Continuous Spatial Automata

B. J. MacLennan

Department of Computer Science
University of Tennessee
Knoxville, TN 37996-1301
maclennan@cs.utk.edu

CS-90-121

November 26, 1990

Abstract

A *continuous spatial automaton* is analogous to a cellular automaton, except that the cells form a continuum, as do the possible states of the cells. After an informal mathematical description of spatial automata, we describe in detail a continuous analog of Conway's "Life," and show how the automaton can be implemented using the basic operations of field computation.

1 Introduction

A *continuous spatial automaton* is the continuous analog of a cellular automaton [Codd]. Typically a cellular automaton has a finite (sometimes denumerably infinite) set of cells, often arranged in a one or two dimensional array. Each cell can be in one of a number of states. In contrast, a continuous spatial automaton has a one, two or higher dimensional continuum Ω of *loci* $x \in \Omega$ (corresponding to cells), each of which has a state σ_x drawn from a continuum (typically [0, 1]). The state is required to vary continuously with the locus, which means that $\sigma : \Omega \to [0,1]$ is continuous. We make the additional stipulation that σ have a finite L_2 norm, which makes it a field and allows us to apply the theory of field computation [MacLennan, p. 12]. We write $\Phi(\Omega)$ for the set of all fields over Ω.

In a cellular automaton there is a transition function that determines the state of a cell at the next time step based on the state of it and a finite number of neighbors at the current time step. A *discrete-time* spatial automaton is very similar: the future state of a locus is a continuous function of the states of the loci in a (closed or open) bounded neighborhood of the given locus. A *continuous-time* spatial automaton is much the same, except that the states change continuously in time, rather than at discrete time intervals.

It is usually convenient to assume that all the neighborhoods are the same "shape," i.e., they are all translations of each other. Unfortunately this conflicts with an assumption of field computation, namely that the domain Ω is closed and bounded [MacLennan, p. 10], since boundary points and interior points have neighborhoods of different shapes. Thus we must make some provision for "edge effects;" typical solutions are discussed below.

2 Example

An example will illustrate continuous spatial automata. Consider Conway's "Game of Life," a well-known two-dimensional cellular automaton with binary states [Gardner]. The new state of a cell is determined by a population density rule. Let n be the number of the eight surrounding cells in state 1. Then, if $n \leq 1$ or $n \geq 3$, the new state is 0, otherwise it is 1 if $n = 3$ and unchanged from its previous state if $n = 2$. Much of the complexity of "Life" results from the fact that the transition rule is nonmonotonic (and hence nonlinear) in the population density.

2.1 Continuous "Life"

Now we consider a discrete-time, continuous analog of "Life." For convenience we take the set of loci to be $\Omega = [0, 1]^2$, so that the state of the automaton is $\sigma \in \Phi(\Omega)$. The future state at locus $x \in \Omega$ is based on the states of the loci in a neighborhood N_x of x. For example, N_x could be a circle of radius ϵ centered at x (with appropriate provisions for edge effects):

$$N_x = \{y \in \Omega \mid \|x - y\| \leq \epsilon\}.$$

There are several ways to handle edge effects. One approach, common in cellular automata, is to treat Ω as a torus, thus extending σ to function $\hat{\sigma}$, defined on the infinite plane $\hat{\Omega} = \mathcal{R}^2$, by:

$$\hat{\sigma}(i + x, j + y) = \sigma(x, y) \text{ for } 0 \leq x < 1, \quad 0 \leq y < 1,$$

and i, j integers. Continuity of $\hat{\sigma}$ requires that the edges of σ match, but this is easily accomplished. Another way to extend σ to the infinite plane is to assume it is constant outside of $[0, 1]^2$.

By analogy with "Life" we will make the new state at x a nonmonotonic function ζ of the "population density" ψ_x in the neighborhood N_x. The function $\zeta : [0, 1] \to [0, 1]$ is defined by a parameter m, $0 < m < 1$, and has the following properties:

$$\zeta(0) \quad = \quad 0$$

$$\begin{aligned} \zeta(1) &= 0 \\ \zeta(m) &= 1 \\ \zeta'(p) &\geq 0 \text{ for } 0 \leq p \leq m \\ \zeta'(p) &\leq 0 \text{ for } m \leq p \leq 1. \end{aligned}$$

This means that $\zeta(p)$ increases continuously from 0 at $p = 0$ to 1 at $p = m$, and then decreases continuously to 0 at $p = 1$.[1]

Since the population density ψ_x around x is just the average of σ over N_x we can write the new state $\sigma' \in \Phi(\Omega)$ as follows:

$$\sigma'_x = \zeta\left(\frac{\int_{N_x} \sigma_y \mathrm{d}y}{\int_{N_x} \mathrm{d}y}\right) = \zeta\left(\frac{1}{a_x}\int_{N_x} \sigma_y \mathrm{d}y\right),$$

where $a_x = \int_{N_x} \mathrm{d}y$ is the area of N_x.

Next we define a "neighborhood aggregation" operator $A : \Phi(\Omega) \rightarrow \Phi(\Omega)$ to compute the field ψ:

$$A(\sigma) = \psi \text{ where } \psi_x = \frac{1}{a_x}\int_{N_x} \sigma_y \mathrm{d}y.$$

Then $\sigma'_x = \zeta(\psi_x)$, so we may write

$$\sigma' = \bar{\zeta}(\psi)$$

where $\bar{\zeta} : \Phi(\Omega) \rightarrow \Phi(\Omega)$ is the "local transformation" $\bar{\zeta}(\psi)$ that applies ζ at each point of ψ [MacLennan, p. 48]. Hence the state transition operator $T : \Phi(\Omega) \rightarrow \Phi(\Omega)$ is defined

$$T(\sigma) = \sigma' = \bar{\zeta}[A(\sigma)],$$

or $T = \bar{\zeta} \circ A$.

2.2 Computing $\bar{\zeta}$

There are several candidates for ζ; perhaps the simplest is a parabola (possibly rotated) with maximum $(m, 1)$ and passing through the points $(0, 0)$ and $(1, 0)$. Such a parabolic nonlinearity can result in very complex — indeed, chaotic — dynamics [Hofbauer & Sigmund, pp. 35–40]. In the simplest case, where $m = 1/2$, we have

$$\zeta(p) = 4p(1 - p).$$

Hence we can express $\bar{\zeta}$ directly by the field computation

$$\bar{\zeta}(\psi) = 4\psi \times (1 - \psi),$$

where '$\times$' denotes "local product" [MacLennan, p. 40]: $(\phi \times \psi)_x = \phi_x \psi_x$.

[1] The closest analogy to the standard version of "Life" has $m = 3/8$.

2.3 Computing A

Our next task is to find a way to compute the aggregation operator A. Let $\nu \in \Phi(\Omega)$ be the field

$$\nu_x = \begin{cases} a^{-1} & \text{if } x \in N_0 \\ 0 & \text{otherwise} \end{cases}$$

where N_0 is the neighborhood of the origin and $a = a_0$ is the area of N_0. For circular neighborhoods we have:

$$\nu_{(x_1,x_2)} = \begin{cases} 1/\pi\epsilon^2 & \text{if } x_1^2 + x_2^2 \leq \epsilon^2 \\ 0 & \text{otherwise} \end{cases}$$

Although this function is discontinuous, it can be approximated arbitrarily closely by continuous fields.[2] If all neighborhoods are translations of N_0 we have

$$y \in N_x \quad \text{iff} \quad y - x \in N_0 \quad \text{iff} \quad \nu_{y-x} = a^{-1}.$$

Since for all x, $a_x = a$, we have

$$\begin{aligned} [A(\sigma)]_x &= \psi_x \\ &= a_x^{-1} \int_{N_x} \sigma_y \mathrm{d}y \\ &= \int_{N_x} a^{-1} \sigma_y \mathrm{d}y \\ &= \int_{\hat{\Omega}} \nu_{y-x} \hat{\sigma}_y \mathrm{d}y \\ &= (\nu \star \hat{\sigma})_x \end{aligned}$$

where $\nu \star \hat{\sigma}$ is the correlation of ν and $\hat{\sigma}$, the result of which we assume to be restricted to Ω. Alternately we could do a convolution with $\bar{\nu}$ where $\bar{\nu}_x = \nu_{-x}$:

$$A(\sigma) = \nu \star \hat{\sigma} = \bar{\nu} \otimes \hat{\sigma}.$$

Hence the new state is defined by

$$\sigma' = T(\sigma) = \bar{\zeta}(\nu \star \hat{\sigma}).$$

If we use the $m = 1/2$ parabola, then the new state is computed in two steps:

$$\begin{aligned} \psi &= \nu \star \hat{\sigma} \\ \sigma' &= 4\psi \times (1 - \psi) \end{aligned}$$

Use of other values of m is a straight-forward extension, as is the use of other ζ functions.

---

[2] Indeed, this definition suggests a generalization in which ν is any probability density function.

2.4 Simulation

Tomislav Goles has implemented a version of the continuous Life system described above. As expected it exhibits interestingand complex behavior, including the formation of strings and networks of high-density regions (Figs. 1–4). The scale of these structures seems to be closely related to the neighborhood radius. Details of the implementation can be found in Goles' MS thesis [Goles].

References

[Codd] Codd, E. F. *Cellular Automata.* Academic Press, NY, 1968.

[Gardner] Gardner, Martin. "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'." *Scientific American* **223**, 4 (October 1970), pp. 120–123.

[Goles] Goles, Tomislav, "General Purpose Field Computer Simulator." MS thesis, Computer Science Department, University of Tennessee, Knoxville, December 1990.

[Hofbauer & Sigmund] Hofbauer, Josef, and Sigmund, Karl. *The Theory of Evolution and Dynamical Systems*, London Mathematical Society Student Texts **7**. Cambridge University Press, Cambridge UK, 1984.

[MacLennan] MacLennan, B. J. "Field Computation: A Theoretical Framework for Massively Parallel Analog Computation, Parts I–IV." University of Tennessee, Knoxville, Computer Science Department Technical Report CS-90-100, February 1990.

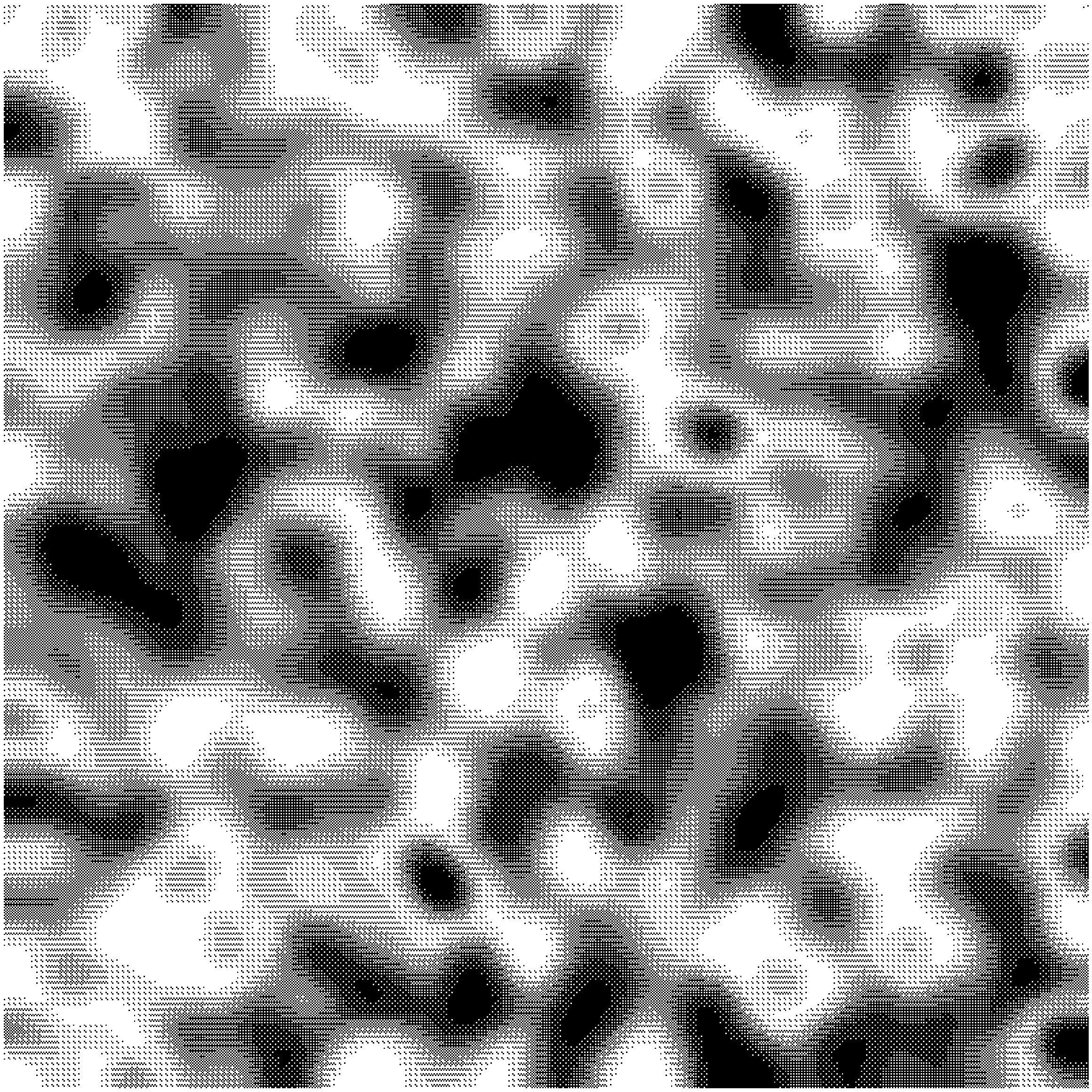

Figure 1: Typical State, $\Omega = [0,1]^2, \epsilon = 0.02$

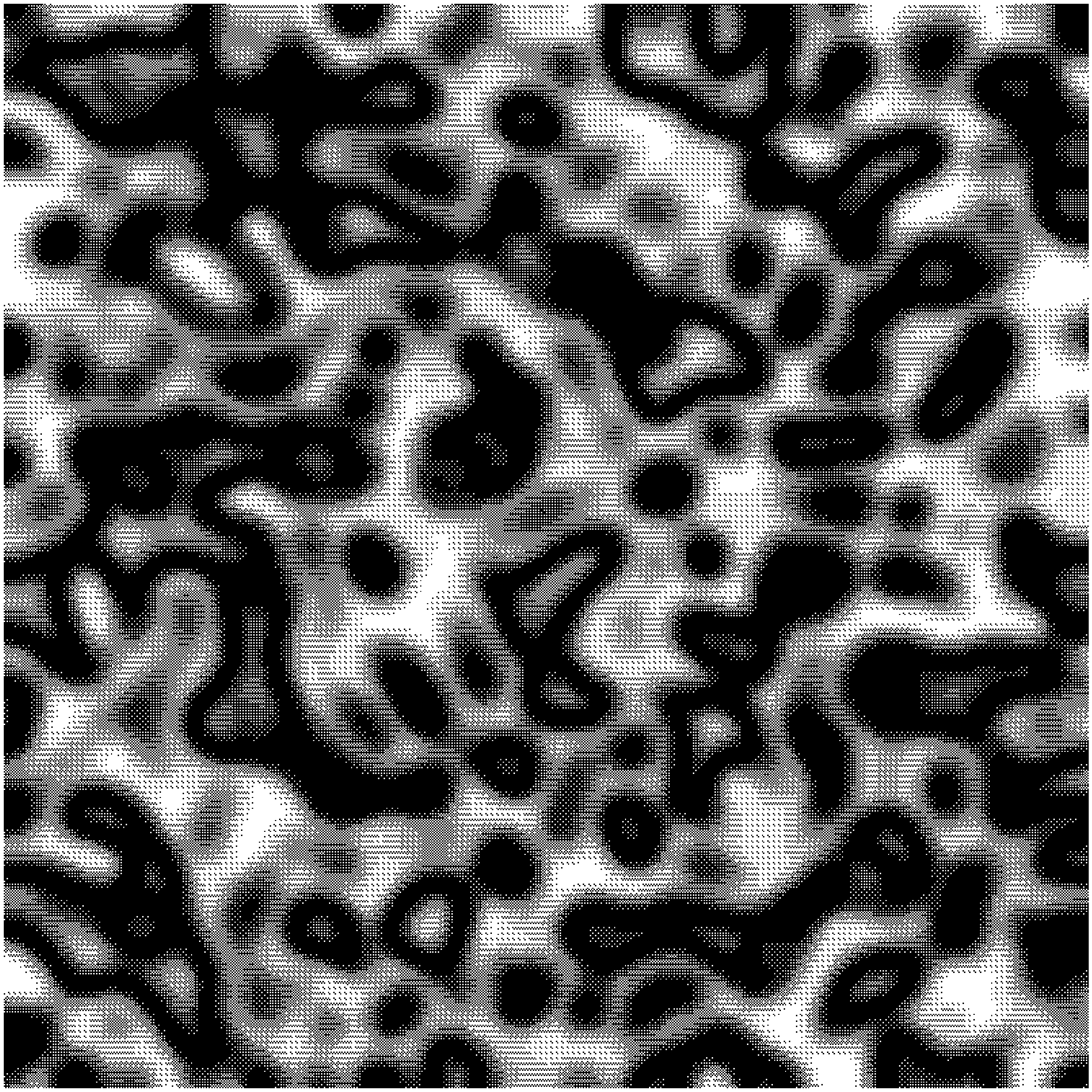

Figure 2: Typical State, $\Omega = [0,1]^2, \epsilon = 0.02$

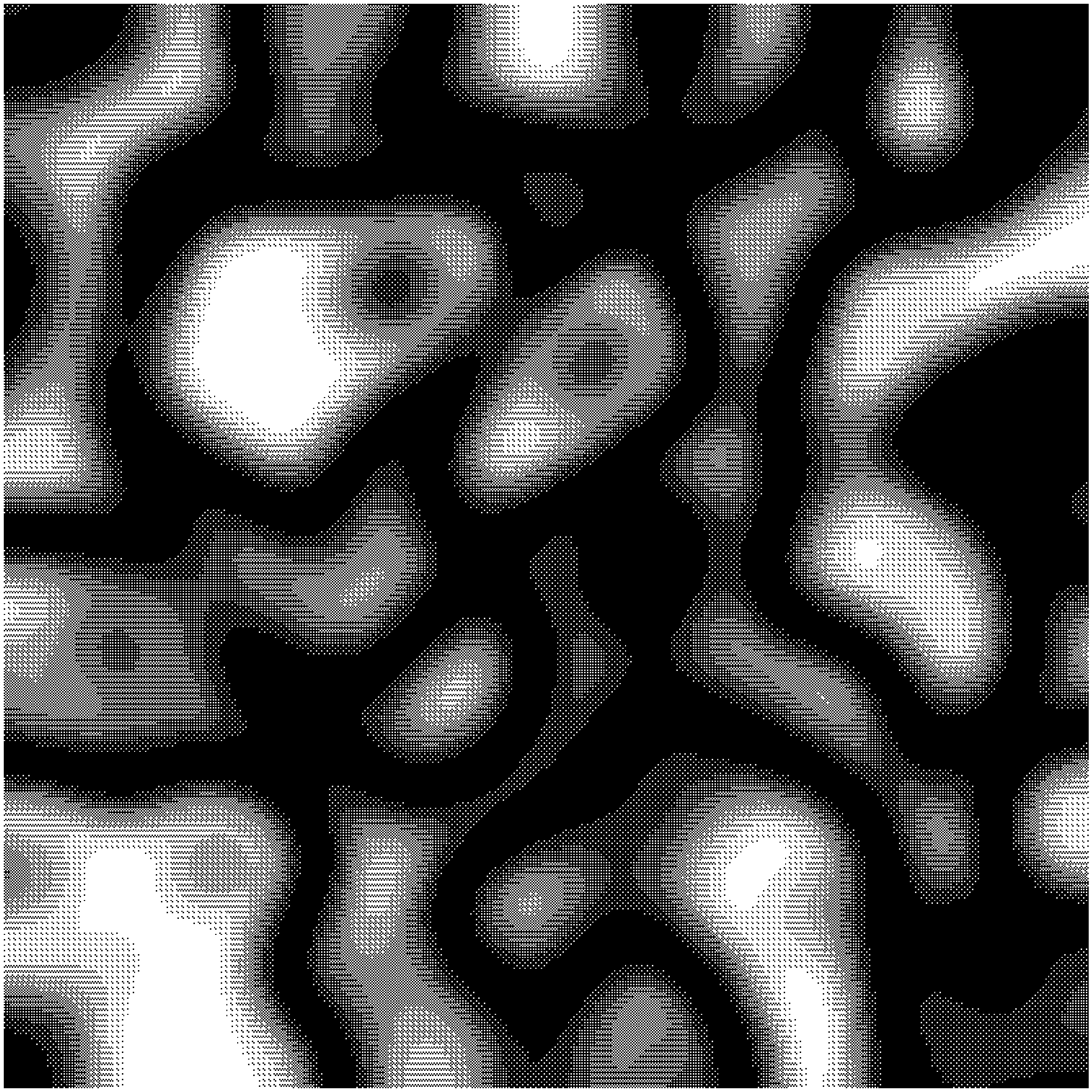

Figure 3: Typical State, $\Omega = [0,1]^2, \epsilon = 0.04$

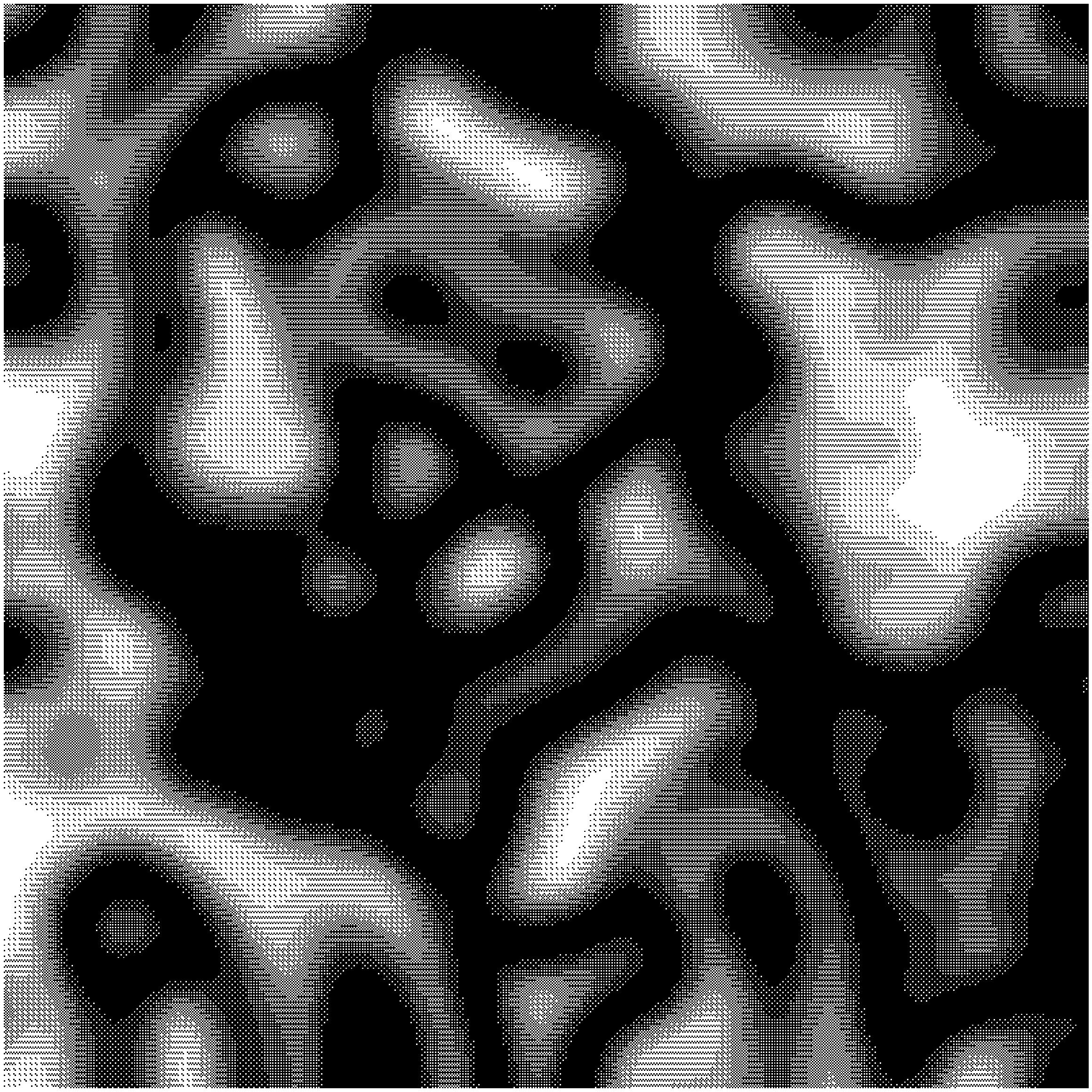

Figure 4: Typical State, $\Omega = [0, 1]^2, \epsilon = 0.04$