M.Tech (Computer Science) Dessertation Series

CONNECTIONIST MODEL BASED ON CELLULAR NEURAL NETWORK FOR OBJECT EXTRACTION

a dessertation submitted in partial fulfilment of the requirements for the M.Tech (Computer Science) degree of the Indian Statistical Institute

by

Harish P

under the sunervision of

Prof. Sankar K. Pal
and
Dr. Malay K. Kundu

Machine Intelligence Unit
Indian Statistical Institute
Calcutta 700 035.

Acknowledgements

I wish to express my deepest sense of gratitude to Prof. S. K. Pal for having given me an opportunity to work under him. The present work is a result of his relentless motivation and his valuable guidance. I wish to gratefully acknowledge the efforts of Dr. M. K. Kundu, my co-guide, for his guidance and incessant encouragement. I take this opportunity to thank A. Ghosh, D. P. Mandal, and D. Bhandari for their altruistic attitude at every stage. I consider myself very fortunate for having interacted with such an erudite group.

Calcutta *Harish P.*

Certificate

This is to certify that the work in the dessertation titled *Connectionist model based on cellular neural network for object extraction* has been undertaken by **Harish P.** under our guidance and supervision. The dessertation is found worthy of acceptance for the award of the degree of *Master of Technology in Computer Science.*

19.7.93

Prof. S. K. Pal

Dr. M. K. Kundu

July, 1993.

Machine Intelligence Unit
Indian Statistical Institute
Calcutta - 700 035

ABSTRACT

Cellular neural networks are made of massive aggregate of analog circuit components called cells, and interconnections between them. Any cell in a cellular neural network is connected only to its nearest neighbor cells and interact only with them. Cells not directly connected together affect each other due to propagation effects. Since a cellular neural network is a highly parallel analog circuit it has an ability to process signals in real time. In view of the local interconnections they are more suitable to VLSI implementation than general neural networks. A typical cell of a cellular neural network contains both linear and non-linear circuit elements such as resistors, capacitors, controlled voltage & current sources and independent voltage & current sources. The dynamics of cellular neural networks have both feedback and control operators. The feedback and control to a cell can be set using appropriate cloning templates. The input to a cell is restricted to $[-1, +1]$ and the output of a cell is binary valued (-1 or +1). A properly chosen cloning template can impart a cellular neural network an ability to extract some spatial properties from the input. We have conducted the following two investigations.

(i) The first part demonstrates an application of cellular neural networks for object extraction problem. Object extraction involves classifying all the pixels of image as belonging to object or background depending upon the spatial and grey level properties of the concerned pixel. In order to apply cellular neural network for object extraction a cell is applied to each pixel and the grey value of each pixel(normalized between $[-1, +1]$) is the input to the cell. Global and local information of the image and the cloning template in use determine its dynamics. The output of each cell determines whether the pixel belongs to object or background (-1 or +1).

The study was conducted on synthetic images with different signal to noise ratios and also on real images using different cloning templates. The performance has also been quantitatively determined in terms of percentage pixels correctly classified. The results are compared with those obtained from other neural network (such as Hopfield network and Self-organizing neural network based techniques. It has been found that a 2-connected and 3-connected cloning template performs better in the images with predominantly thin

elongated objects, while 4-connected cloning template performs better for compact objects. Extraction of boundaries of object regions has also been accomplished using cellular neural network as a part of the experiment.

(ii) Cloning template plays an major role in the dynamics of cellular neural network, hence selection of an appropriate cloning template is an important step in object extraction using cellular neural network. An attempt has been made to generate a cloning template automatically by using Genetic Algorithms which is an adaptive, robust and parallel search technique for machine learning. The fitness function used in Genetic Algorithms was the divergence measure between two fuzzy sets. The cloning template obtained by the above technique was found to perform uniformly well under different signal to noise ratios.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

For any image analysis / vision system, one desires to achieve robustness of the system with respect to random noise and failure of components, and to obtain output in real time. Moreover, a system can be made artificially intelligent if it is able to emulate some aspects of human information processing system. Connectionist model or neural network based approaches are attempts to achieve these goals. An artificial neural network can formally be defined as *massively parallel interconnected network of simple (usually adaptive) processing elements which are intended to interact with the objects of the real world in the same way as the biological systems do.* The massive connectivity among the neurons usually make the system fault tolerant (with respect to noise and component failure), while parallel processing capability enables the system to produce output in real time. On the other hand, most of the image processing operations use neighborhood information to reduce the local ambiguity and to attain a global consistency, i.e., the operations are distributed and parallel in nature. The parallel and distributed processing characteristics of neural networks suggest that many image processing applications are candidates for neural network implementation. There are many practical applications, for example, handwritten character recognition, spoken language translation, speech and image recognition etc., at which biological systems perform faster and better than any currently available information processing systems. Connectionist models are intended to emulate electronically the performance of the biological systems in such tasks.

Connectionist models or artificial neural networks are a fundamentally new and different information processing paradigm. Connectionist models emulate the organization and the massively parallel computing abilities of a biological nervous system. It may be noted that these models are extreme simplifications of the actual biological nervous system. The computational elements (called neurons / nodes / processors) in connectionist models are analogous to the fundamental constituents of biological nervous system. A biological nervous system has a large number of neurons (roughly 10^{11}) and high connectivity between them, whereas in the artificial neural network models the connectivity and the number of neurons is very low.

Like biological systems artificial neural networks exhibit the following similarities

- *LEARNING :* Artificial neural networks can modify their behaviour in response to their environment.
- *ABSTRACTION :* Some neural networks are capable of abstracting the fundamental features of the inputs. For example, if a network is trained on a sequence of distorted versions of the alphabet 'A' adequately, the output produced by such a distorted version will retain the main characteristics of the pattern.
- *GENERALIZATION :* Once an artificial neural network is trained, its response can be to a large extent insensitive to minor variations in the input. The ability to see through noisy and disturbed patterns is vital to real-world pattern recognition.

A variety of neural network models have been proposed [1, 2, 3] in the last decade. Although they differ in structure, functioning and behaviour, they exhibit some common features. The common features are

- Each node/neuron/processor is usually described by a single real variable called its state, which represents its output.
- The nodes are densely connected so that the state of one node affects the potential of many other nodes according to the strength of the connections.

- The new state of the neurons is a non-linear function of the potential created by the firing activity of other neurons.
- Input to the network is given by setting the states of a subset of the nodes to a specific values initially.
- The processing takes place through the evolution of states of all nodes according to its dynamics and continues until it stabilizes.
- The training (learning) of the network is a process whereby the values of the connection strengths are modified in order to achieve the desired output.

Neural networks are designated by the network topology, connection strengths between pairs of neurons, node characteristics and the status updating rules. Generally, an objective function is defined which represents the complete status of the network and a set of minima of the objective function gives the stable states of the neural network. The neurons operate asynchronously or synchronously, thereby providing the output in real time. Because of massive parallelism and high connectivity connectionist models are reputed to enjoy the following major advantages.

- *ADAPTIVITY* : Adjusting the strengths to new data / information.
- *SPEED* : Due to massively parallel architecture and analog circuit components.
- *ROBUSTNESS* : To missing, confusing and ill-defined / noisy data. Neural networks are also robust to component failure due to the redundancy of links and nodes.
- *OPTIMALITY* : With regards to the error rates in classification.

One of the interesting characteristics of a neural network is its ability to learn. During learning the network weights gradually converge to values such that each input vector produces the desired output vector. Learning algorithms are categorized as *supervised* and *unsupervised*. Supervised learning requires the pairing of input vector and target vector representing the desired output. Usually a network is trained over a number of such training pairs, till the error is negligible. Despite many application successes, supervised learning has been criticized as biologically implausible. Unsupervised learning is far more plausible model of learning in the biological systems. In unsupervised learning the training set consists solely of input vectors.

Several neural network models have been proposed in the last few years. They are

1. *Hopfield's network :* Associative memory / content addressable memory.
2. *Kohonen's model of self-organizing neural network :* Regularity detector / unsupervised classifier.
3. *Multi-layer perceptron :* supervised classifier.
4. *Cellular neural network :* unsupervised classifier.

Artificial neural network models have been shown successful in different pattern recognition applications areas, especially in

- *Document Processing :* For recognition of printed and handwritten characters.
- *Biology and Medicine :* Application in radiology, EEG / ECG analysis.
- *Forensic science :* For finger print analysis, face recognition.
- *Defense applications :* Target detection, missile guidance.
- *Industrial automation :* Robotics, non-destructive testing.
- *Remote sensing :*

The present work demonstrates some applications of connectionist models for pattern recognition problems. The problem is to use *cellular neural network for object extraction and use of genetic algorithms to select optimum parameters.* The study was conducted with both noisy synthetic and real images using different cloning templates. The performance has been quantitatively evaluated. The results are compared with those obtained from existing other neural network based methods.

The rest of the report is organized as follows. Chapter 2 contains investigations on object extraction problem using a cellular neural network, and use of genetic algorithms for choosing the network parameters automatically. Concluding remarks along with discussions are given in Chapter 3.

Chapter 2

Object Extraction Using Cellular Neural Networks

The present chapter describes the utility of cellular neural network for object extraction. Genetic Algorithms are also incorporated here to select some of the network parameters automatically. Before giving the details of how a cellular neural network can be used for object extraction, let us desribe the architecture and dynamics of cellular neural network briefly.

2.1 Description of Cellular Neural Network

This is an unsupervised neural network proposed by Chua and Yang in [4, 5]. Cellular neural networks are made of massive aggregate of analog circuit components called cells. Any cell in a cellular neural network is connected only to its neighboring cells and interact only with them. Key features of this network are asynchronous processing, continuous time dynamics and local interaction between the cells (processing elements). Cells not directly connected together affect each other due to propagation effects. The continuous time feature of cellular neural network imparts it real-time signal processing capability, while the local inter-connections make it amenable to VLSI implementation. A two dimensional cellular neural network of size 4×4 is shown in Figure 1. The square units are called cells. The links

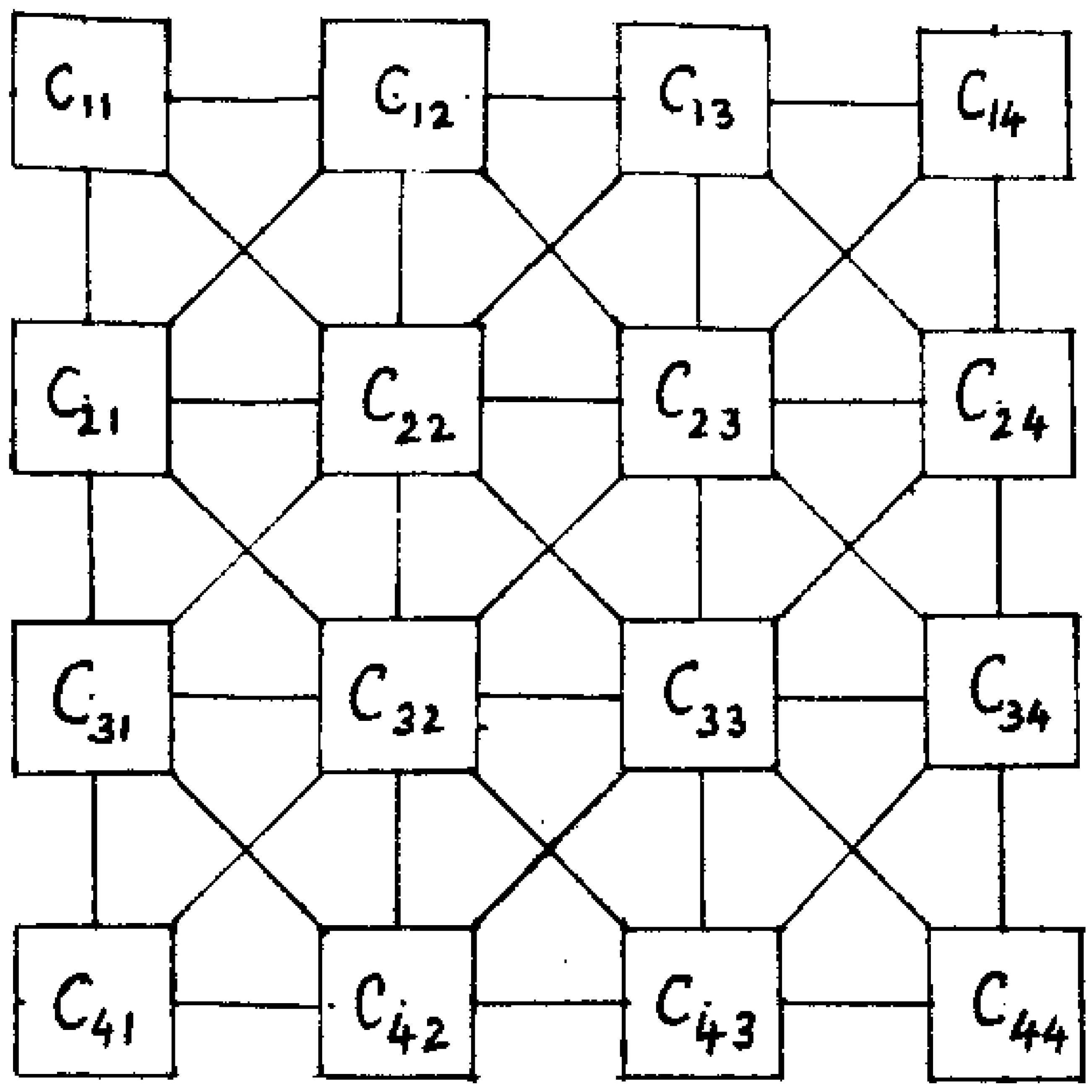

Figure 1. A two-dimensional cellular neural network.

between cells indicate the interconnections between them.

2.1.1 Architecture

Let us consider a $M \times N$ cellular neural network $\{ C_{ij} \}$ with MN cells. where C_{ij} represents the cell of i^{th} row and j^{th} column. Before describing the structure of the cell let us introduce a few definitions

1. *r neighborhood* : The r neighborhood of C_{ij} in a cellular neural network is defined by

   $N^r_{ij} = \{C_{kl} \mid max(|k-i|,|l-j|) \leq r, 1 \leq k \leq M; 1 \leq l \leq N\}$. •

2. *Inner cells* : Cells which have exactly $(2r+1)^2$ neighboring cells are called *inner cells*. All other cells are called *boundary cells* . •

3. *Symmetric property* : If $C_{ij} \in N^r_{kl}$ *then* $C_{kl} \in N^r_{ij}$. $\forall$ C_{ij}, C_{kl}. •

Structure of a *cell*

A typical circuit diagram of a cell is shown in Figure 2. Voltage v_{xij} is called the state of the cell C_{ij}, voltage v_{uij} is the input to the cell C_{ij} and voltage v_{yij} is the

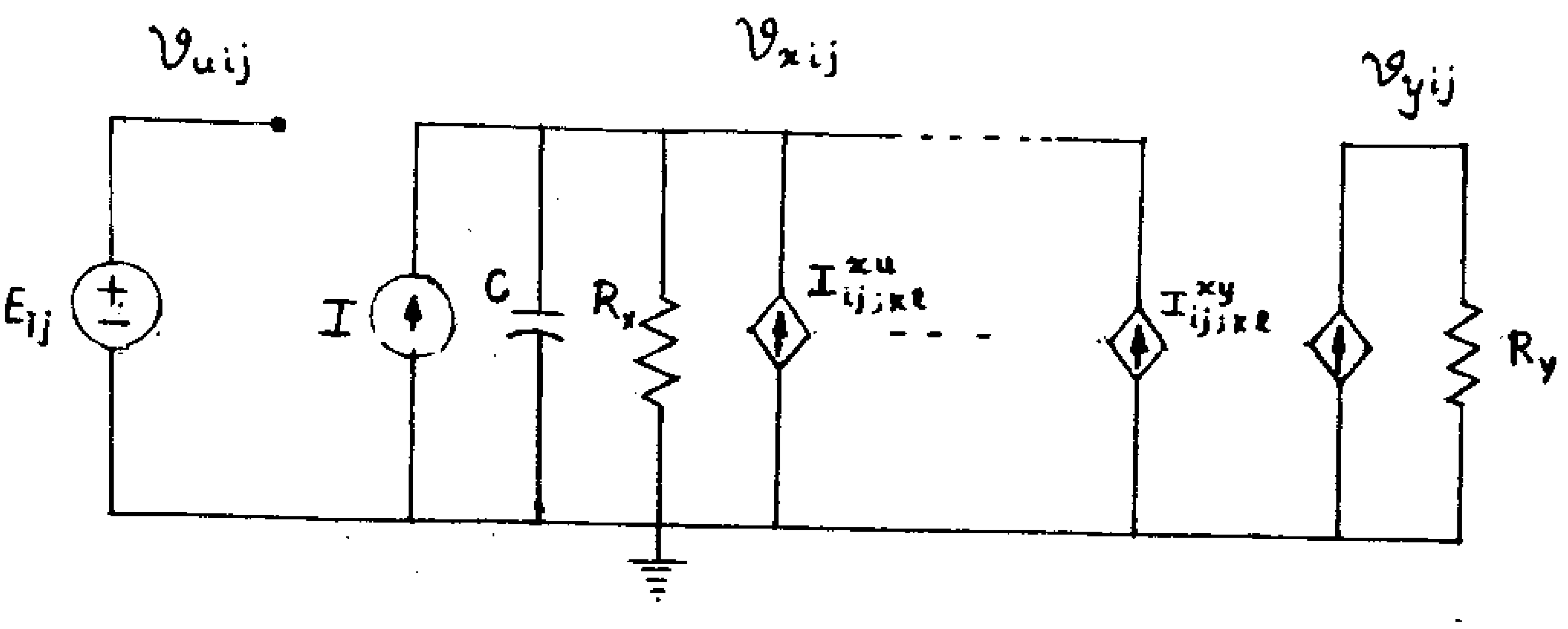

Figure 2. Circuit diagram of a *cell*

output of the cell C_{ij}. Initially input and state voltages of cell C_{ij} are assumed to lie between -1 and +1.

Every cell C_{ij} contains one independent voltage source E_{ij}, one independent current source I, two resistors R_x and R_y,one capacitor C, and atmost 2m voltage controlled current sources (where m is the number of neighbor cells) which are coupled to its neighbor cells via the controlling input voltage v_{ukl}, and the feedback from the output voltage v_{ykl} of the neighbor cell C_{kl}. $I^{xy}_{ij\,;\,kl}$ and $I^{xu}_{ij\,;\,kl}$ are voltage controlled current sources with the properties : $I^{xy}_{ij\,;\,kl} = A_{ij\,;\,kl}\, v_{ykl}$ and $I^{xu}_{ij\,;\,kl} = B_{ij\,;\,kl}\, v_{ukl} \quad \forall \ C_{kl} \in N^r_{ij}$. The output feedback effect depends on $A_{ij\,;\,kl}$ and the input control effect depends on $B_{ij\,;\,kl}$. So $A_{ij\,;\,kl}$ and $B_{ij\,;\,kl}$ are known as feedback operator and control operator respectively. In addition a piecewise-linear voltage controlled current source $I_{yij} = (1/R_y)f(v_{xij})$ is also present in a cell. The graphical representation of f is shown in Figure 3. ♣

2.1.2 Dynamics

By applying Kirchoff's current law on the circuit given in Figure 2. the following equations are easily obtained.

State equation :

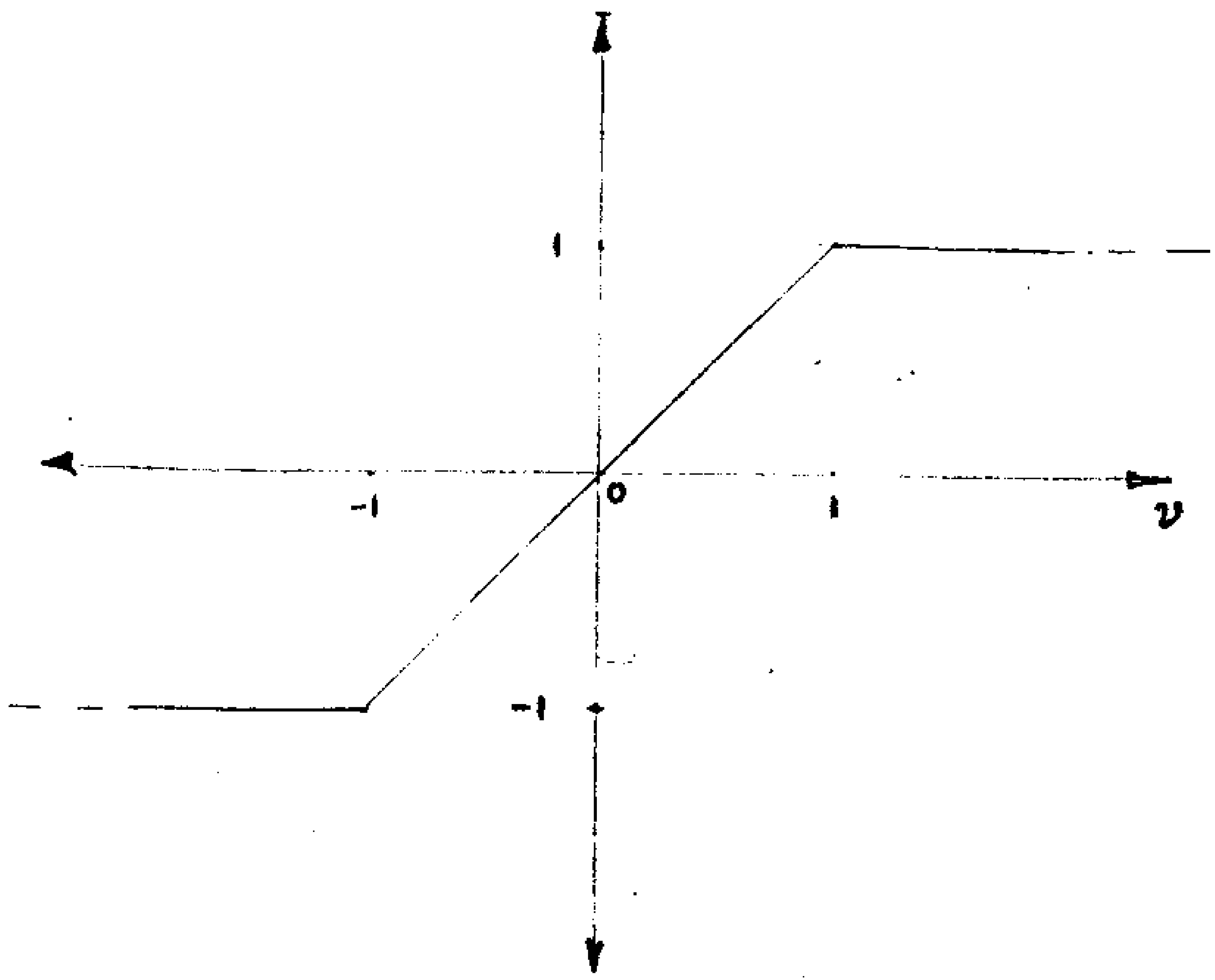

Figure 3. Input-Output characteristics of the piecewise-linear function f

$$C\frac{dv_{xij}(t)}{dt} = \frac{-1}{R_x}v_{xij}(t) + \sum_{C_{kl}\in N^r_{ij}} A_{ij\,;\,kl}\; v_{ykl} + \sum_{C_{kl}\in N^r_{ij}} B_{ij\,;\,kl}\; v_{ukl} + I, \quad 1 \le i \le M; 1 \le i \le N. \tag{2.1}$$

Output equation : The nonlinear function shown in Figure 3 can be written as

$$v_{yij}(t) = 0.5(|v_{xij}(t) + 1| - |v_{xij}(t) - 1|), \qquad 1 \le i \le M; 1 \le i \le N. \tag{2.2}$$

Input equation :

$$v_{uij} = E_{ij}, \qquad 1 \le i \le M; 1 \le i \le N. \tag{2.3}$$

Constraints :

$$|v_{xij}(0)| \le 1, \qquad 1 \le i \le M; 1 \le i \le N. \tag{2.4}$$

and

$$|v_{uij}| \le 1, \qquad 1 \le i \le M; 1 \le i \le N. \tag{2.5}$$

Assumptions :

Since the connection strengths are assumed to be symmetric i.e.,

$$A_{ij\ ;\ kl} = A_{kl\ ;\ ij}, \quad 1 \le i, k \le M; 1 \le j, l \le N. \tag{2.6}$$

and

$$C > 0 \ and \ R_x > 0. \tag{2.7}$$

A cellular neural network has both output feedback and input control mechanisms. CR_x is the time constant of the circuit. R_x and R_y determine the power dissipation in the circuit.

Regarding the bounds of the parameters and the stability of the system the following comments can be made

- The state of each cell in a cellular neural network is bounded. The upper bound v_{max} is

  $$v_{max} = 1 + R_x|I| + R_x \max_{1\le i\le M; 1\le j\le N} \left[\sum_{C_{kl}\ \in\ N^r_{ij}} (|A_{ij\ ;\ kl}| + |B_{ij\ ;\ kl}|) \right].$$

  and the lower bound is

  $v_{min} = -v_{max}$.

- After the transients settle down, a cellular neural network approaches one of its stable states, i.e.,

  $$\lim_{t\longrightarrow\infty} v_{xij}(t) = constant, \quad 1 \le i \le M; 1 \le j \le N.$$

  or

  $$\lim_{t\longrightarrow\infty} \frac{d\ v_{xij}(t)}{dt} = 0, \quad 1 \le i \le M; 1 \le j \le N.$$

- If the circuit parameters satisfy

  $$A_{ij\ ;\ ij} > \frac{1}{R_x}$$

then

$$\lim_{t \longrightarrow \infty} |v_{xij}(t)| \geq 1, \qquad 1 \leq i \leq M; 1 \leq j \leq N;$$

or equivalently

$$\lim_{t \longrightarrow \infty} v_{yij}(t) = \pm 1, \qquad 1 \leq i \leq M; 1 \leq j \leq N.$$

- At equilibrium the state v^*_{xij} of a cell C_{ij} takes any value of the state variable v_{xij} which satisfies

  1. $[\frac{d\ v_{xij}(t)}{dt}]_{v_{xij}=v^*_{xij}} = 0$;
  2. $v_{ykl} = \pm 1$

  $\forall C_{kl} \ \in \ N^r_{ij}$.

- A cell equilibrium state v^*_{xij} of a cell C_{ij} is said to be stable iff $|v^*_{xij}| > 1$

- A stable equilibrium point of a cellular neural network with $A_{ij\ ;\ kl} > \frac{1}{R_x}$ is any state vector $\mathbf{V}_x$ whose state variable components v_{xij} consists of stable cell equilibrium states.

- The transient response of the cellular neural network is the trajectory from initial state to the final state of the system. Any stable system equilibrium point of a cellular neural network is the limit point of the differential equation (2.1). ♠

2.2 Cellular Neural Network for Object Extraction

Image processing [6, 7] refers to operations that transform an input image into other image forms which are in some sense improved versions of the input image. It is concerned with the manipulation and analysis of images by a computer. Its major sub-areas include

1. *Digitization and Compression :* Converting images into discrete (digital) form; efficient coding or approximation of image so as to save storage space.

2. *Enhancement, Restoration and Reconstruction :* Improving the quality of the degraded (low-contrast, blurred, noisy) images; Reconstructing images from a set of projections.

3. *Matching, Description and Recognition :* Segmenting images into parts, measuring properties of and relationships among the parts, describing them and comparing the resulting descriptions to models that define classes of images.

2.2.1 Object Extraction

The process of partitioning a scene into connected/non-overlapping meaningful regions (representing the naturally occurring components in an image) is known as segmentation. This is one of the most important tasks for image analysis / computer vision system. Various approaches like histogram thresholding, clustering, edge detection and iterative pixel classification have been suggested for this purpose. Once the meaningful regions have been extracted their properties (features) can be computed. For recognition tasks we need to distinguish between object regions and background region.

Object extraction involves segmenting the whole image into two region types namely, object region and background region. Two adjacent pixels of an image belong to the same region if they have similar gray level properties. So, both gray level and positional properties play an important role in making a decision whether a pixel belongs to object or background. ♣

2.2.2 Cellular neural network and object extraction

In order to demonstrate how a cellular neural network can be used for object extraction let us first approximate equations (2.1) and (2.2) by a difference equation.

$$
\begin{aligned}
\tfrac{C}{h}[v_{xij}((n+1)h) - v_{xij}(nh)] = \tfrac{-1}{R_x} v_{xij}(nh) \quad &+ \sum_{C_{kl} \in N^r_{ij}} A_{ij\ ;\ kl}\ v_{ykl} \\
&+ \sum_{C_{kl} \in N^r_{ij}} B_{ij\ ;\ kl}\ v_{ukl} + I, \qquad (2.8) \\
&\quad 1 \le i \le M; 1 \le i \le N,
\end{aligned}
$$

where $t = nh$, where h is constant time step.

and

$$
\begin{aligned}
v_{yij}(nh) = \;& 0.5(|v_{xij}(nh) + 1| - |v_{xij}(nh) - 1|), \\
= \;& f(v_{xij}(nh)), \qquad (2.9) \\
& 1 \le i \le M; 1 \le i \le N.
\end{aligned}
$$

Now let,

$$
\begin{aligned}
I_{ij} = & \sum_{C_{kl} \in N^r_{ij}} B_{ij\ ;\ kl}\ v_{ukl} + I, \qquad (2.10) \\
& 1 \le i \le M; 1 \le i \le N.
\end{aligned}
$$

Then from equations (2.8),(2.9) and (2.10) we can write

$$
\begin{aligned}
v_{xij}(n+1) = v_{xij}(n) \quad + \quad & \tfrac{h}{C}\left[\tfrac{-1}{R_x} v_{xij}(n) + \sum_{C_{kl} \in N^r_{ij}} A_{ij\ ;\ kl}\ f(v_{xkl}) + I_{ij}\right] \qquad (2.11) \\
& 1 \le i \le M; 1 \le i \le N.
\end{aligned}
$$

For simplicity the time step has been suppressed from h to nh i.e., $v_{xij}(n) \equiv v_{xij}(nh)$

Equation (2.11) can be interpreted as a two-dimensional filter transforming the state vector $\mathbf{V}_x(n)$ to another one, represented by $\mathbf{V}_x(n+1)$. In the one-step filter in equation (2.11), any component of the state vector $v_{xij}(n+1)$ of the cell C_{ij} is determined directly from state vector components of the previous time instant,$v_{xkl}(n)$, where $C_{kl} \in N^r_{ij}$. A one-step filter can only make use of properties of its nearest neighbors. Local properties are propagated further with time. So we can use a cellular neural network to obtain a dynamic transform of an initial state vector $\mathbf{V}_x(0)$. When $t \longrightarrow \infty$, the state vector $\mathbf{V}_x$ tends to a constant and output vector $\mathbf{V}_y$ tends to either +1 or -1. In other words, a cellular neural network processes signals by mapping from one signal space to another one. If the initial state space is $[-1.0, 1.0]^{M \times N}$ and the output space as $\{-1, 1\}^{M \times N}$, then the dynamic map F, can be defined as

$$F : [-1.0, 1.0]^{M \times N} \longrightarrow \{-1, 1\}^{M \times N}. \tag{2.12}$$

This mapping can be used to partition a continuous signal space into a discrete one. This property of the cellular neural network can be used for object extraction in image processing.

Object extraction in an image of size $M \times N$ can be considered as a mapping

$$E : [a, b]^{M \times N} \longrightarrow \{A, B\}^{M \times N} \tag{2.13}$$

where the range of gray levels of the images is $[a, b]$, and A,B are gray levels of object and background respectively or vice versa. Equations (2.12) and (2.13) are analogous. Thus if we can transform the gray level of the input image into [-1.0 , 1.0] we can achieve a transformation of the image into {-1,1} by using a cellular neural network.

In order to design a cellular neural network for object extraction we must assign appropriate values to the circuit parameters. Circuit parameters include values of the capacitor C, resistors R_x and R_y, the independent current source (cell bias). We should also decide upon the neighborhood for each cell and the values of the feedback $A_{ij\ ;\ kl}$ and control $B_{ij\ ;\ kl}$ operators. For all the image processing problems we assume the neighborhood to be constant and the control and feedback operators to be position invariant. Generally, for object extraction problem the control operator $B_{ij\ ;\ kl}$ and the cell bias are set to 0. C, R_x and R_y are selected after choosing the time constant and power dissipation acceptable for the circuit; a typical choice [5] may be

$$
\begin{aligned}
A_{ij\ ;\ i-1,j-1} &= 0, \\
A_{ij\ ;\ i-1,j} &= 10^{-3}\Omega^{-1}, \\
A_{ij\ ;\ i-1,j+1} &= 0, \\
A_{ij\ ;\ i,j-1} &= 10^{-3}\Omega^{-1}, \\
A_{ij\ ;\ i,j} &= 2\times 10^{-3}\Omega^{-1}, \\
A_{ij\ ;\ i,j+1} &= 10^{-3}\Omega^{-1}, \\
A_{ij\ ;\ i+1,j-1} &= 0, \\
A_{ij\ ;\ i+1,j} &= 10^{-3}\Omega^{-1}, \\
A_{ij\ ;\ i+1,j+1} &= 0,
\end{aligned}
$$

$$C = 10^{-9}\ \mathrm{F},\ R_x = 10^3\Omega,\ \mathrm{I} = 0\ \mathrm{A},$$

$$and\ \ B_{ij\ ;\ kl} = 0\ for\ C_{kl}\ \in\ N^r_{ij}.$$

We can often specify the feedback operator coefficients in the form of a square matrix, which is called *cloning template* . *Cloning template* specifies the dynamic transformation of the cellular neural network. The equations which govern the dynamics corresponding to the above parameters are

$$\frac{dv_{xij}(t)}{dt} = 10^6[-v_{xij}(t)+2v_{yij}(t)+v_{yij-1}(t)+v_{yi-1j}(t)+v_{yi+1j}(t)+v_{yij+1}(t)] \quad (2.14)$$

and

$$v_{yij}(t) = 0.5(|v_{xij}(t)+1| - |v_{xij}(t)-1|) \qquad 1 \le i \le M; 1 \le i \le N.$$

It is convenient to recast the right-hand side of equation (2.14) into a symbolic form

$$\frac{dv_{xij}(t)}{dt} = -10^6 v_{xij}(t) + 10^6 \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 2.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \star v_{yij}(t) \qquad (2.15)$$

where $\star$ is a convolution operator. For any cloning template T, the convolution operator $\star$ is defined as

$$T \star v_{ij} = \sum_{C_{kl} \in N_{ij}^r} T_{k-i,l-j} v_{kl} \tag{2.16}$$

where $T_{m,n}$ denotes the entry in the m^{th} row and n^{th} column of the cloning template, and $-r \leq m, n \leq r$ (r stands for neighborhood). In the above definition $A_{ij\,;\,kl}$ is assumed to be position invariant.

In order to apply cellular neural network for object extraction the range of gray levels of the input image is scaled to [-1,1]. The transformed gray value of each pixel is then given as input to a cell. After the network reaches a stable equilibrium state the outputs of the cell will be either -1 or +1, representing whether it belongs to object and background region or vice versa. ♠

2.3 Optimum Network Parameter Selection using Genetic Algorithms

There is no standard method for selecting circuit parameters for a cellular neural network based system [8]. Some heuristics have been suggested [4, 5] for this task. In this section we attempt to generate optimum parameters of a cellular neural network used for object extraction by using a powerful parallel search technique called genetic algorithms.

2.3.1 Optimum selection of parameters

We can rewrite the equation (2.1) in the form

$$\begin{aligned} \frac{dv_{xij}(t)}{dt} = \; & \frac{1}{C}\left[\frac{-1}{R_x} v_{xij}(n) + \sum_{C_{kl} \in N_{ij}^r} A_{ij\,;\,kl}\, f(v_{xkl}) + I_{ij}\right], \\ & 1 \leq i \leq M; 1 \leq i \leq N. \end{aligned} \tag{2.17}$$

We can choose R_x and C based on the time constant and the power dissipation acceptable for the circuit. If we assume time constant to be 1 micro second, we can arbitarily fix up the values of the resistances R_x and R_y. For a time constant of 1 micro second, $R_x = 10^3\Omega$ and $I_{ij} = 0$. Then the equation (2.17) takes the form

$$\frac{dv_{xij}(t)}{dt} = -10^6 v_{xij}(t) + 10^6 \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \star v_{yij}(t) \tag{2.18}$$

where $x_1, x_2 \ldots, x_9$ are expressed in $10^{-3}\Omega^{-1}$.

The problem of choosing correct parameters of a cellular neural network for object extraction is equivalent to searching an appropriate set of parameters $x_1, x_2 \ldots, x_9$ (subject to constraints) which gives the best object background classification. We adopt a powerful search technique called *genetic algorithms* for deciding on the parameters for cloning template. For convenience, let us describe the genetic algorithms briefly.

Genetic Algorithms : Basic Principles and Features

Genetic algorithms [9, 10] are highly parallel and adaptive search and machine learning processes based on the mechanics of natural selection in natural genetic systems. Genetic algorithms are capable of solving wide range of complex optimization problems using genetic operators (reproduction/selection, crossover and mutation) on coded solutions (strings/chromosomes) in an iterative fashion. They efficiently exploit historical information to speculate on new search points with expected improved performance. Genetic algorithms deal simultaneously with multiple points (called, population), not a single point, which helps to find the global optimal solution without getting stuck at local optima. Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces, even if the search (e.g., optimization) function spaces are not smooth or continuous, which are very difficult (sometimes impossible) to handle using calculus based methods. Genetic algorithms are blind, that is, they use only the payoff or penalty (i.e., objective) function and do not need any other auxiliary information.

To solve an optimization problem, genetic algorithms start with the chromosomal (structural) representation of a parameter set. The parameter set is to be coded as a finite length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's. For example, let $\{a_1, a_2, ..., a_p\}$ be a realization of the parameter set and the binary representation of a_1, a_2, ..., a_p be 10110, 00100, ..., 11001 respectively. Then the string 10110 00100 .... 11001 is a chromosomal representation of the parameter set $\{a_1, a_2, ..., a_p\}$. It is evident that the number of chromosomes (strings) is 2^l where, l is the string length.

Genetic algorithms find the global near optimal solution employing three basic operations over a limited number of strings. The operators are

- Reproduction/Selection.
- Crossover.
- Mutation.

Reproduction is a process in which individual strings are copied according to their objective function values, f, called the fitness function. This operator is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures. More highly fitted strings have a higher number of offsprings in the succeeding generation. These strings are then entered into a mating pool, a tentative new population, for further genetic operator action.

Unlike biological in systems [1], the crossover generates offsprings for the new generation using the highly fitted strings (parents) selected randomly from the mating pool created by the reproduction operation. The crossover may proceed in two steps. First, members of the reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows : an integer position k is selected uniformly at random between 1 and $l-1$, where l is the string length greater than 1. Two new strings are created by swapping all characters from position $k+1$ to l. Let:

$$a = 11000\ 10101\ 01000\ .....\ 01111\ 10001$$
$$b = 10001\ 01110\ 11101\ .....\ 00110\ 10100$$

[1] In biological systems, crossover occurs on two heterogamous chromosomes.

be two strings (parents) selected for the crossing over operation and the generated random number be eleven. Then the newly produced offsprings (swapping all characters after the position 11) will be

$$a' = 11000\ 10101\ 01101\ .....\ 00110\ 10100$$
$$b' = 10001\ 01110\ 11000\ .....\ 01111\ 10001.$$

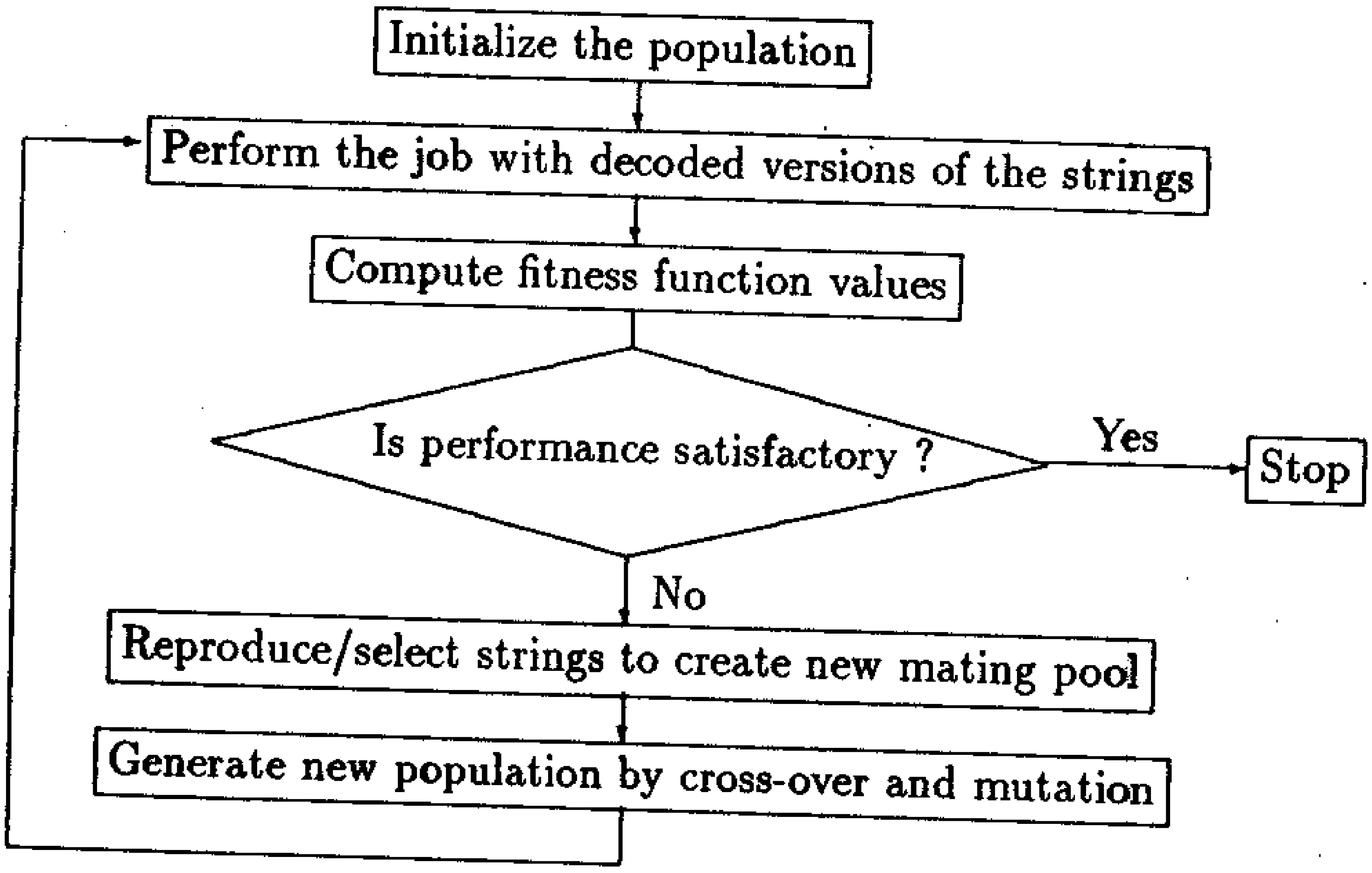

Figure 4. Basic steps of genetic algorithm

In the simple genetic algorithms, mutation is the occasional (with small probability) random alteration of the value of a string position. The mutation operator plays a secondary role in the simple genetic algorithms. Note that the frequency of mutation to obtain good results in the empirical genetic algorithm studies is of the order of one per thousand bit (position). It helps to prevent the irrecoverable loss of potentially important genetic material. A random bit position of a random string is selected and is replaced by an another character from the alphabet. For example, let the third bit of string a, given above, be selected for mutation. Then the transformed string after mutation will be

$$a = 11100\ 10101\ 01000\ .....\ 01111\ 10001.$$

A schematic diagram of the basic structures of a genetic algorithm is shown in Figure 4.

Genetic algorithms are different from the traditional search techniques used in the calculus based approach, dynamic programming and simulated annealing. They are different from most of the normal optimization and search procedures in four ways :

- Genetic algorithms work with the coding of the parameter set, not with the parameters themselves.
- Genetic algorithms search from a population, not from a single point.
- Genetic algorithms search via sampling, a blind search.
- Genetic algorithms search using stochastic operators, not deterministic rules.

Coding the parameter set enhances the search space extensively. Strings of length l explore a^l search points, where a is the size of the set of alphabet. In particular, when the strings are represented in binary coded form then for $l = 3$ GAs explore $2^3 = 8$ different points (e.g. 000, 001, 010, 011, 100, 101, 110, 111) from three strings 100, 010 and 001 using three basic operators.

Genetic Algorithms for Selection of Cloning Template

Fitness function plays an important role in directing the search in genetic algorithms. A proper selection of such a function is essential in obtaining the optimum parameters for the cloning template. Two fitness functions are used for the present work. They are

1. *Number of pixels wrongly classified*: In this measure we count the number of pixels that have been misclassified and use this to direct the search.

2. *Divergence measure between two fuzzy sets* [11, 12]: We try to minimize the divergence between the obtained and desired images to direct the search. Brief description and properties of divergence measure is given below.

   A crisp subset A of the universal set U is a collection of objects from U, which are members of A. An equivalent way of defining A is to specify a characteristic function of A, $\chi_A : U \longrightarrow \{0,1\} \forall x \in U$, such that

$$\begin{aligned} \chi_A = & \ 1, \ x \in A, \\ = & \ 0, \ x \notin A. \end{aligned} \tag{2.19}$$

   Generalizing the characteristic function from {0,1} to [0,1] one can obtain the fuzzy sets. More specifically, the above concept of characteristic function generalizes to a membership function $\mu : U \longrightarrow [0,1]$. In general a fuzzy set A in the universe of discourse is defined as

   $A = \{\mu_A(x_i)|x_i, i = 1, 2, \ldots, n\}$,

   where $\mu_A(x_i)$ is the membership value for x_i indicating the degree of belongingness to A.

   Let S be the set of supports x_i, $i = 1, 2, \ldots, n$, and $A = \{\mu_1(x_i)|x_i\}$ and $B = \{\mu_2(x_i)|x_i\}$ be two fuzzy sets defined on S. Divergence $D(A, B)$ between A and B is defined as

$$D(A,B) = \frac{1}{n}\sum_{i=1}^{n}[D_i(A,B) + D_i(B,A)] \tag{2.20}$$

   where

$$\begin{aligned} D_i(A,B) = & \ \mu_1(x_i) log\frac{\mu_1(x_i)}{\mu_2(x_i)} + [1 - \mu_1(x_i)] log\frac{1-\mu_1(x_i)}{1-\mu_2(x_i)} \\ & \ 0 < \mu_1(x_i), \mu_2(x_i) < 1 \end{aligned} \tag{2.21}$$

   and

$$\begin{aligned} D_i(B,A) = & \ \mu_2(x_i) log\frac{\mu_2(x_i)}{\mu_1(x_i)} + [1 - \mu_2(x_i)] log\frac{1-\mu_2(x_i)}{1-\mu_1(x_i)} \\ & \ 0 < \mu_1(x_i), \mu_2(x_i) < 1. \end{aligned} \tag{2.22}$$

   Divergence quantifies the descrepancy between two fuzzy sets. Divergence between two fuzzy sets A and B, $D(A, B)$ is maximum if B is the farthest non fuzzy set of A. Divergence between two fuzzy sets is minimum if the two fuzzy sets are similar, i.e., $B = A$.

For an r neighborhood cellular neural network the total number of cloning parameters to be selected in a cloning template is atmost $(2r+1)^2$. Before using such a network for object extraction we must select the parameters for cloning template that ensure uniform performance for different noise levels. The method is written in algorithmic form as follows.

1. *Create an initial pool.* Generate 'p' random strings of 0's and 1's each of length 'l' to find optimum set of parameters $x_1, x_2, \ldots, x_p$. Each of these strings represent a parameter of the cloning template in a coded form. 'p' strings are concatenated to form a chromosomal/string representation of the parameter set. Generate 'm' such chromosomes to form the initial pool.

2. *Set up cellular neural network with the decoded versions of strings.* Decode the strings representing the parameters of the cloning template and use these to set up a cellular neural network. The gray level range of the input image is scaled to [-1,1] and is given as input to the cellular neural network.

3. *Compute the fitness function (from the obtained and the target images).* The objective function is evaluated using the image obtained from the cellular neural network and the target image.

4. *If optimum value is reached then STOP.*

5. *Reproduce/select strings to create a new mating pool.* The strings are reproduced or selected to create a new mating pool as described earlier.

6. *Generate a new population by crossover and mutation.*

7. *Goto step 2.* ♠

2.4 Computer Simulation and Results

The effectiveness of the algorithms developed in the previous sections were tested for object extraction of gray tone images. Images tested consisted of both real (Figure 23.) synthetic images (Figure 5.).

2.4.1 Simulation of Cellular Neural Network

Simulating a cellular neural network involves choosing the values of circuit parameters and then updating the states. For the present study the cell time constant is 1 micro second. Values of R_x and R_y are assumed to be $10^3\Omega$. Under such a situation, the governing differential equation of the cell C_{ij} is given by

$$\frac{dv_{xij}(t)}{dt} = -10^6 v_{xij}(t) + 10^6[T \star v_{yij}(t) + I_{ij}] \tag{2.23}$$

where T is a cloning template and $\star$ is a convolution operator. Several types of cloning templates have been used for simulation. The following cloning templates are used in the present simulation

$$\begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 2.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}. \tag{2.24}$$

(known as 2-connected or C2 cloning template.)

$$\begin{bmatrix} 0.5 & 1.0 & 0.5 \\ 1.0 & 2.0 & 1.0 \\ 0.5 & 1.0 & 0.5 \end{bmatrix}. \tag{2.25}$$

(known as 3-connected or C3 cloning template.)

$$\begin{bmatrix} 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 1.0 & 2.0 & 1.0 & 0.0 \\ 0.5 & 2.0 & 4.0 & 2.0 & 0.5 \\ 0.0 & 1.0 & 2.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \end{bmatrix}. \tag{2.26}$$

(known as 4-connected or C4 cloning template.)

For an image of size $M \times N$ we need a network of size $M \times N$. Gray value of each pixel is scaled into [-1,1] and given as input to each cell of the network. The differential equations governing the dynamics of cells are iterated till the circuit reaches a stable state. After the network has stabilized the outputs of cells are found to be either -1 or +1, representing whether the pixel belongs to object or background region. ♣

Cellular neural network can also be used for extraction of boundaries of object regions in gray images. For this purpose we use a feed forward cloning template of the form

$$\begin{bmatrix} 0.0 & -1.0 & 0.0 \\ -1.0 & 4.0 & -1.0 \\ 0.0 & -1.0 & 0.0 \end{bmatrix} \tag{2.27}$$

and bias $I_{ij} = -2 \times 10^3$.

2.4.2 Results of Simulation

Two simulation experiments were conducted to determine the effect of the cell bias I_{ij} .

1. Bias term, $I_{ij} = 0$ (Zero bias).
2. Bias term, $I_{ij} = \bar{v}_{uij}$ (Proportional bias).

Images of size (128 × 128) of geometric objects, corrupted by various amounts of Gaussian noise (Figures 6-8.) were considered for the study. The study was conducted using C2 and C3 cloning templates. The results of the simulation on the noisy images with zero bias using C2 and C3 cloning templates is shown in Figures 9-10, Figures 12-13., and Figures 16-17. Table 2.1 gives the percentage of correct classification achieved at various σ (standard deviation of gaussian noise) levels. The results indicate the deterioration of the percentage of correct classification at

Table 2.1: Effect of bias on classification accuracy (in percentage)

σ	C2		C3	
	Proportional bias	Zero bias	Proportional bias	Zero bias
10	98.97	99.71	99.76	99.74
15	98.56	98.93	89.46	99.31
20	98.02	96.16	87.30	98.26
24	97.74	92.58	78.37	96.04
32	94.69	84.46	70.20	89.62

Table 2.2: Percentage of correct classification using $C2$,$C3$ and $C4$ cloning templates

σ	C2	C3	C4
10	99.71	99.74	99.80
15	98.93	99.31	99.46
20	96.16	98.26	98.61
24	92.58	96.04	97.38
32	84.46	89.62	91.88

higher noise levels in case of proportional bias, which is not so significant in case of zero bias.

Another experiment was conducted to determine the effect of size of cloning template on the classification accuracy. This study was conducted using C2, C3 and C4 cloning templates (Figures 9-17.). The results are shown in Table 2.2. This experiment was repeated on a real image (Figures 24-26.). The classification accuracy increases with the size of the cloning template for thick and compact objects. Thin and elongated regions are destroyed by using the cloning templates of larger size, due to greater neighborhood effects. The cost of realizing the circuit also increases as the square of size of the neighborhood. It may therefore be pragmatic to choose cloning templates depending upon the need. Extraction of boundaries of object regions has been accomplished using the present network (Figure 28.).

Table 2.3: Percentage of correct classification using $C3$ and GA (cloning template obtained by using Genetic Algorithm) cloning templates

σ	C3	GA
10	99.74	98.97
15	99.31	98.56
20	98.26	98.02
24	96.04	97.74
32	89.62	94.69

Genetic algorithms were used to obtain optimum cloning template parameters of size C3 automatically. The image of geometric objects (size 128×128) requires higher computation (nearly 16 times) than the image of character 'B' (size 32×32) when simulating on digital computer. Hence for the selection of cloning template parameter we have used the image of character 'B' (Figures 18-19.). Using the algorithm described in section 2.3.2 we have obtained a cloning template of size C3 after 50 generations. This cloning template is shown below

$$\begin{bmatrix} 0.37 & 0.97 & 0.51 \\ 0.94 & 0.95 & 0.95 \\ 0.54 & 0.92 & 0.47 \end{bmatrix}. \tag{2.28}$$

(henceforth this cloning template will be called GA).

This cloning template was applied on the images of geometric objects corrupted by noise, the results are given in Table 2.3. The cloning template obtained by using genetic algorithms was found to perform uniformly well under different signal to noise ratios. The images were also found to be better visually than those obtained from the cloning template designed with analogy from digital filters (Figures 20-22. and Figure 27.).

The results obtained using other existing neural network based techniques [13, 14, 15] are shown in Table 2.4. The performance of the present network shows a

Table 2.4: Performance (percentage of correct classification) of cellular neural network compared with existing neural network methods

σ	Cellular neural network		Hopfield	Self-Organizing
	C3	GA		
10	**99.71**	98.97	99.71	99.58
20	**99.26**	98.02	99.12	98.90
32	**89.62**	94.69	97.55	97.17

marginally less classification accuracy as compared to self-organizing neural network and hopfield network (the neighborhood size in all the three cases is same). ♠

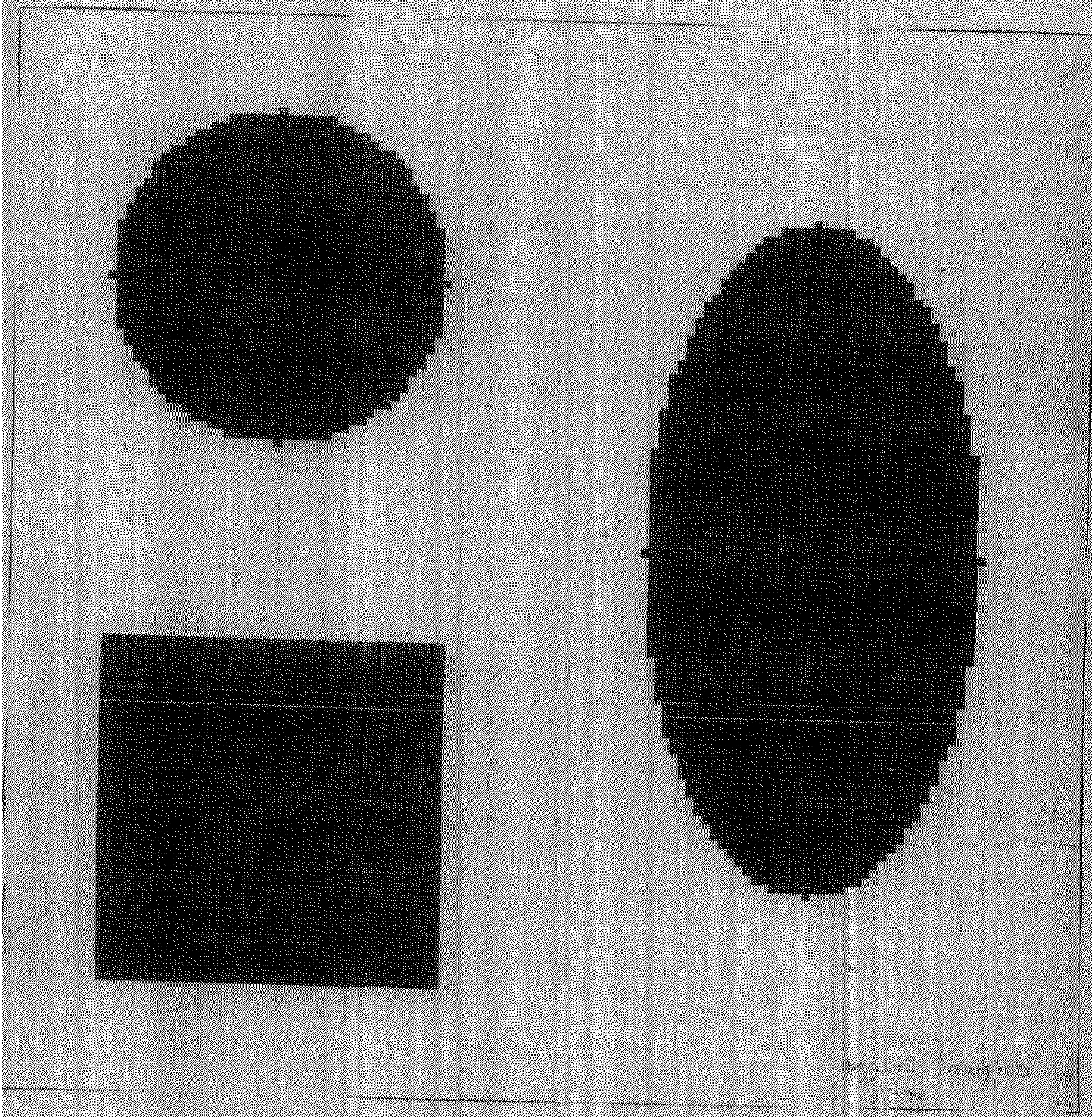

Figure 5. Original image of geometric objects (128×128).

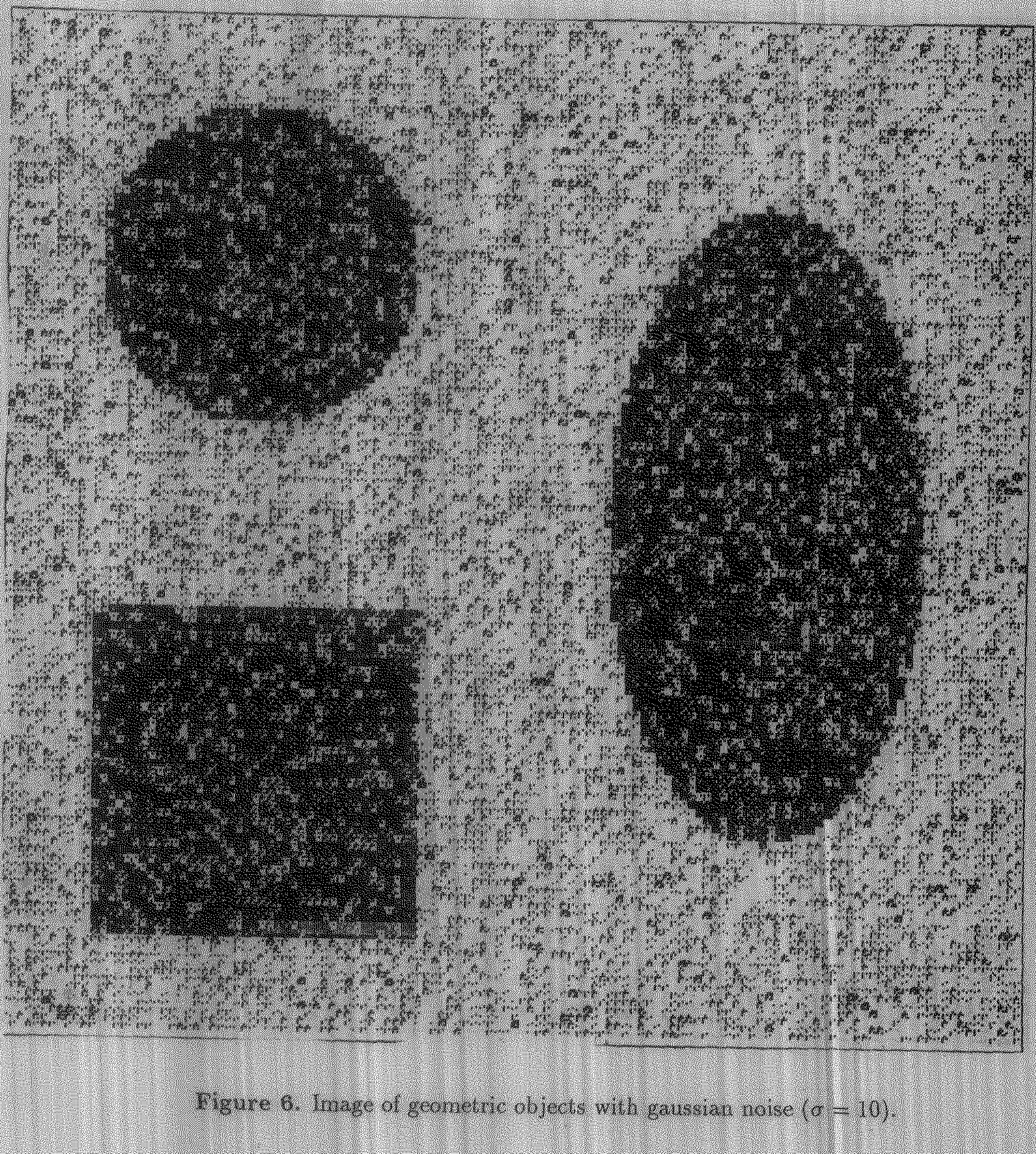

Figure 6. Image of geometric objects with gaussian noise ($\sigma = 10$).

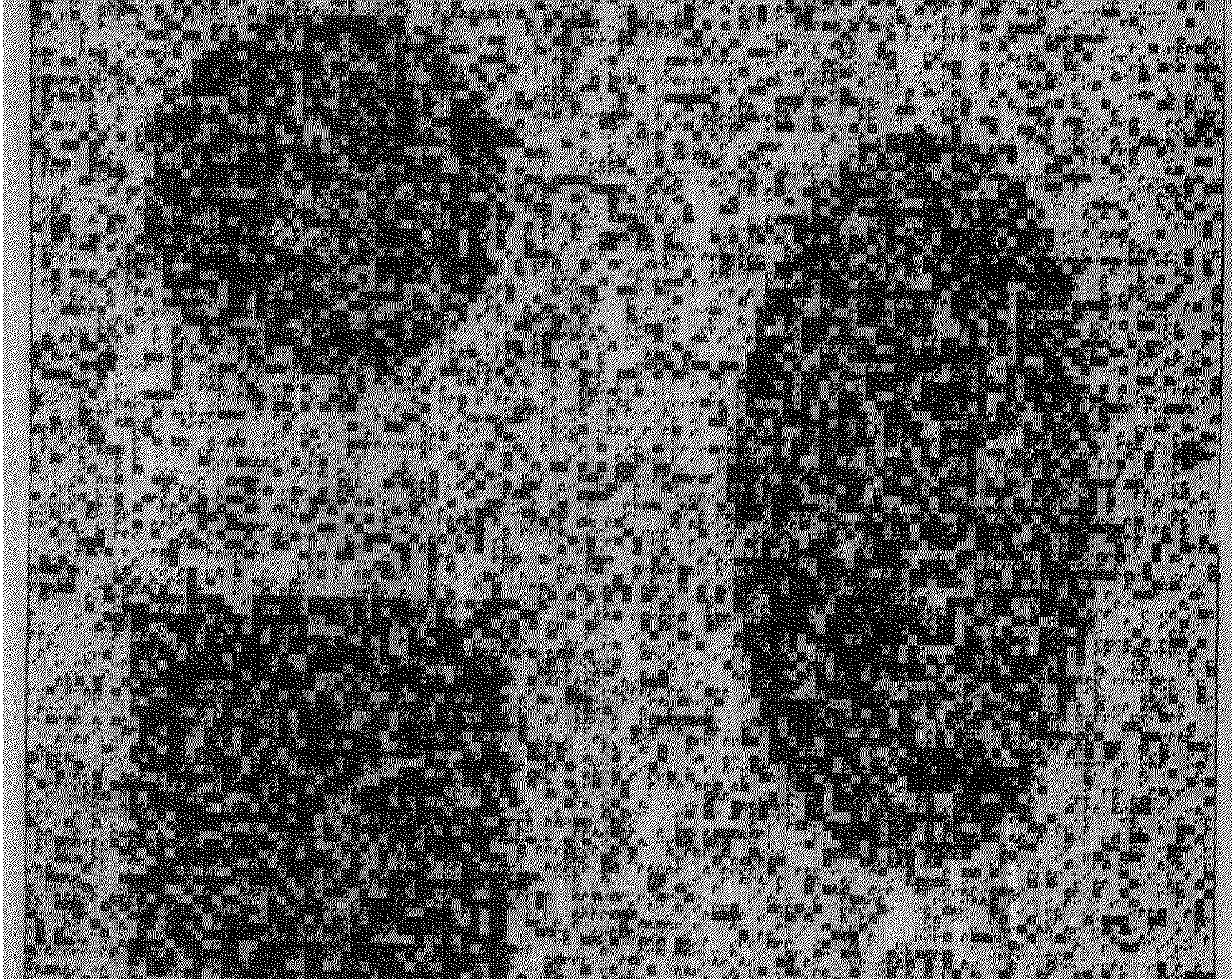

Figure 7. Image of geometric objects with gaussian noise ($\sigma = 20$).

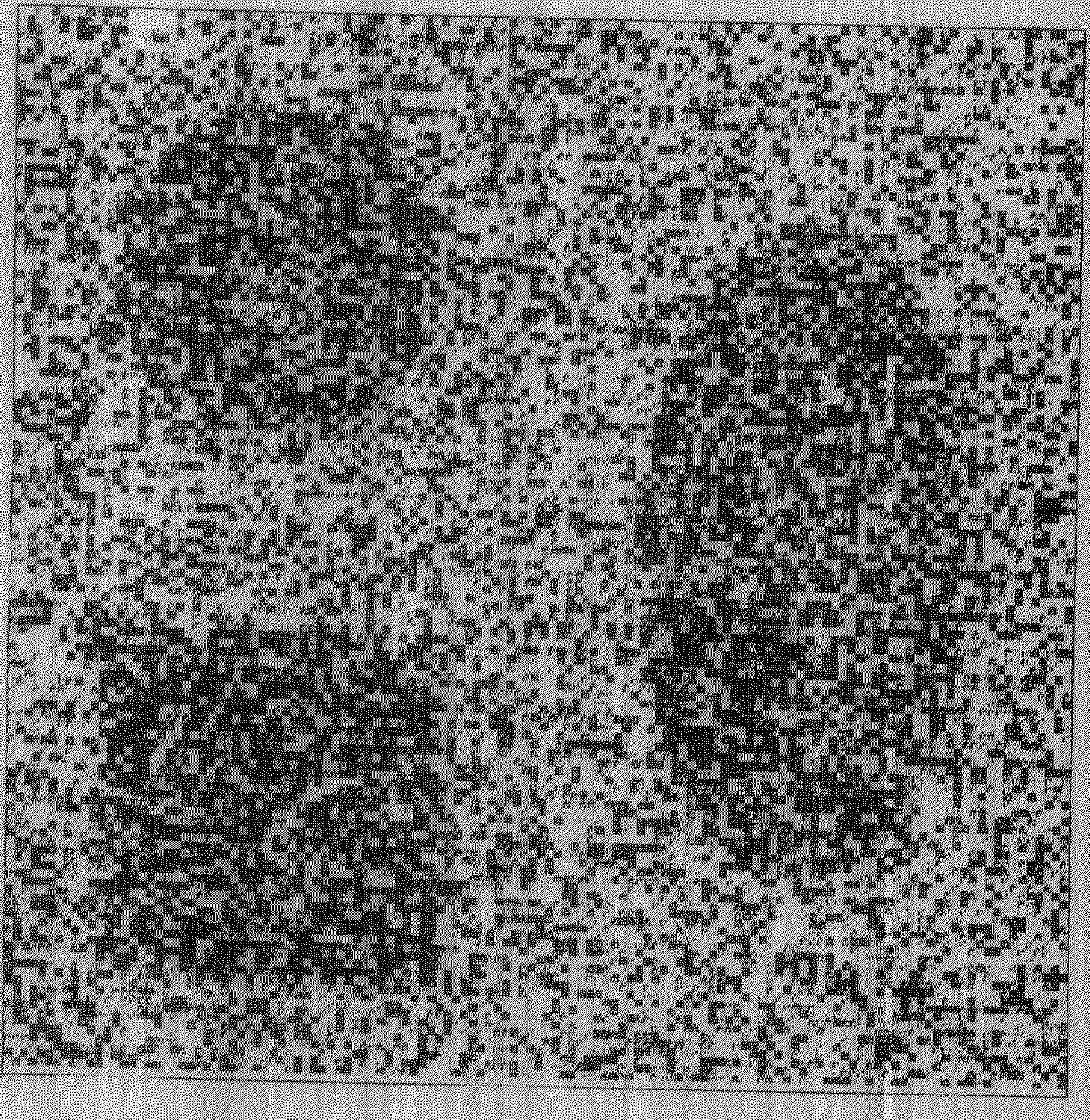

Figure 8. Image of geometric objects with gaussian noise ($\sigma = 32$).

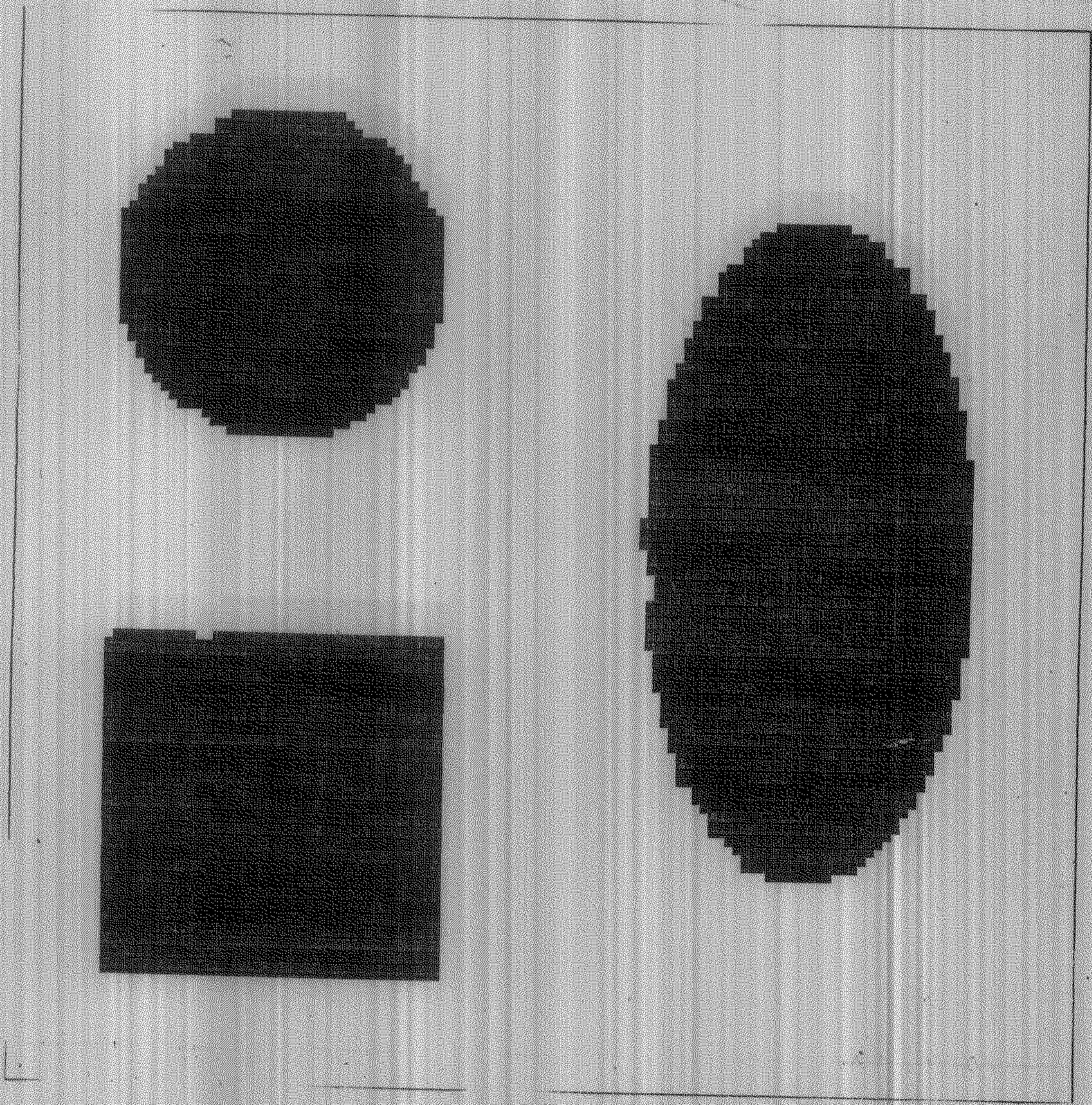

Figure 9. Extracted objects from noisy image($\sigma = 10$) using C2.

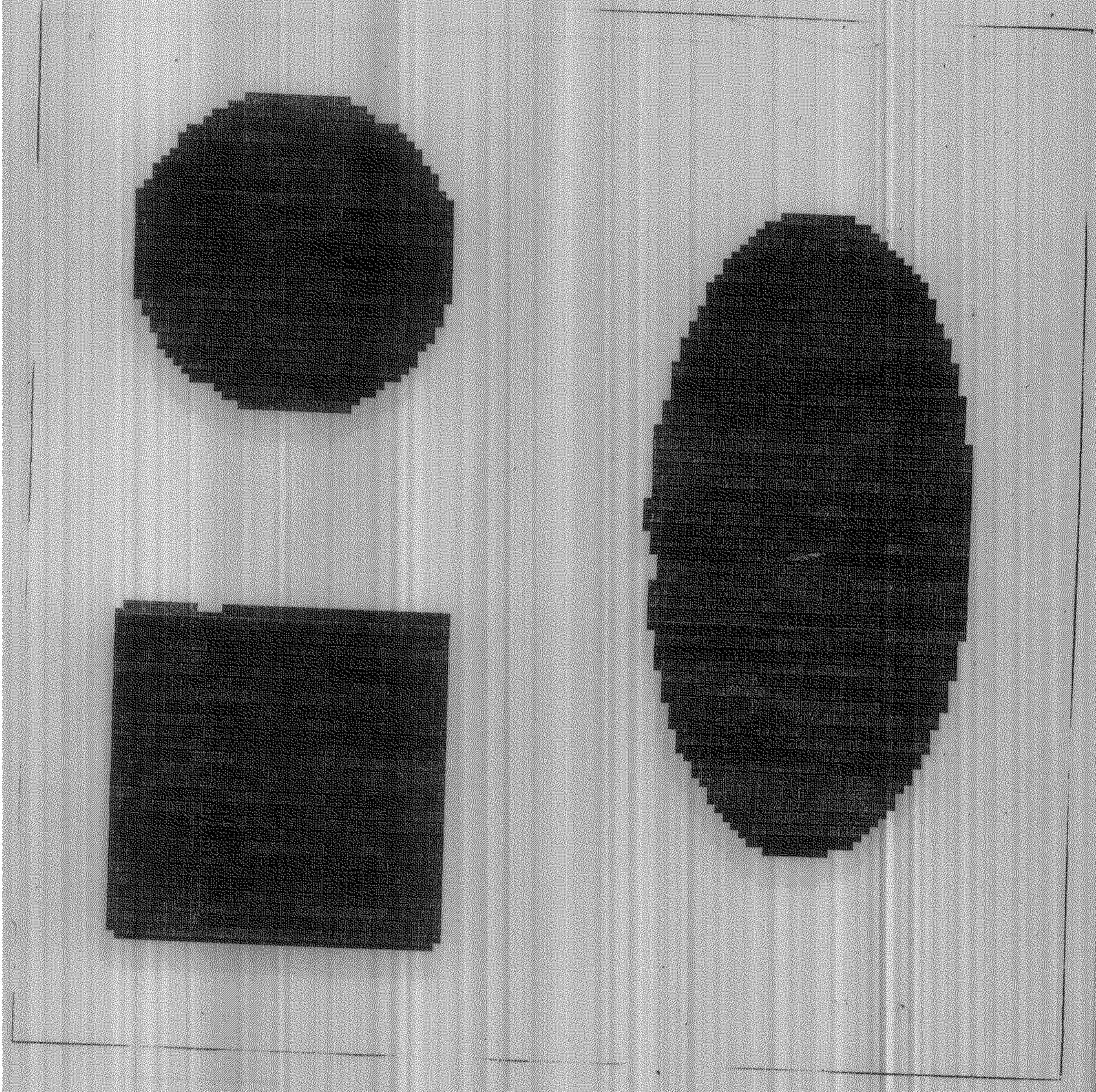

Figure 10. Extracted objects from noisy image($\sigma = 10$) using C3.

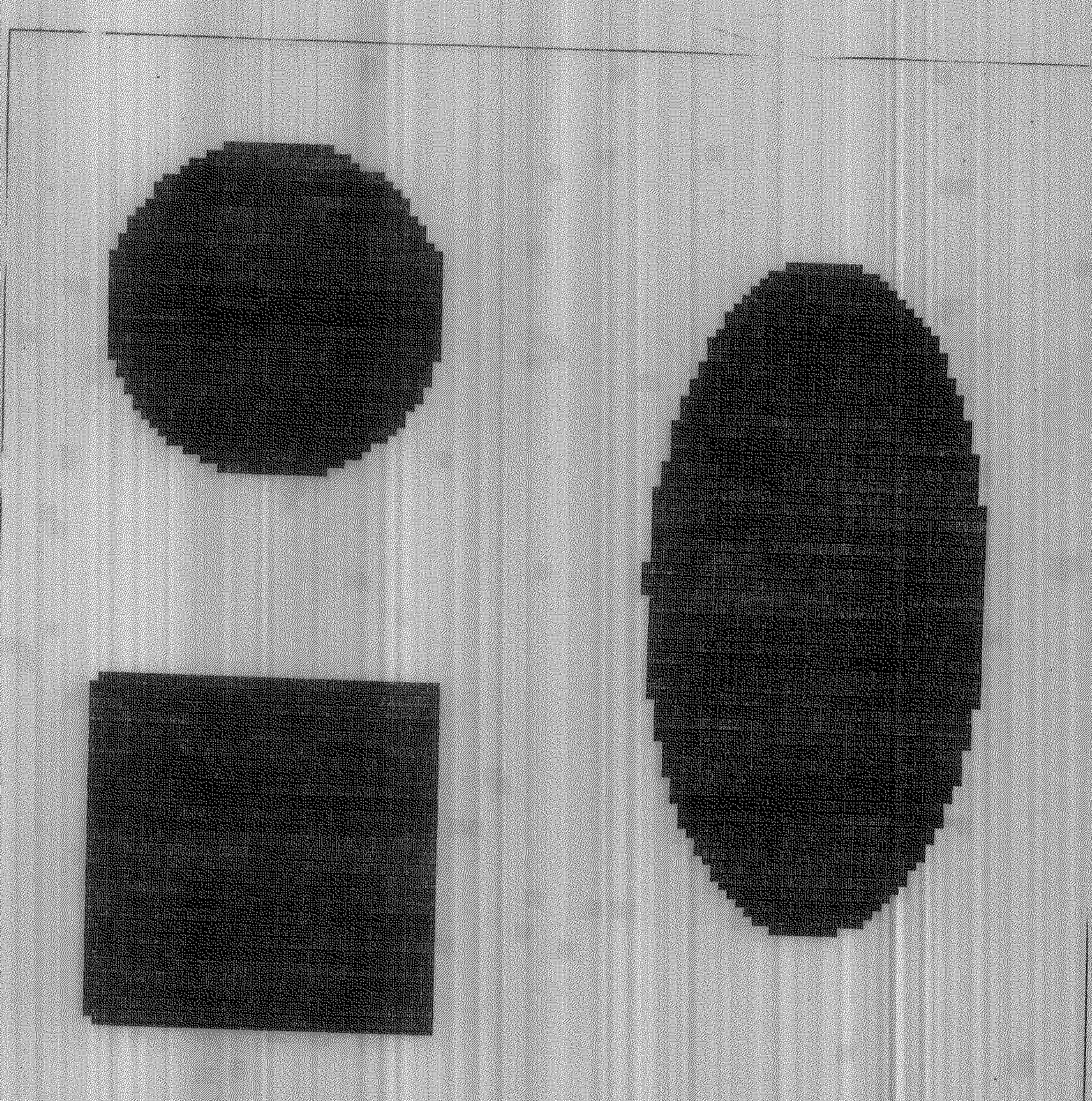

Figure 11. Extracted objects from noisy image($\sigma = 10$) using C4.

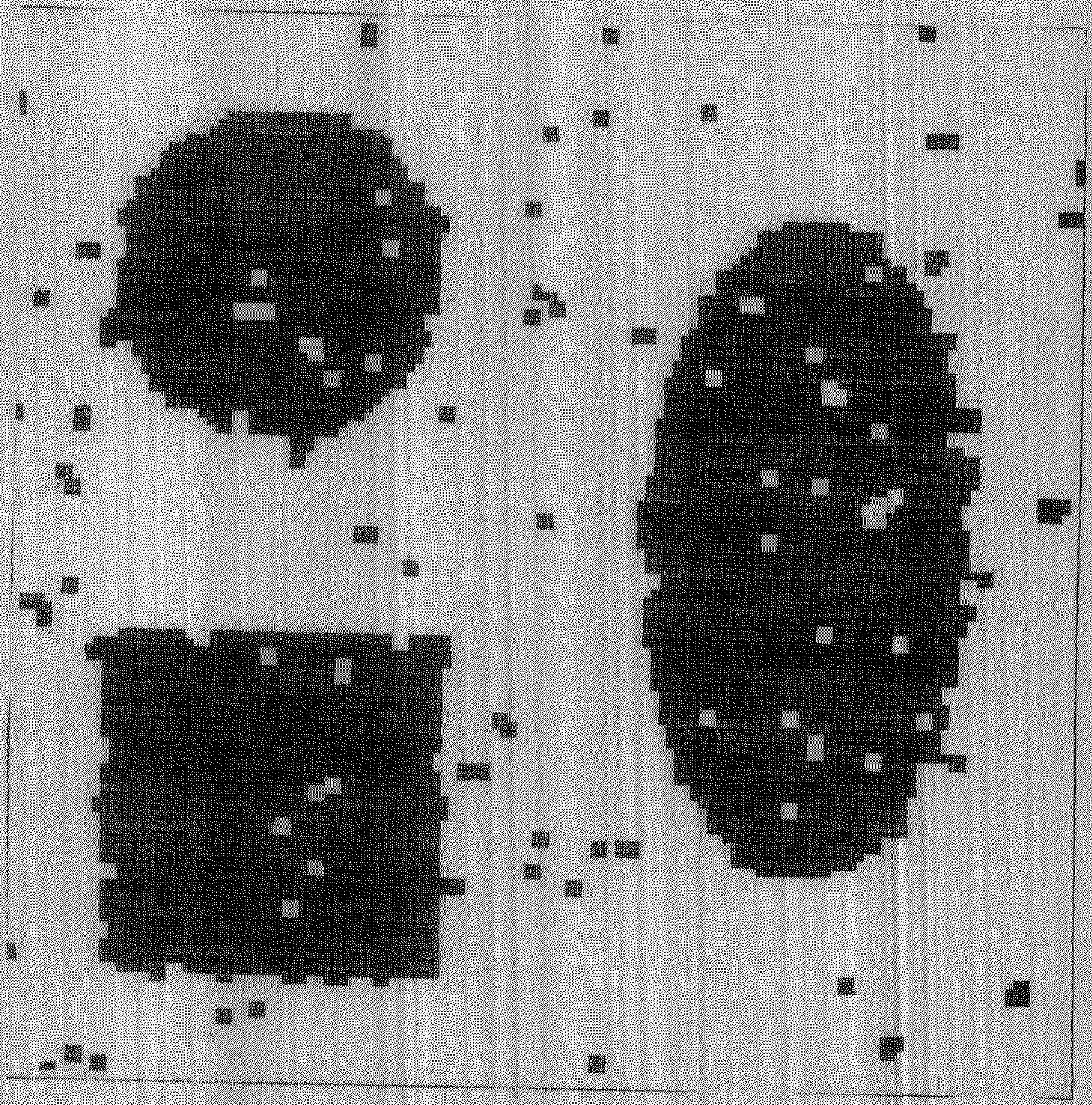

Figure 12. Extracted objects from noisy image($\sigma = 20$) using C2.

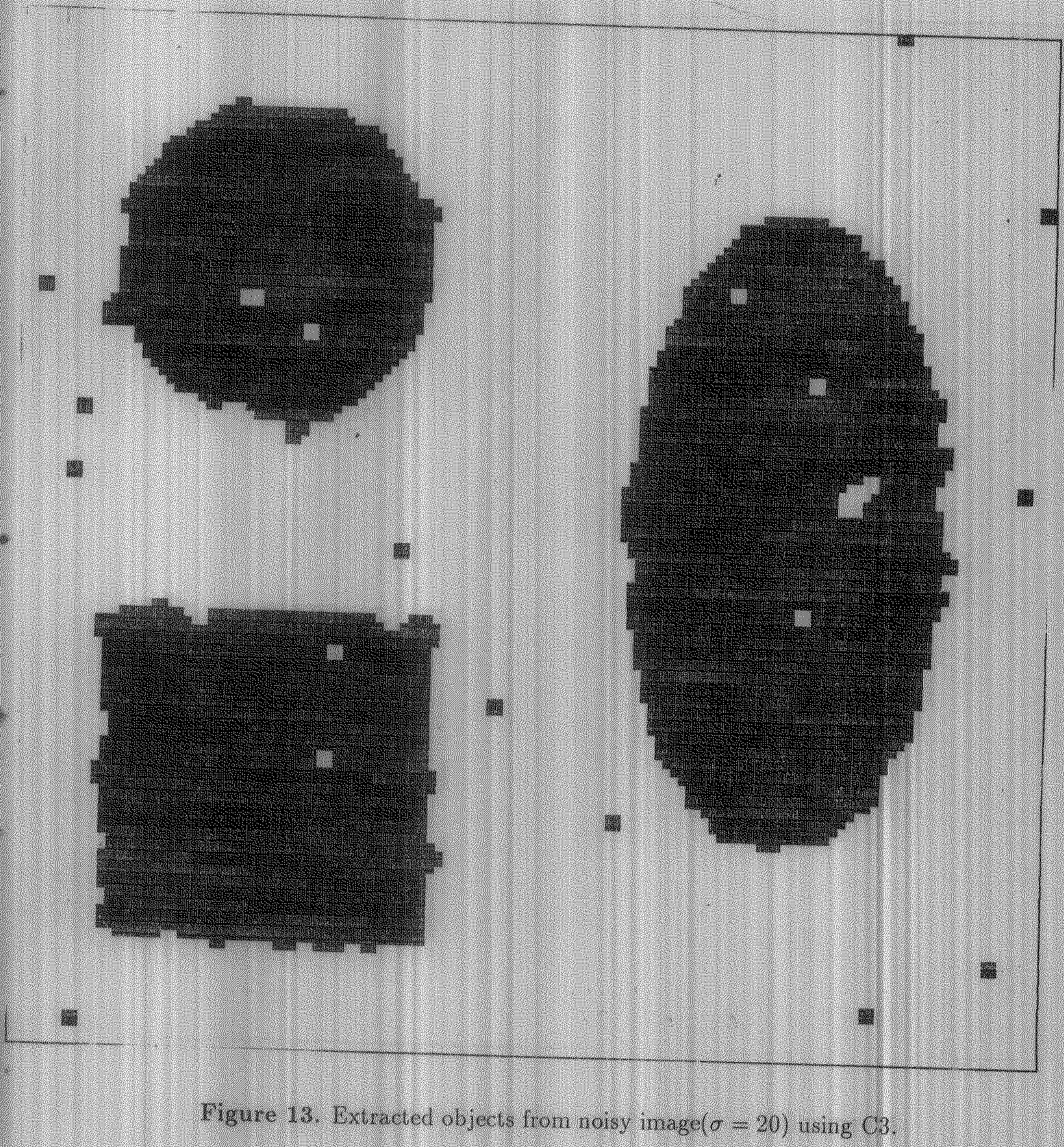

Figure 13. Extracted objects from noisy image($\sigma = 20$) using C3.

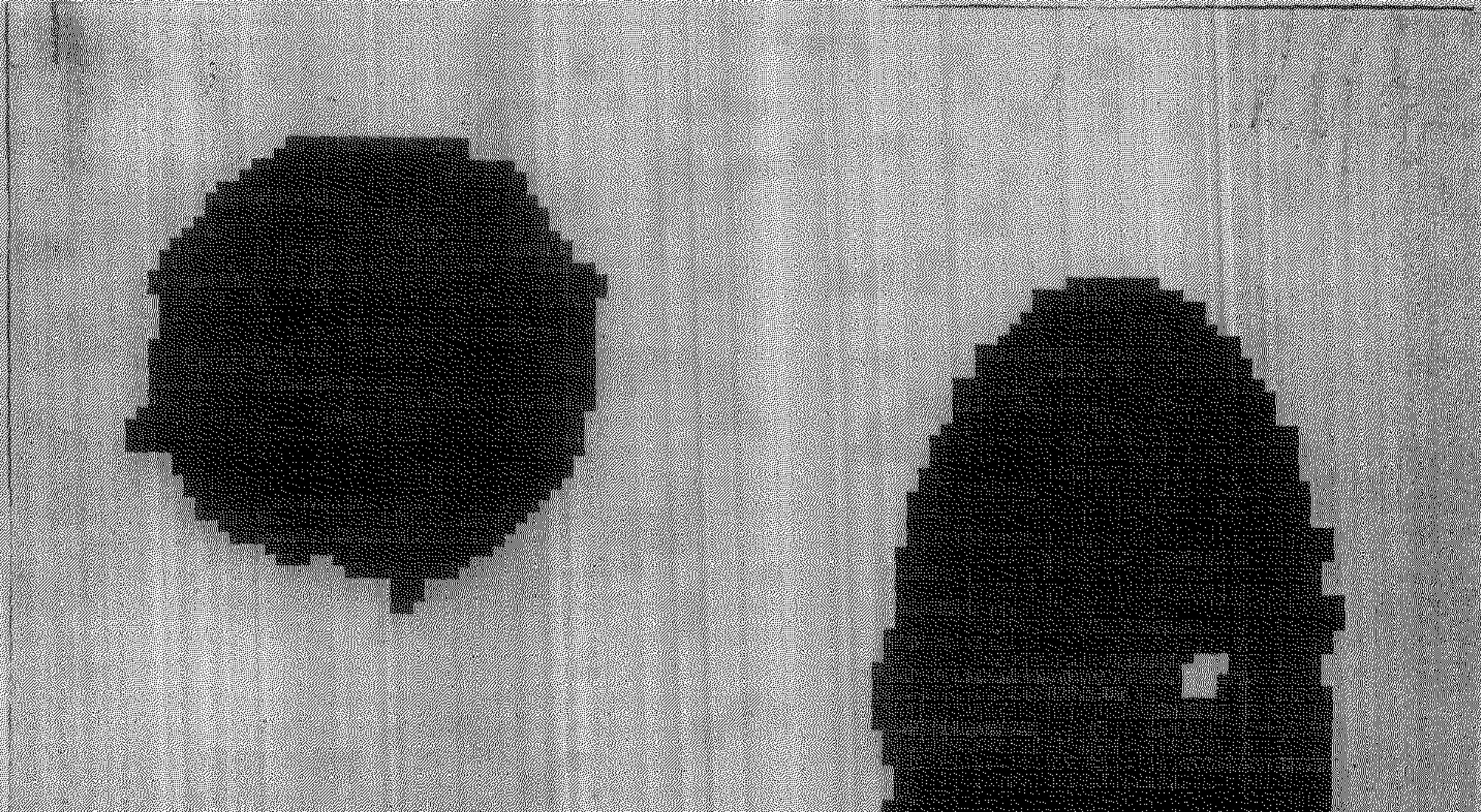

Figure 14. Extracted objects from noisy image($\sigma = 20$) using C4.

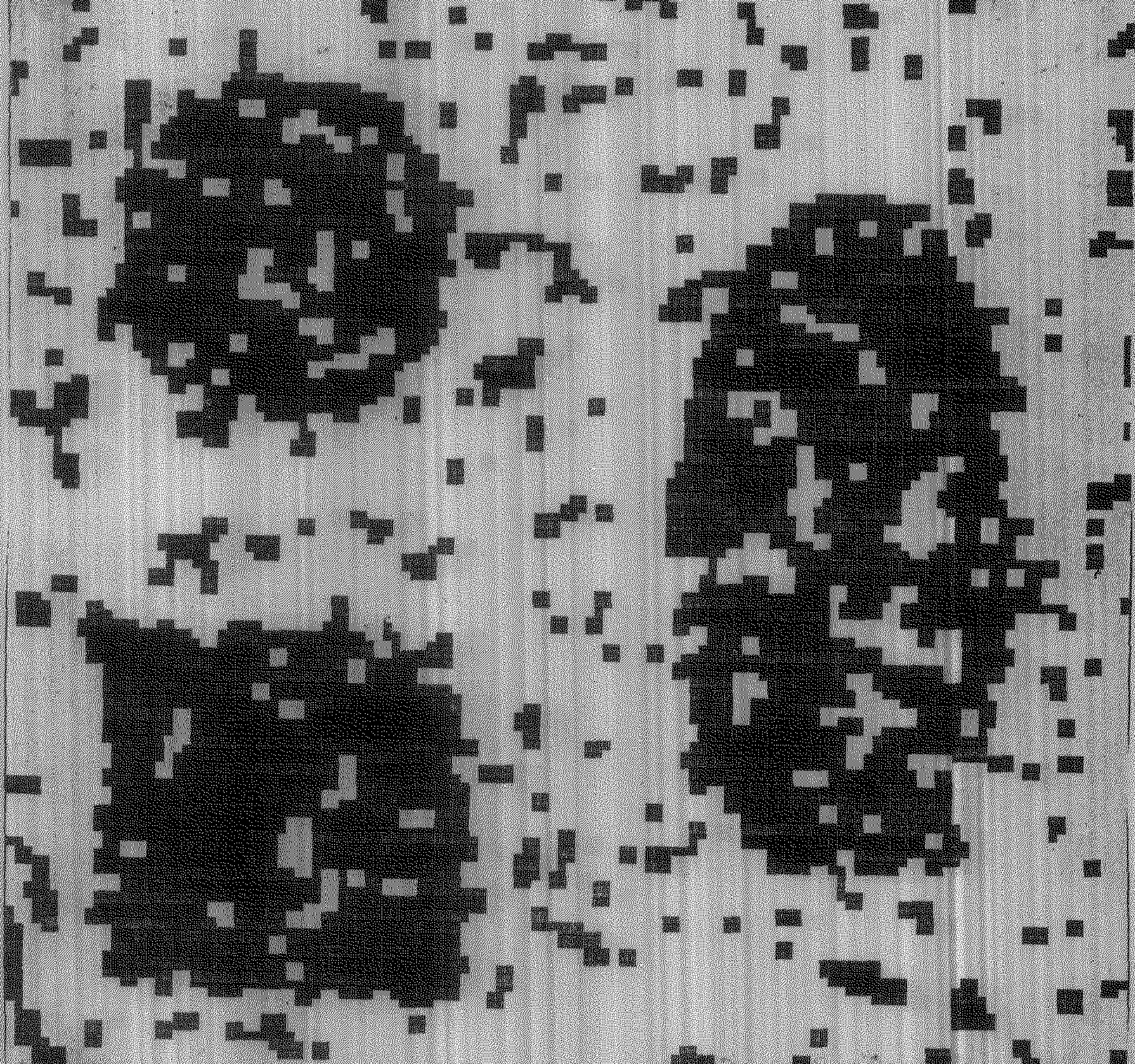

Figure 15. Extracted objects from noisy image($\sigma = 32$) using C2.

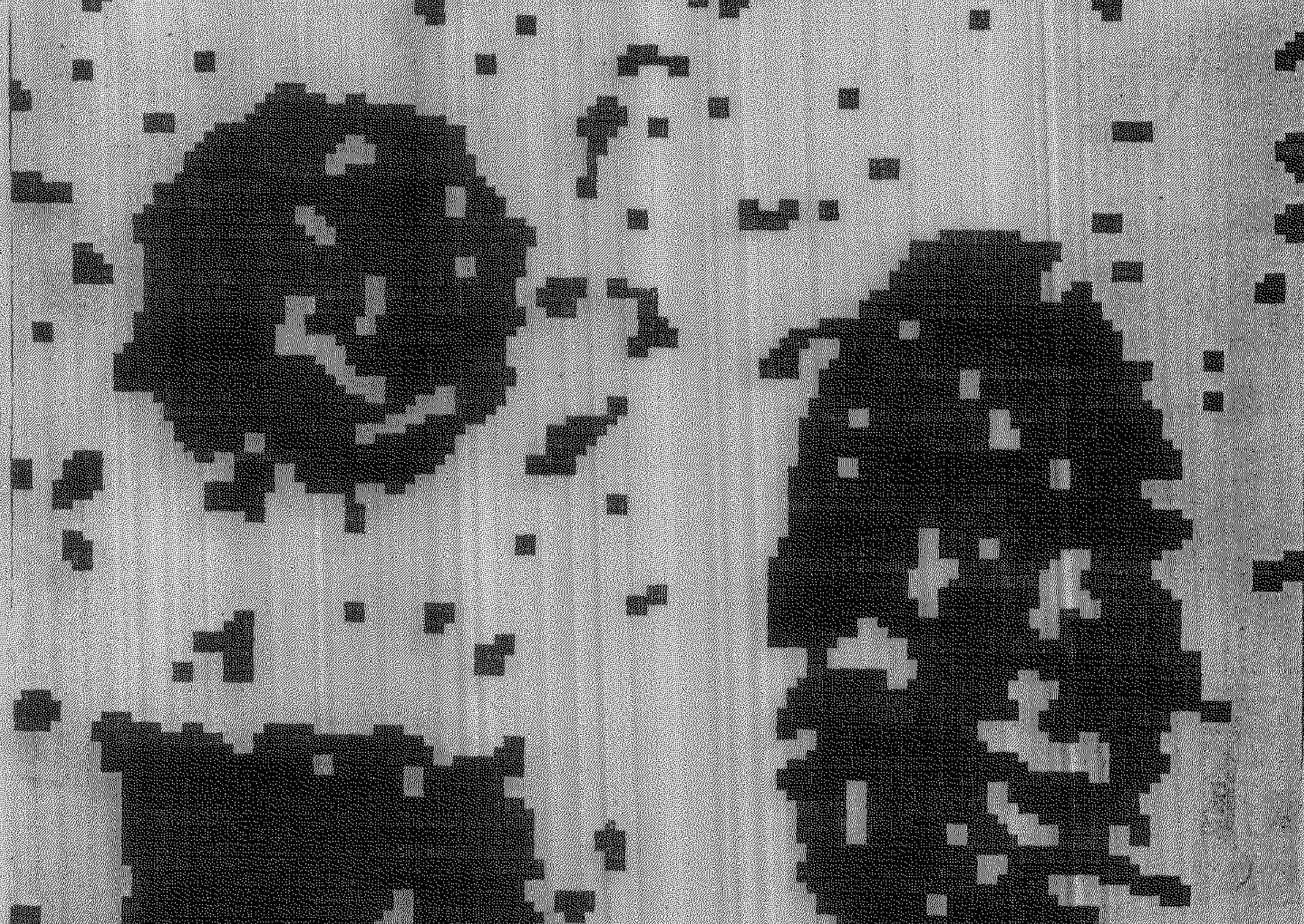

Figure 16. Extracted objects from noisy image($\sigma = 32$) using C3.

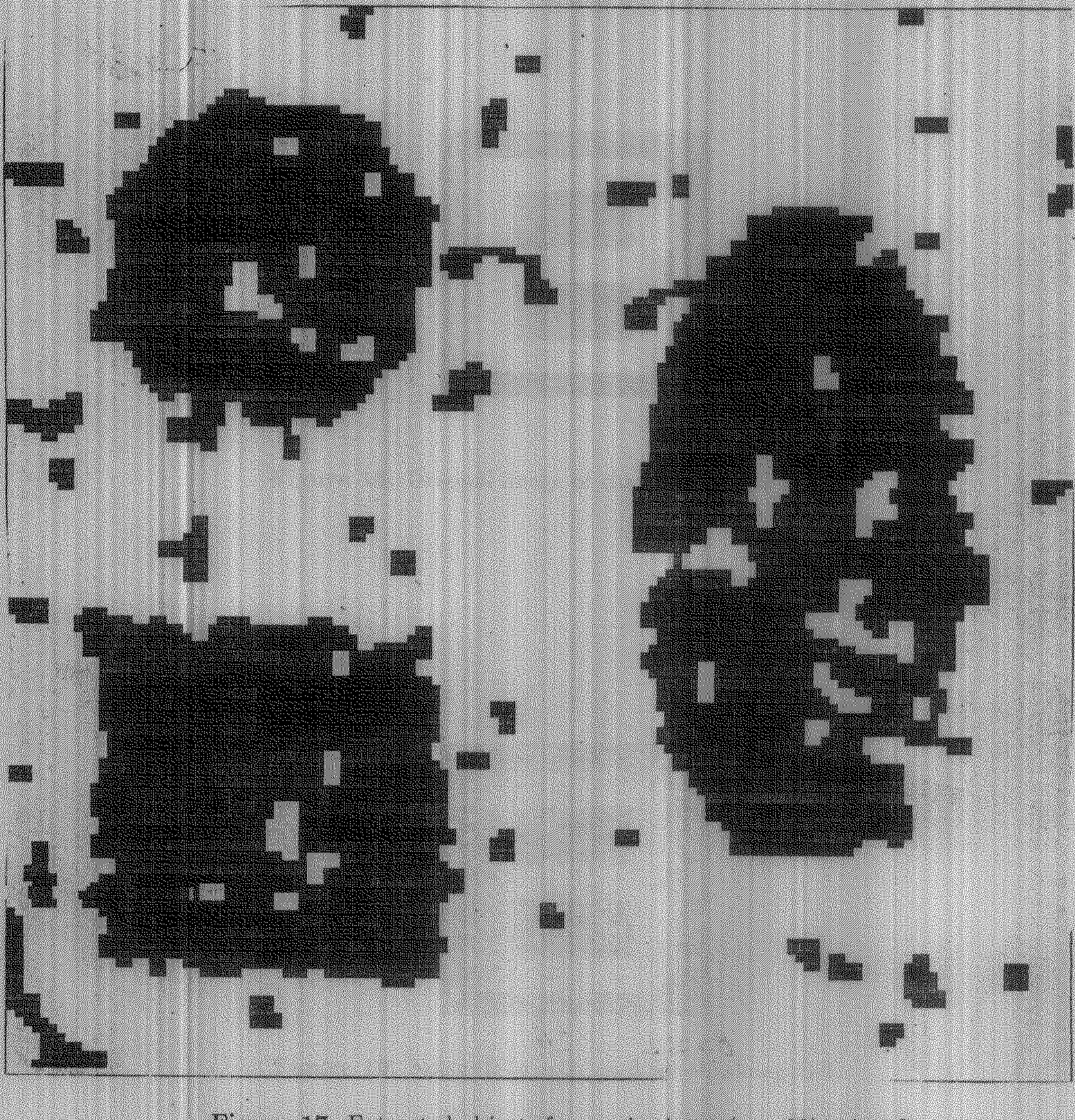

Figure 17. Extracted objects from noisy image($\sigma = 32$) using C4.

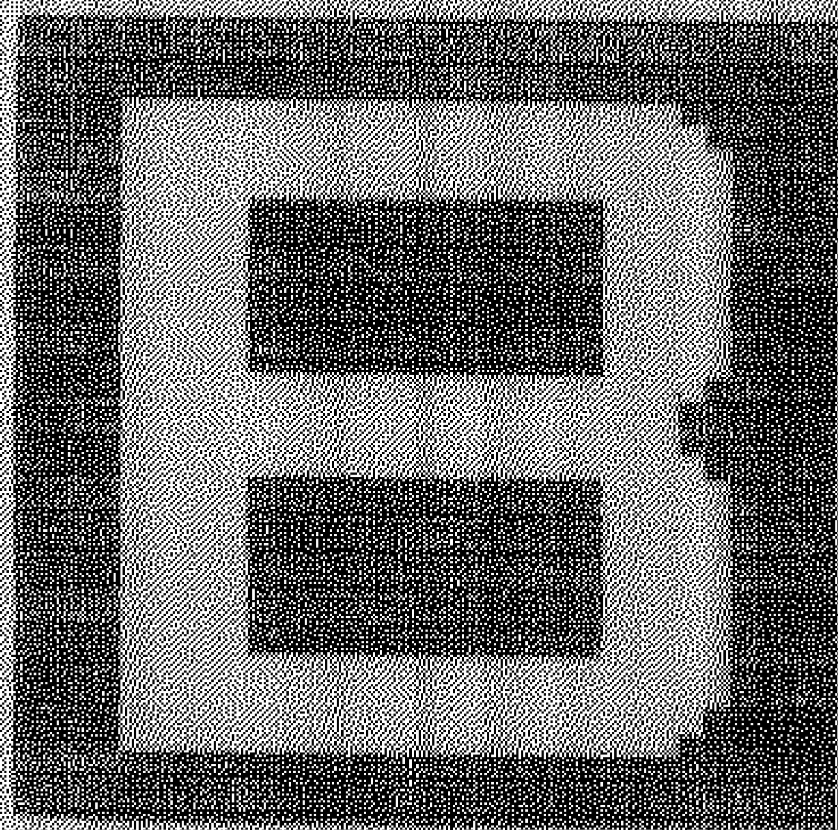

Figure 18. Original image of character 'B'(32×32).

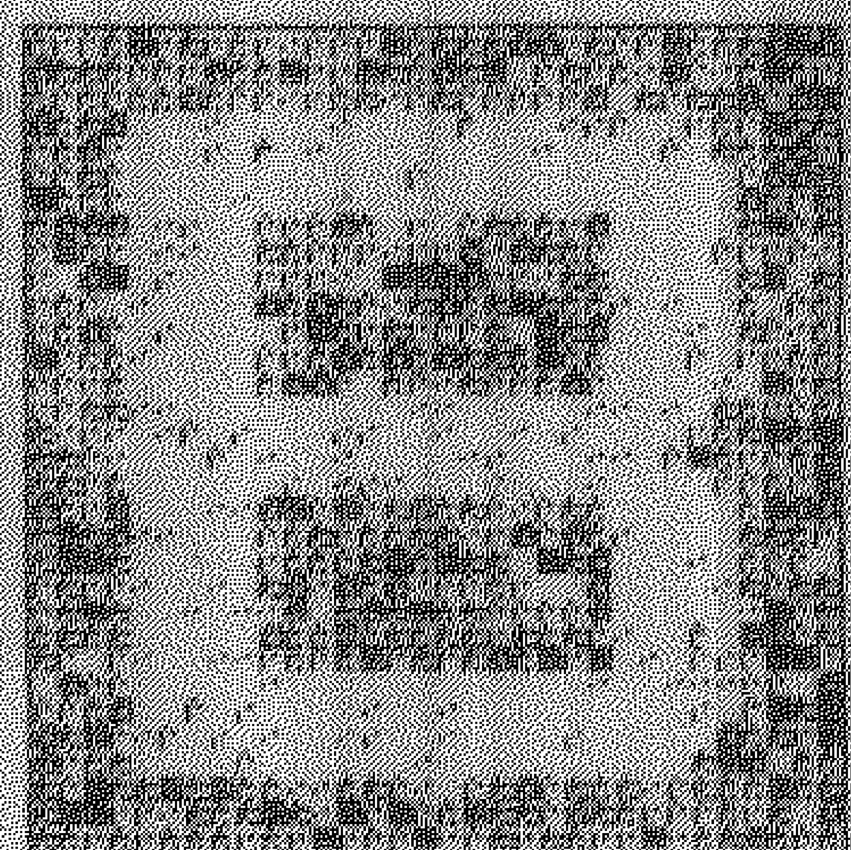

Figure 19. Noisy 'B'($\sigma = 2$) used for obtaining GA.

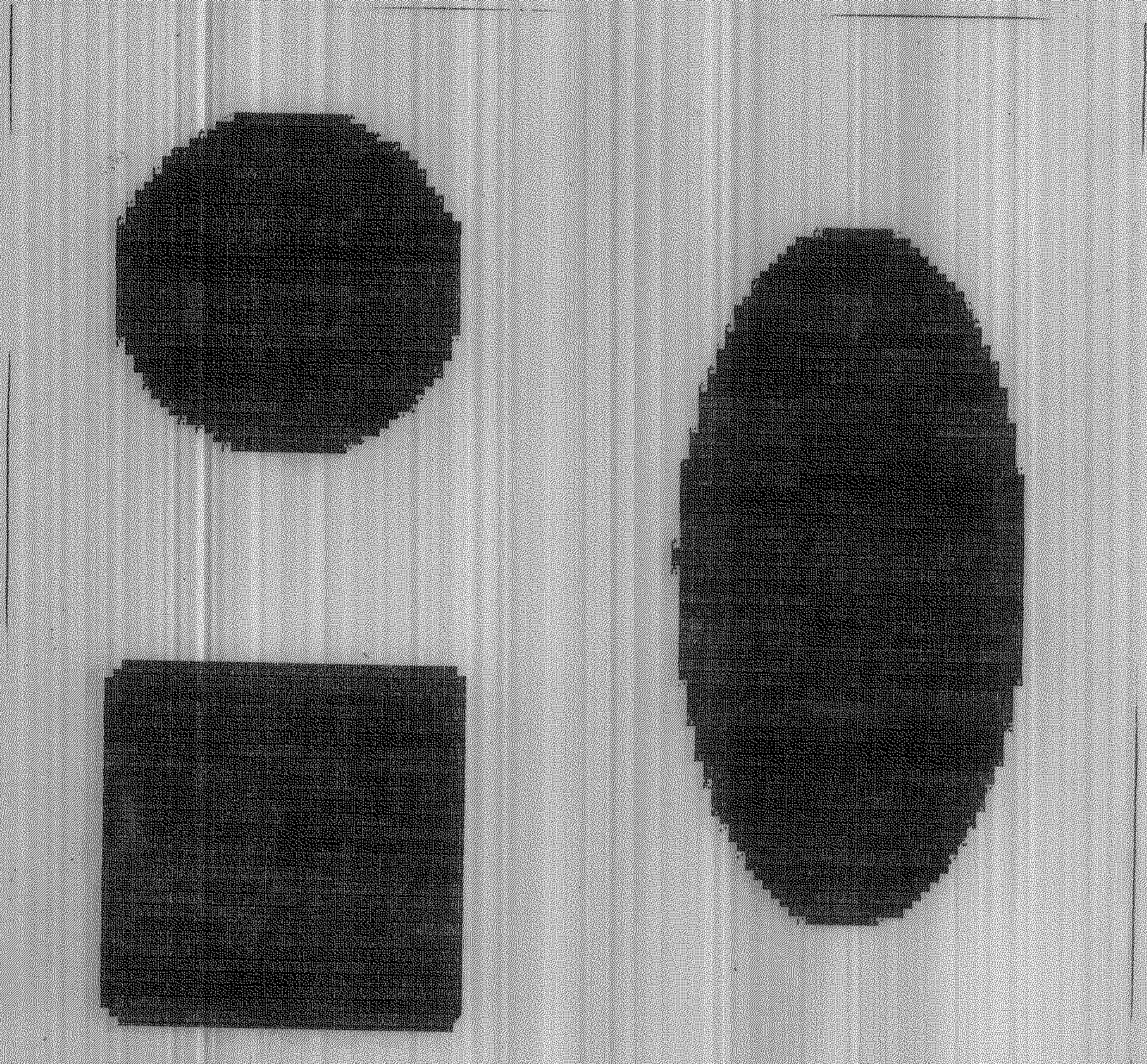

Figure 20. Extracted objects from noisy image($\sigma = 10$) using GA.

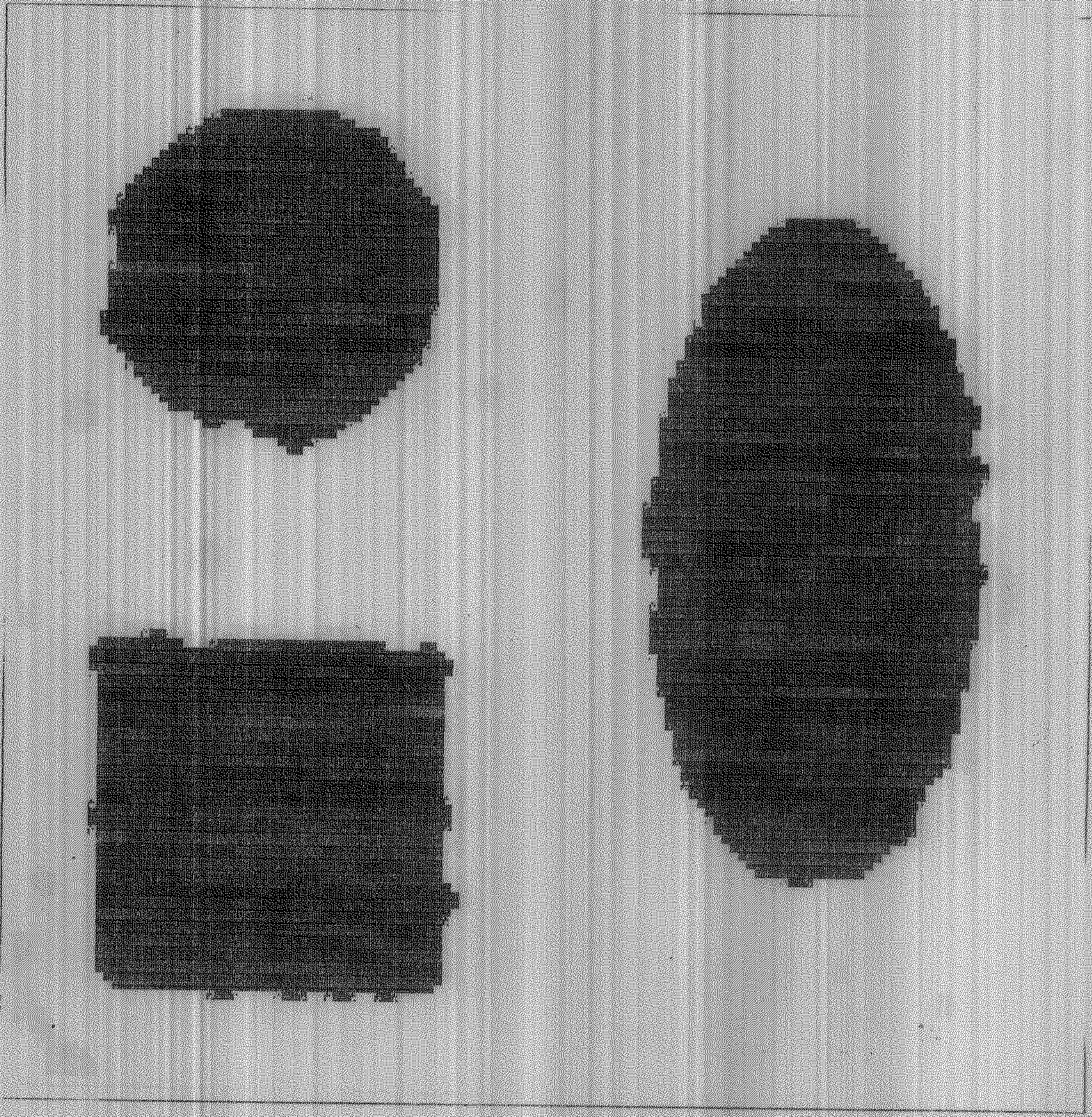

Figure 21. Extracted objects from noisy image($\sigma = 20$) using GA.

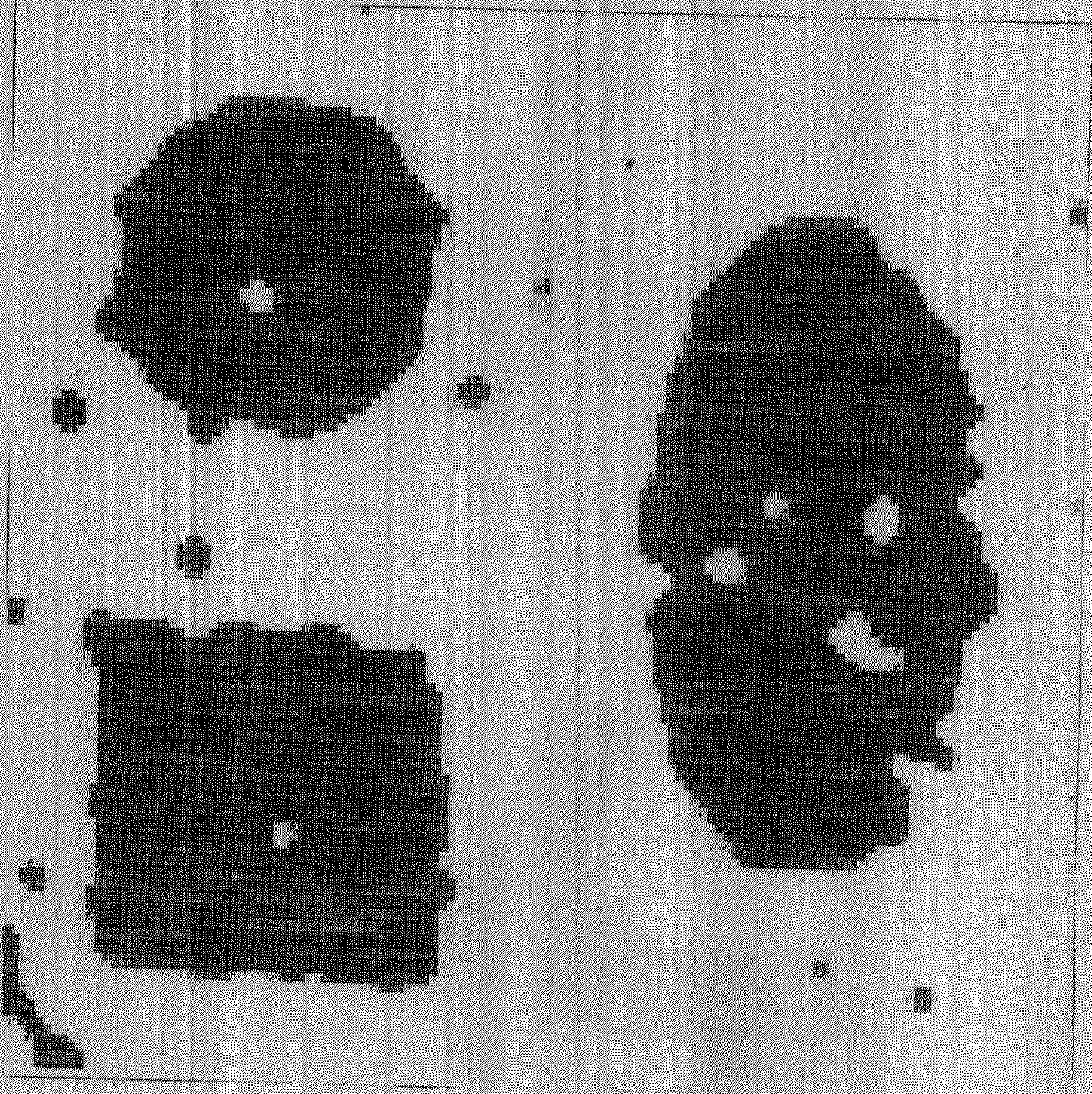

Figure 22. Extracted objects from noisy image($\sigma = 32$) using GA.

Figure 23. Original image of BIPLANE (64×64).

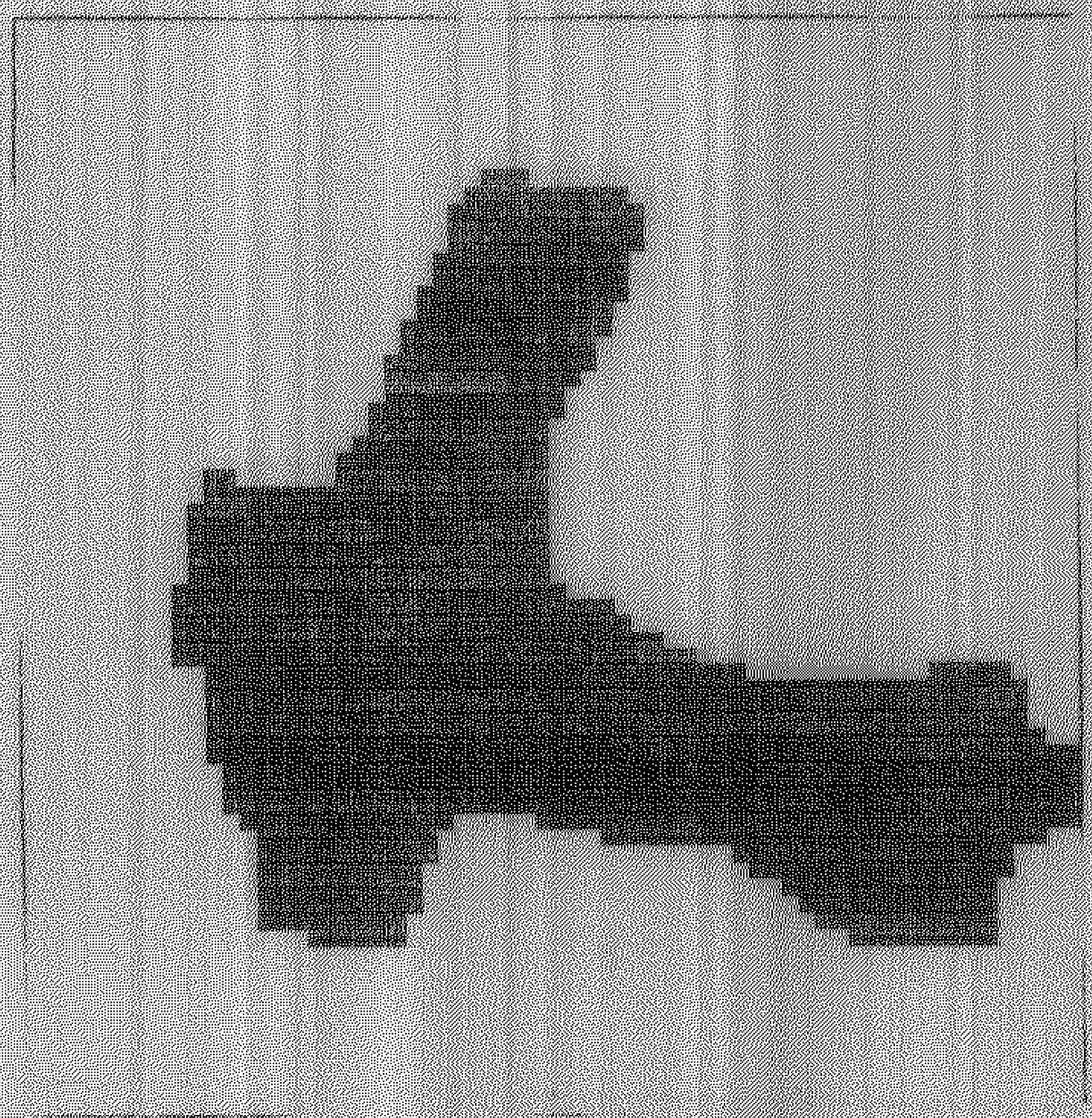

Figure 24. Extracted object from BIPLANE image using C2.

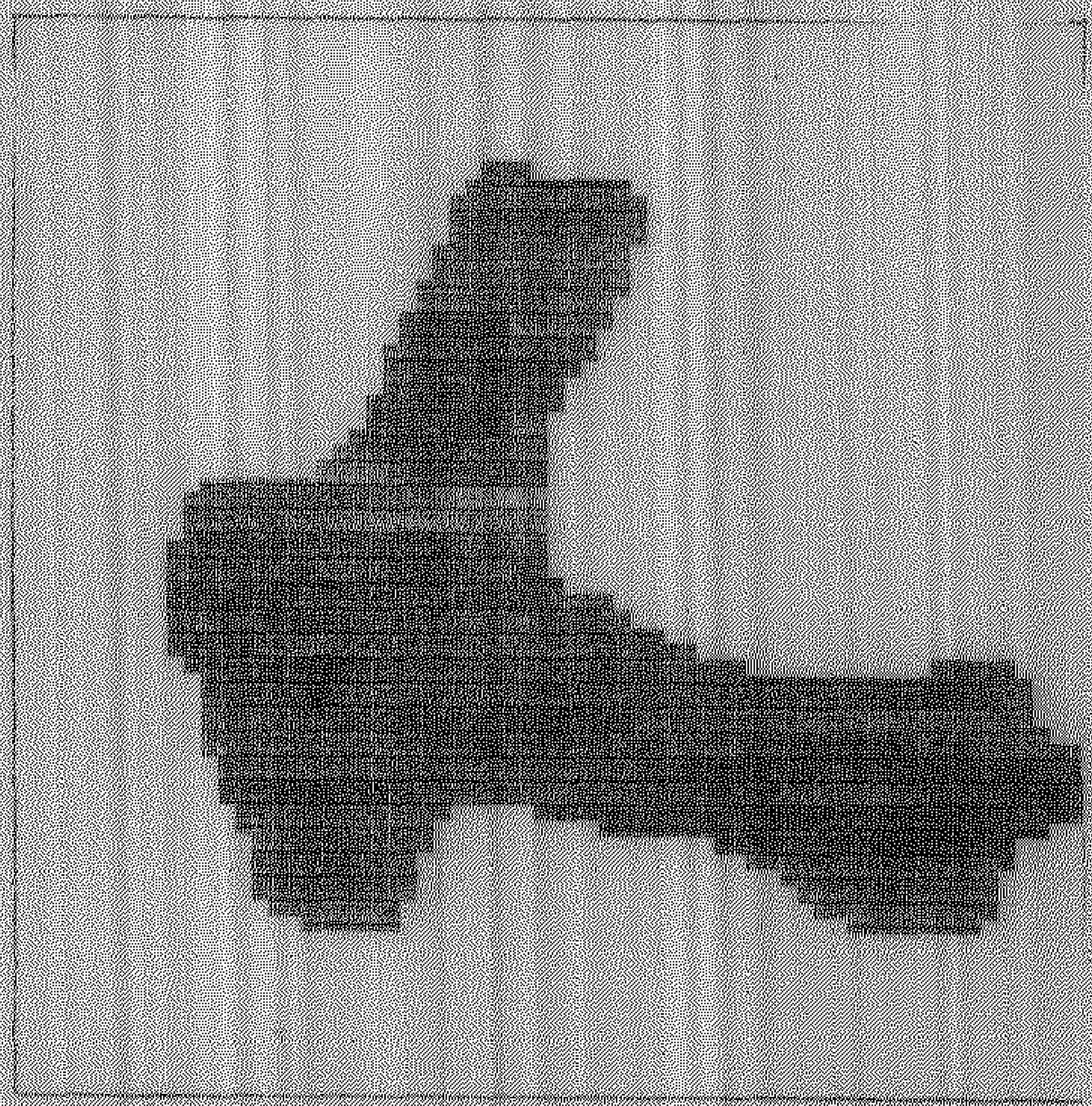

Figure 25. Extracted objects from BIPLANE image using C3.

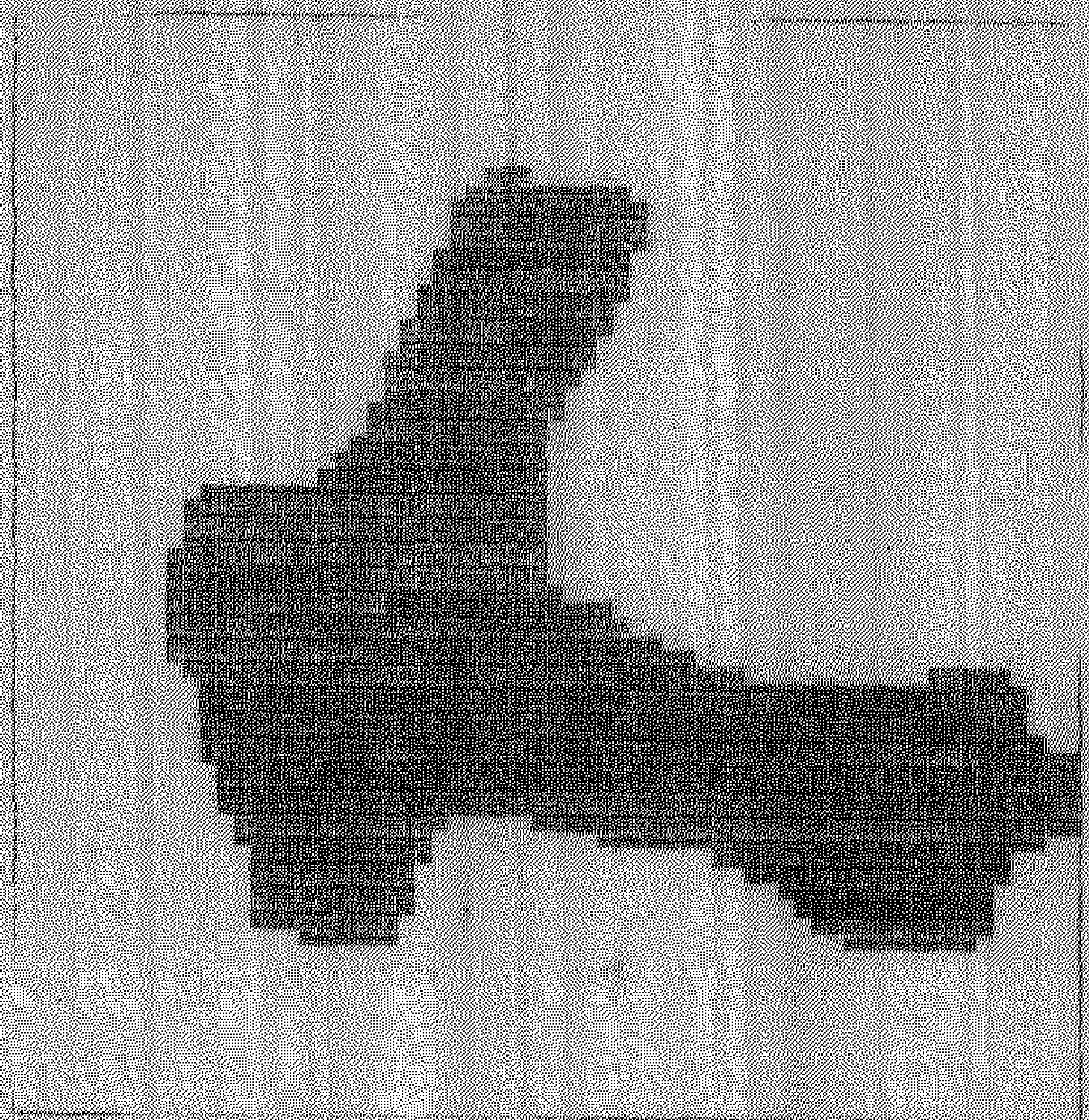

Figure 26. Extracted objects from BIPLANE image using C4.

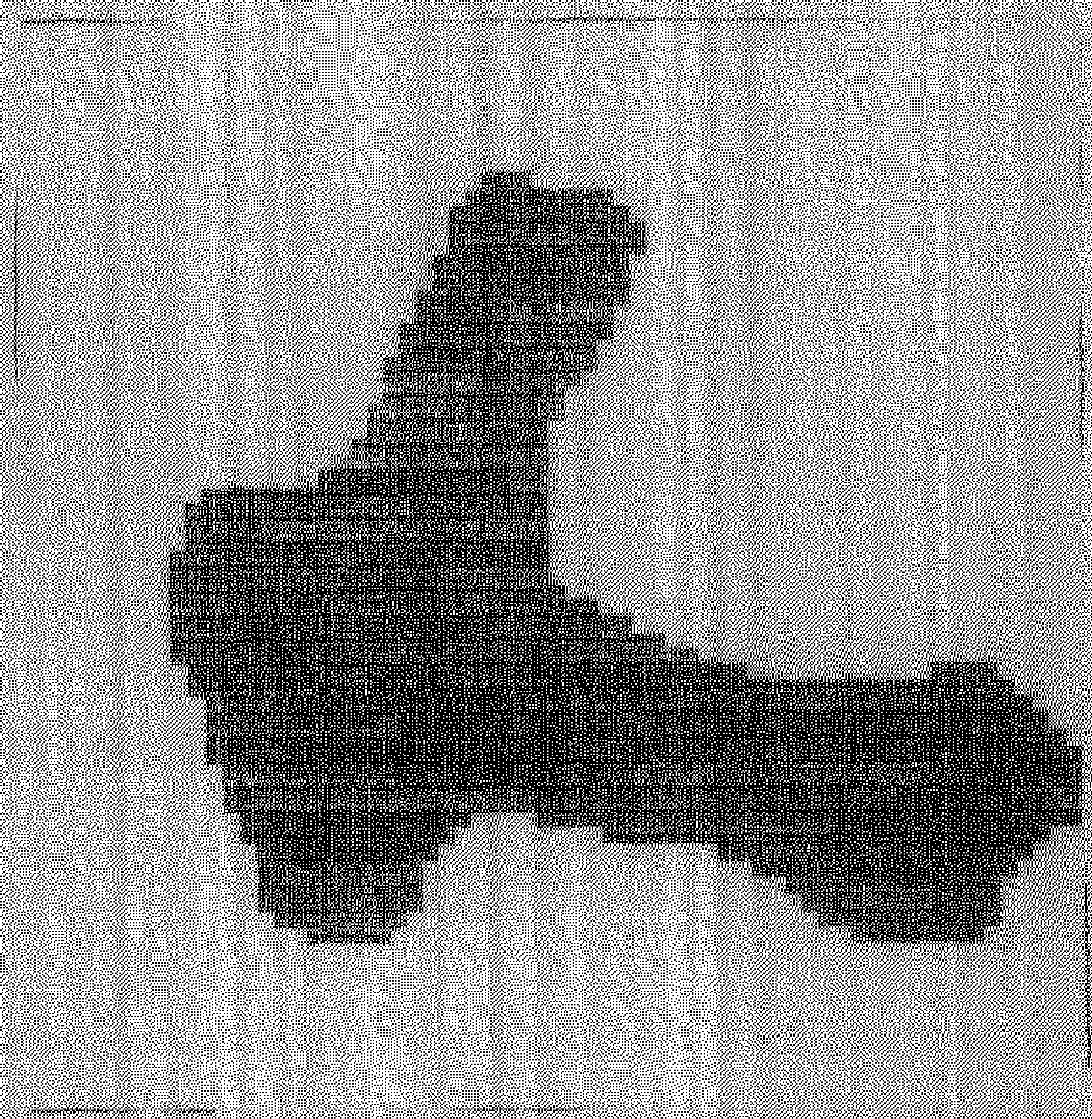

Figure 27. Extracted objects from BIPLANE image using GA.

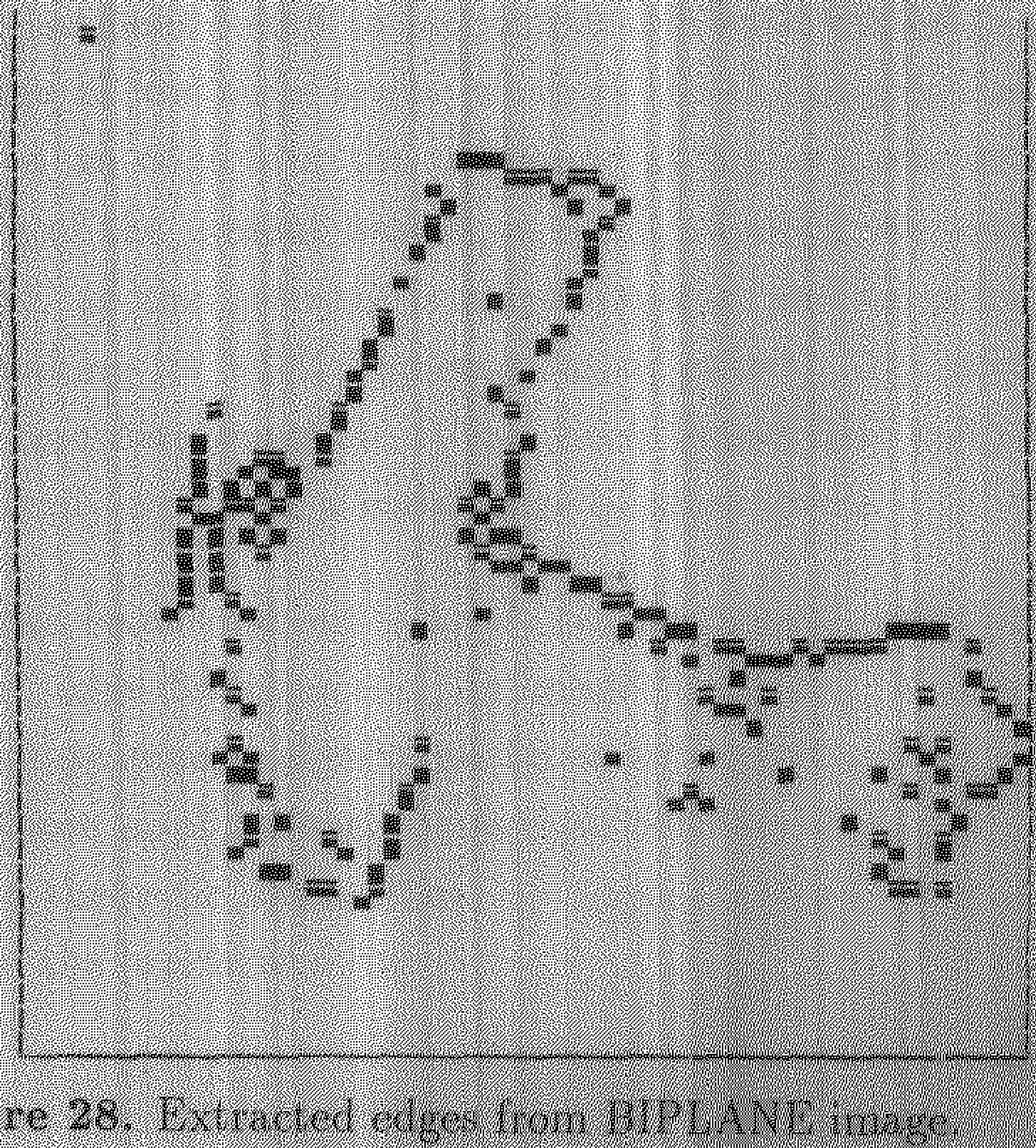

Figure 28. Extracted edges from BIPLANE image.

Chapter 3

Conclusions and Discussions

In the present study, use of cellular neural network for object extraction problem has been demonstrated. The results have been quantified in terms of the number of pixels correctly classified. A new method for selecting optimum parameters of cloning template by genetic algorithms has also been developed. Performance of cellular neural network has been found to be comparable with the existing neural network based techniques at low noise levels; whereas the existing neural network based algorithms showed an edge over the technique at higher noise level. Even though the performance of the present technique is not superior over the existing methods, it is preferred, as the network is single layered and have local interconnections (thereby making its VLSI implementation easier). By using genetic algorithms for cloning template selection, it has been found that cellular neural network performs uniformly well over a wide range of noise levels.

Further investigations can be carried out to apply cellular neural network for other areas (like enhancement, edge detection etc.) in image processing. Attempts could also be made to incorporate genetic algorithms for selecting optimum cloning template parameters for those applications. Investigations can also be carried out on the use of fuzzy set theoretic approaches, at input / output stage to handle uncertainties present in the practical data.

Bibliography

[1] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, April 1987.

[2] P. D. Wassermann, *Neural Computing : Theory and Practice.* New York: Van Nostrand Reinhold, 1990.

[3] V. Vemuri, "Artificial neural networks: an introduction," in *Artificial Neural Networks: Theoretical Concepts*, (V. Vemuri, ed.), pp. 1–12, Computer Society Press, 1988.

[4] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE trans on Circuits and Sytems*, vol. 35, pp. 1257–1272, 1988.

[5] L. O. Chua and L. Yang, "Cellular neural networks: applications," *IEEE Trans on Circuits and Sytems*, vol. 35, pp. 1273–1290, 1988.

[6] A. Rosenfeld and A. Kak, *Digital Picture Processing.* Vol. I & II, Academic Press, 2 ed., 1982.

[7] R. C. Gonzalez and P. Wintz, *Digital Image Processing.* Addison Wesley, 2 ed., 1987.

[8] L. Vandenberghe, S. Tau, and J. Vandewalle, "Cellular neural networks: dyanamic properties and adaptive learning algorithm," in *Lecture Notes in Computer Science, No. 412 : Neural Networks*, (L. B. Almeida and C. J. Wellekens, eds.), pp. 141–150, Springer Verlag, 1990.

[9] D. E. Goldberg, *Gentic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[10] S. K. Pal,D. Bhandari and M. K. Kundu, "Genetic Algorithms For Optimal Image Enhancement," Pattern Recognition Letters (accepted).

[11] D. Bhandari and N. R. Pal, "Some new information measures for fuzzy sets,". Information Sciences(in press).

[12] D. Bhandari, N. R. Pal, and D. D. Majumdar, "Fuzzy divergence, probability measure of fuzzy events and image thresholding," *Pattern Recognition letters*, pp. 857–867, December 1992.

[13] A. Ghosh, N. R. Pal, and S. K. Pal, "Image segmentation using a neural network," *Biological Cybernetics*, vol. 66, no. 2, pp. 151–158, 1991.

[14] A. Ghosh and S. K. Pal, "Neural network, self-organization and object extraction," *Pattern Recognition Letters*, vol. 13, pp. 387–397, 1992.

[15] A. Ghosh, N. R. Pal, and S. K. Pal, "Object extraction using hopfield type neural network," *International Journal of Pattern recognition and Artificial Intelligence*, vol. 6, no. 5, (in press), 1992.