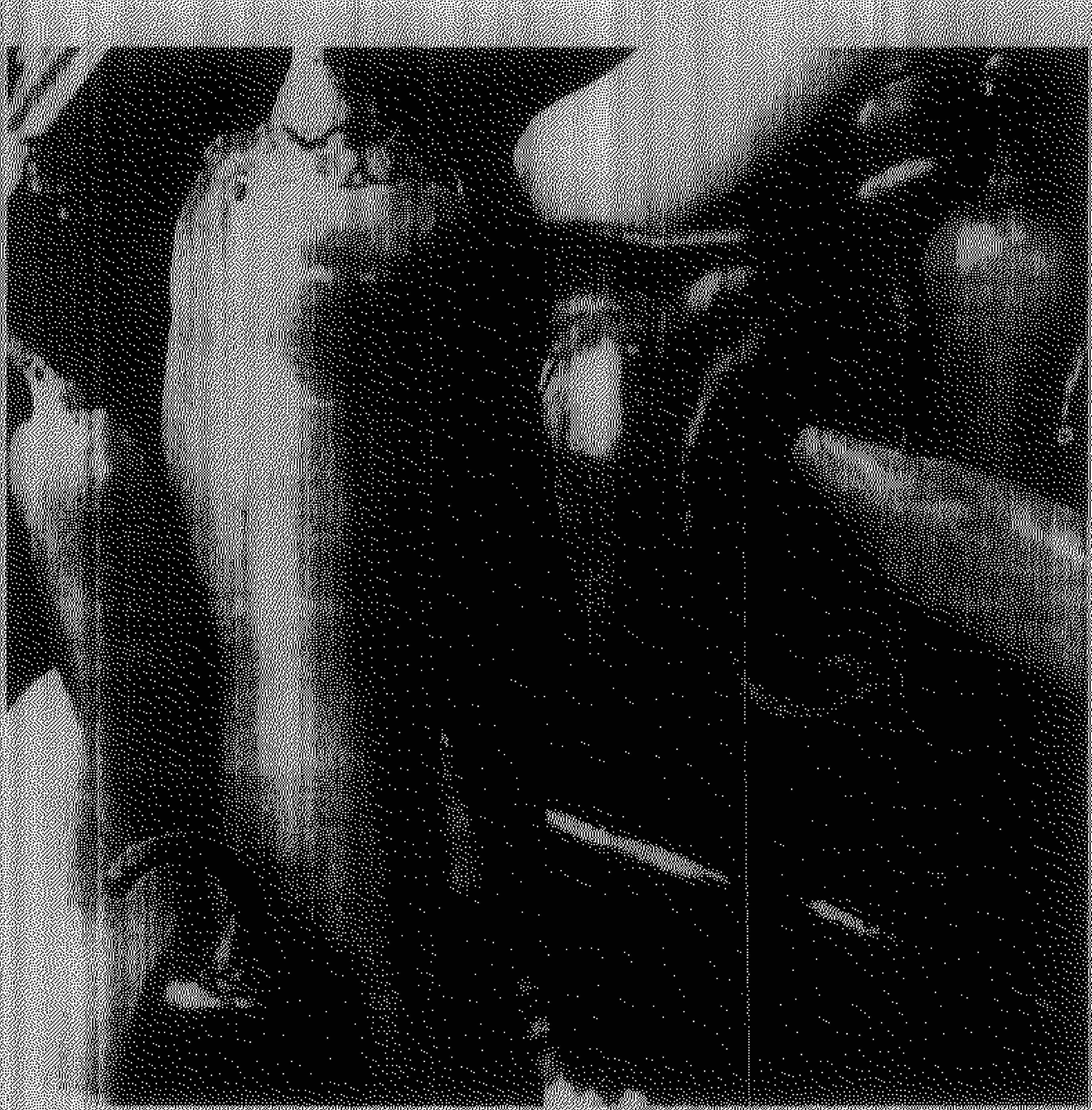
Level of Decomposition	P (Quantization Constant)	PSNR	Compression Ratio
6	1	39.26	32.48:1
6	2	38.19	51.50:1
6	3	37.43	65.18:1
6	5	36.51	81.27:1
6	7	35.85	89.94:1
6	9	35.40	95.10:1
6	11	34.96	98.21:1
6	13	34.68	100.43:1



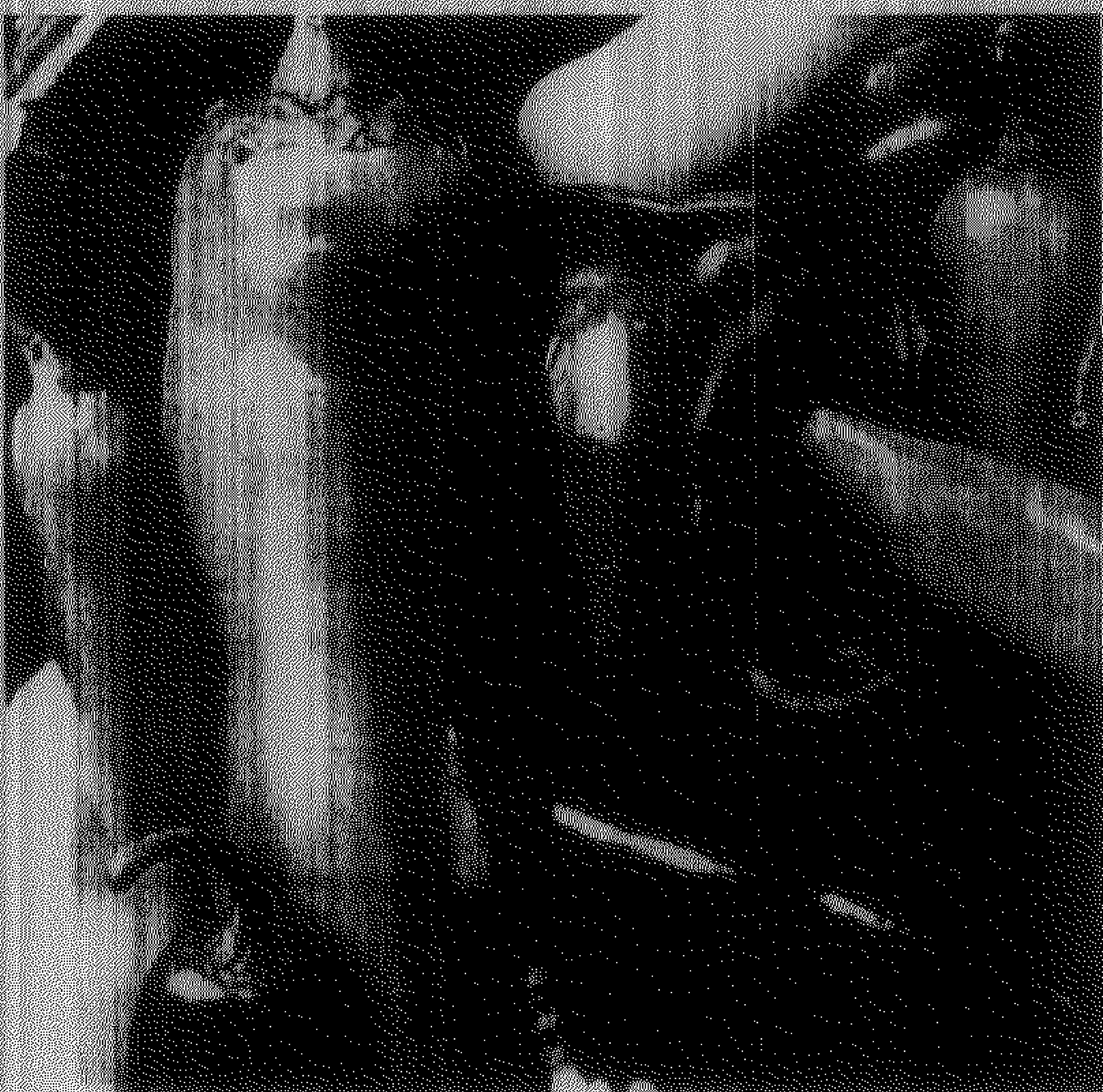
PSNR=39.26 CR=32.48:1



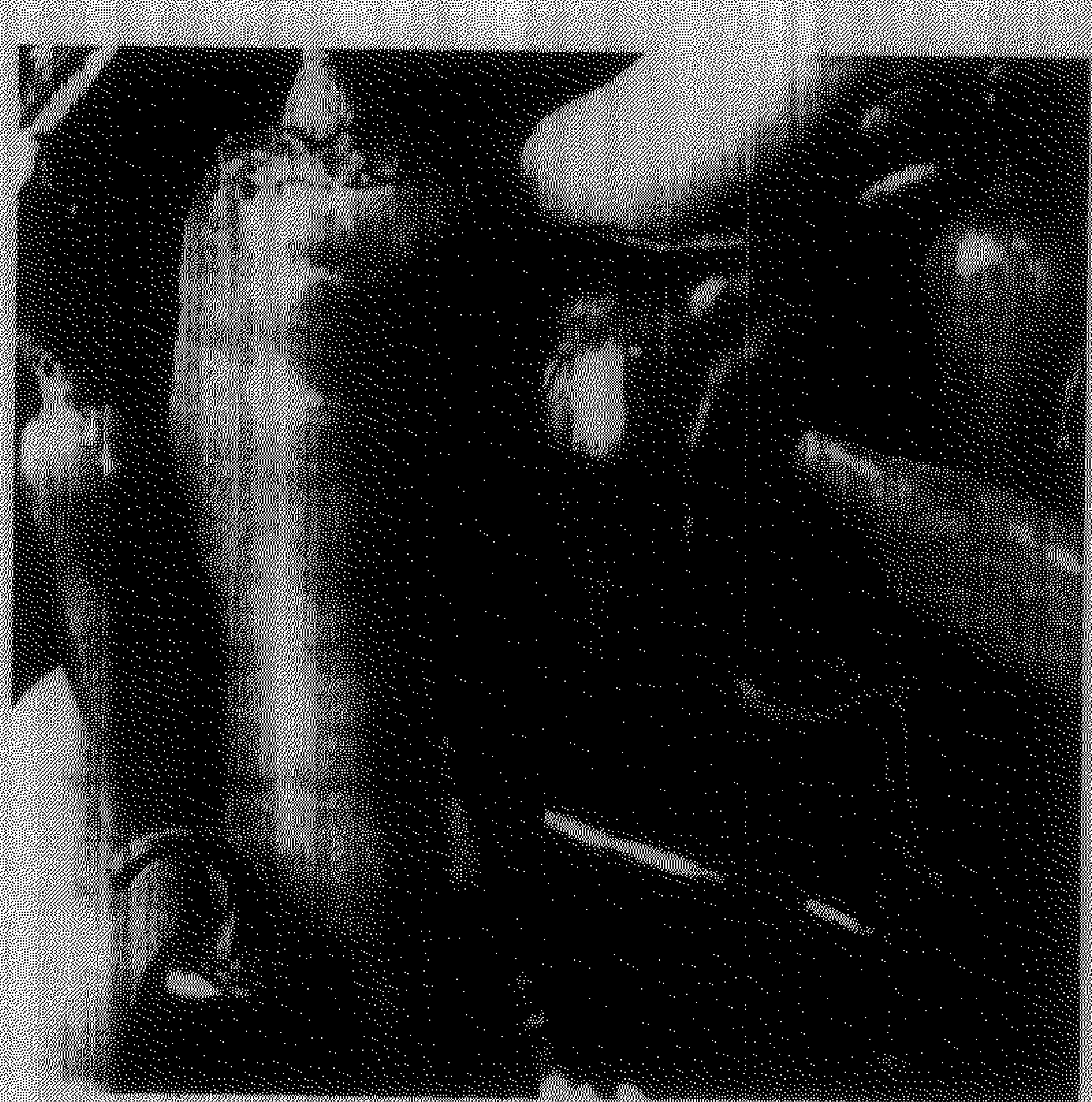
PSNR=38.19 CR=51.50:1



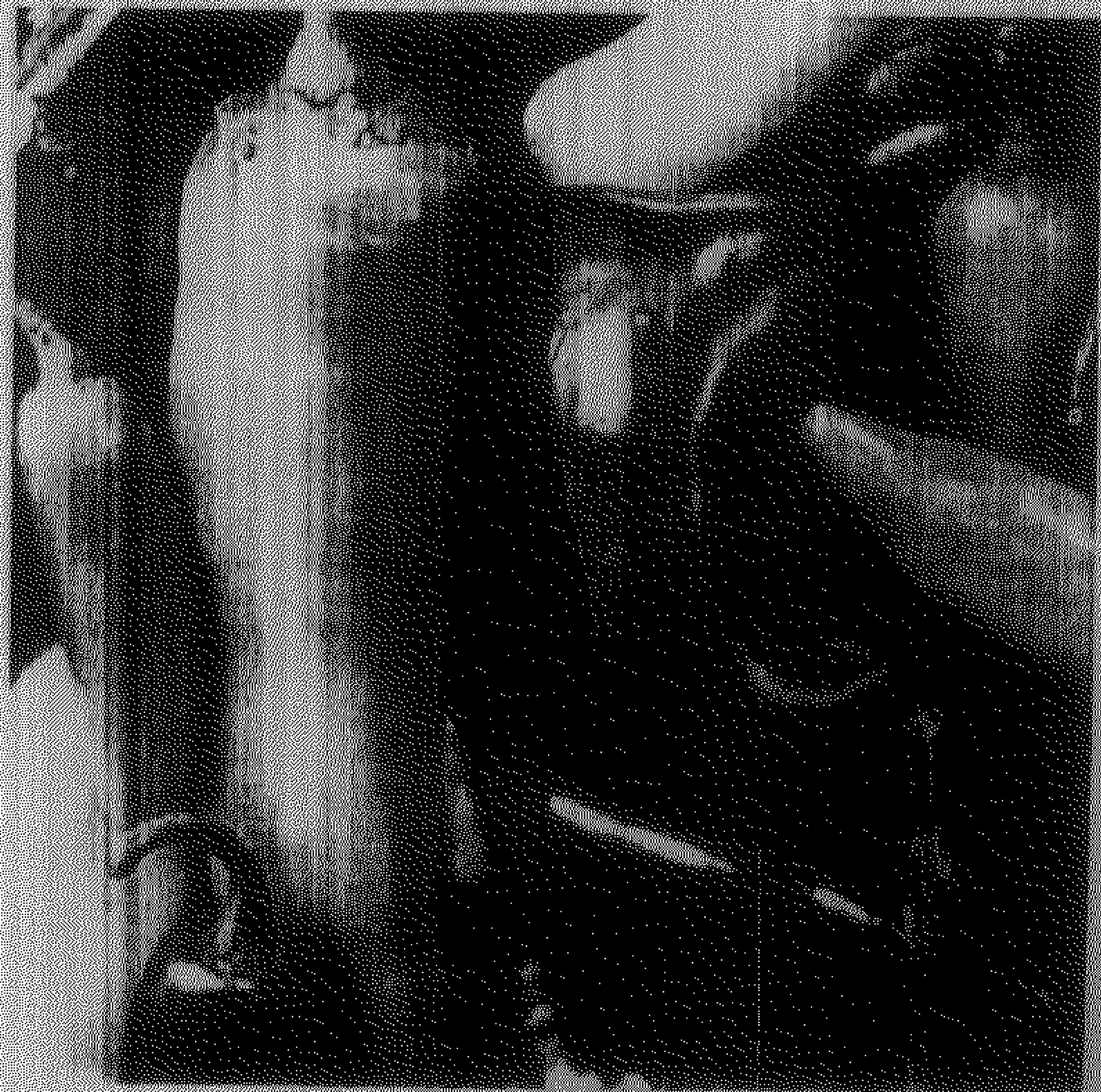
PSNR=37.43 CR=65.18:1



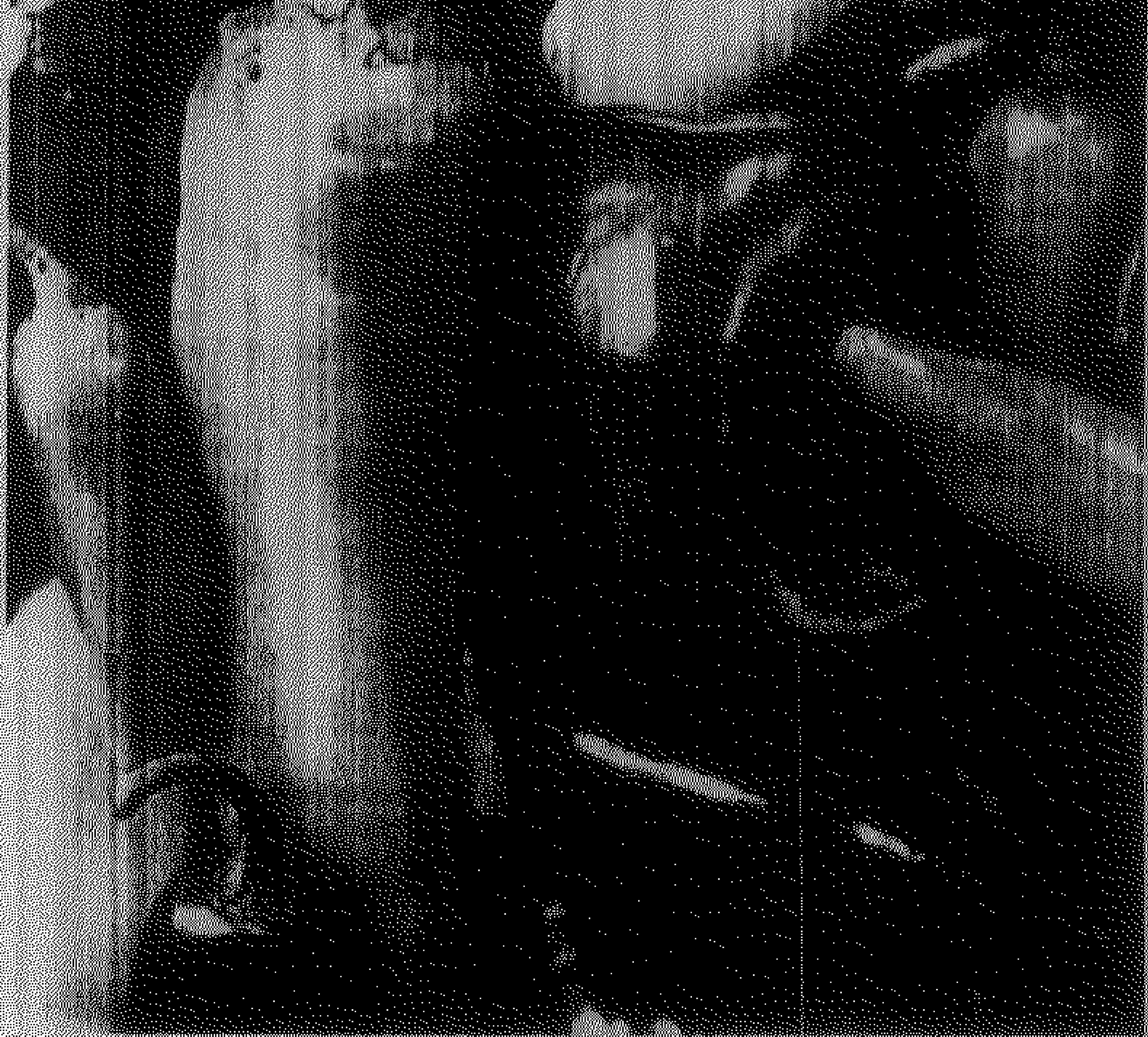
PSNR=36.51 CR=81.27:1



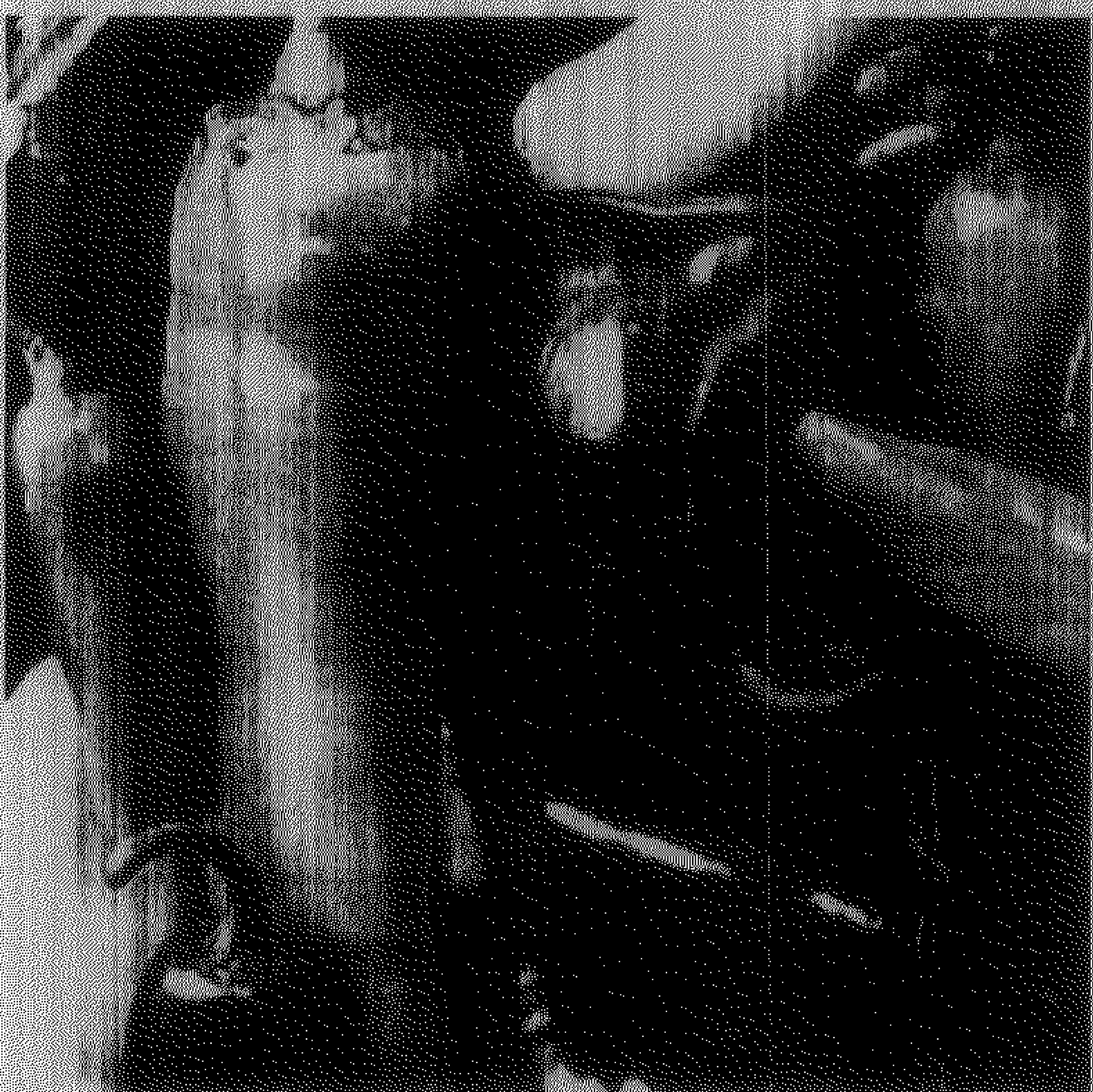
PSNR=35.85 CR=89.94:1



PSNR=35.40 CR=95.10:1



PSNR=34.96 CR=98.21:1



PSNR=34.68 CR=100.43:1

Comparisons:

The results of the proposed algorithm are compared with that of JPEG 2000 and the comparisons are given below. The comparison is done with the JJ2000 [13] software, the java implementation of JPEG 2000.

Comparison for Lena Image (512x512)

Compression Ratio	PSNR Using Proposed Algorithm	PSNR Using JPEG 2000
28.55:1	41.84	42.14
45.59:1	39.96	41.15
58.76:1	38.71	39.48
77.59:1	37.25	37.75
88.27:1	36.39	37.10
95.47:1	35.90	36.28
99.58:1	35.51	35.96
102.10:1	35.20	35.42

Comparison for Fruits Image (512x512)

Compression Ratio	PSNR Using Proposed Algorithm	PSNR Using JPEG 2000
25.00:1	39.57	41.12
41.22:1	38.05	39.49
54.40:1	37.11	37.91
73.26:1	36.08	36.54
85.02:1	35.43	35.97
92.48:1	34.93	35.30
97.06:1	34.58	35.11
99.90:1	34:39	35.05

From the above table we can see that for the same compression ratio the proposed algorithm gives almost similar PSNR values as given by JPEG 2000.

Chapter 6

Conclusions, discussions and scope for further work

The proposed algorithm has been tested for different level decomposition and the best compression ratio has been achieved using $(\log_2 S-3)$ level decompositions, where S is the size of the image. As the value of the quantization constant P increases the compression ratio increases but at the same time image quality decreases. This is because, as the value of P increase the less no of wavelets coefficient are taken into account. If we take more no of wavelet coefficients then we will get a better quality of image with lesser compression ratio. So there is a tradeoff between Image quality and compression ratio.

Though the proposed algorithm is implemented using Haar Wavelets one can develop one's own wavelets and use the best one which is suitable for Color Image to attain more compression.

In the proposed algorithm only the $\mathbf{HH_1}$ portion has been removed, one can remove all the HH portions, since the HH portion of the transformed image contains information about the corner points and the diagonal edges. Majority of the information is present in the other portions. So, generally the loss incurred by removing HH portion is negligible and not discernable to the naked eye. By removing all the HH portion of the decomposed image a better compression ratio can be achieved.

After quatization the most of the values of the block become zero. If the block is scan in zigzags fashion, which produces the string of 64 numbers that starts with some nonzero values and ends with many consecutive zeros. Now if only the nonzero values are output using Modified Huffman coding followed by Modified Run length coding and are followed by a special end of-block (EOB) code, instead of sending the code of the run count of all the trailing zeros. Then more compression can be achieved.

Though the YCrCb Color space has been used in the proposed algorithm one can develop suitable color transformation for which more compression can be achieved in case of color image.

Now a day's computer becomes more interactive. User will be satisfied if he/she can specify the compression ratio or image quality and accordingly gets the output. This can be done using embedded Zero tree wavelets coding [14].

Bibliography

[1] K.N. Plataniotis, A.N. Venetsanopoulos, Color Image Processing and applications, Springer-Verlag Berlin Heidelberg New York, 2000

[2] C.Sidney Burrus, Ramesh A.Gopinath and Haitao Guo, Introduction to Wavelets and Wavelet Transforms, Prentice Hall, New Jersey, 1998.

[3] David Salomon, Data Compression - The Complete Reference, second edition, Springer-Verlag Newyork, Inc., 2000

[4] Daubechies, I., Ten Lectures on Wavelets, SIAM, Philadelphia, 1992.

[5] Yun Q Shi, Huifang Sun, Image and Video Compression for Multimedia Engineering, CRC Press LLC, 2000

[6] M.F. Barnsley, A. Jacquin, F. Malassenet, L. Reuter, and A.D. Sloan, "Harnessing chaos for image synthesis", Computer Graphics, vol 22, no. 4 pp. 131-140, 1988.

[7] W. Philips, ``Weakly separable bases for fast segmented image coding," in *Hybrid Image and Signal Processing IV*, vol. 2238, pp. 153-163, Apr. 1994. SPIE Conference, Orlando

[8] Diego Santa-Cruz and Touradj Ebrahimi, "An analytical study of JPEG 2000 Functionalities", Proceedings of the IEEE International Conference on Image Processing (ICIP), vol. 2, pp. 49-52, September, 2000.

[9] G. Strang, "The Discrete Cosine Transform", Society for Industrial and Applied Mathematics, 1999.

[10] http://home.earthlink.net/~lumenmedia/designNotes/colorSpaces.html

[11] http://www.newi.ac.uk/computing/modules/cim/coursework/mpeg/Mpeg.htm

[12] http://www.jpeg.org/public/jpeglinks.html

[13] http://jpeg2000.epfl.ch/

[14] Shapiro, J. M EMBEDDED *IMAGE CODING USING ZEROTREES OF WAVELET COEFFICIENTS* IEEE Transactions on Signal Processing, Vol. 41, No. 12 (1993), p. 3445-3462.