Abstract—To solve the fuzzy edge detection problems in image processing, a novel fuzzy clustering method based on chaos small-world algorithm (CSWFCM) is presented. The traditional fuzzy clustering method (FCM) is good at local searching capability, but it is sensitive to the initial value and easy to trap into local minimum value. The small-world algorithm (SWA), inspired by the mechanism of small-world phenomenon, is a novel global searching algorithm, which enables to enhance the diversity of the population and avoid trapping into local minimum value. However, the further capability of solving complicated problems is limited for its low efficiency of local short-range searching operator. In this paper, the chaos disturbance is utilized to improve the searching efficiency of SWA after local short-range search, and the chaos small-world algorithm (CSWA) is used to optimize the FCM in image edge detection. The simulation results show that the proposed algorithm can correctly detect the fuzzy and exiguous edges with higher convergence speed.

Keywords—fuzzy clustering algorithm, chaos optimization, small-world algorithm, edge detection

I. INTRODUCTION

Edge detection is extensively used to separate the object from the background in image processing. A lot of research has been done in the field of image segmentation using edge detection. Some of the earliest operators detecting edges in an image were proposed by Sobel, Prewitt, Roberts, etc. [1]. These operators used local gradient methods to detect edges along a specified direction. The common premise conditions of these operators are that the edges of an image must be clear. However, the image in reality is fuzzy and the edges aren’t clear.

In 1966, Bellman and Zadeh firstly put forward fuzzy clustering analysis [2], and the idea of fuzzy clustering quickly becomes an efficient method to solve fuzzy edge detection in image processing. Among numerical fuzzy clustering methods, fuzzy c-means clustering method provides foundation for other fuzzy clustering methods from the theories and applications, and is used widely. In essence, fuzzy c-means clustering method is a kind of local searching algorithm. It is so sensitive to the initial value that it easily traps into local minimum value. To solve the disadvantage, some algorithms such as genetic algorithm (GA) [3] and immune evolutionary algorithm (IEA) [4], have been used to optimize the fuzzy clustering, which result in good effects to some extent. However, some disadvantages of GA and IEA, such as slow convergence speed and instability, affect the accuracy of clustering.

In this paper, a novel fuzzy clustering method based on the optimization of chaos small-world algorithm is proposed and used in image edge detection. In the proposed algorithm, first, the initial population is generated by logistic mapping, and the chaos disturbance is used to improve the searching efficiency of SWA after local short-range search; secondly, the FCM is optimized by the chaos small-world; finally, the CSWFCM is used in image edge detection. The paper is organized as follows: The fuzzy clustering method is presented in Section II. The small-world phenomenon and small-world optimization algorithm are described in Section III. The chaos optimization algorithm is presented in Section IV. Section V discusses the fuzzy clustering method based on chaos small-world algorithm. The experiments and corresponding analyses about CSWFCM are described in Section VI. Finally, Section VII states some conclusions.

II. FUZZY CLUSTERING METHOD

Supposing the finite set \( X = \{x_1, x_2, \cdots, x_n\} \) is belonged to the \( p \)-dimensional Euclidean space \( \mathbb{R}^p \), namely \( \forall \ k = 1, 2, \cdots, n, x_k \in \mathbb{R}^p \). FCM partitions \( X \) into \( c \) fuzzy groups, and finds a clustering center in each group, such that the objective function based on distance is minimal. The objective function for FCM is defined as follows:

\[
J(U, V) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^m (d_{ik})^2
\]

(1)

Where, \( U \) is the fuzzy matrix of \( X \), \( V \) is the clustering center set of \( X \), \( m \in (1, +\infty) \) is the weight coefficient, \( \forall \ 1 \leq i \leq c, \ 1 \leq k \leq n, \ d_{ik} = \|x_k - v_i\| \) is the Euclidean distance between \( x_k \) and \( v_i \), \( \mu_{ik} \in X \) is the degree of membership of data \( x_k \) relevant to the \( i \)-th clustering center \( v_i \), and \( \mu_{ik} \) satisfies the following restriction:

\[
\forall \ 1 \leq k \leq n, \sum_{i=1}^{c} \mu_{ik} = 1
\]

(2)

The concrete operating sequences of FCM are as follows:

**Step1** Set parameters: \( n, m, c, t = 0 \).

**Step2** Set the initial population \( U \) by Eq. (1) with \( m = 2 \), and randomly initialize each line of matrix \( V \), satisfies the restriction of Eq. (2) and \( d_{ik} = 0 \). Set iteration criterion, set iteration times as \( t \).

**Step3** Calculate the Euclidean distance of each data \( x_k \) to each clustering center \( v_i \), namely \( d_{ik} = \|x_k - v_i\| \), and calculate the membership degree of data \( x_k \) to each clustering center:

\[
(\mu_{ik})^m = (d_{ik})^{-2}
\]

**Step4** Update the value of clustering center and membership degree of data:

\[
\forall \ 1 \leq i \leq c, \ \sum_{k=1}^{n} \mu_{ik} = 1
\]

**Step5** If the iteration times are less than the criterion, return to the third step. Otherwise, stop the operation and output the results.
Step 2 Initialize clustering center randomly: \( V(t) = \{v_1, v_2, \ldots, v_r \} \).

Step 3 Calculate \( U(t) \).
\[
\forall 1 \leq i, j \leq c , 1 \leq k \leq n ,
\]
If \( d_{ki} \neq 0 \), \( \mu_{ki} = 1 / \sum_{j=1}^{c} \left[ d_{ki} / d_{kj} \right]^{2/(m-1)} \) \( \quad (3) \)
Otherwise if \( i = j \), \( \mu_{ki} = 1 \);
if \( i \neq j \), \( \mu_{ki} = 0 \).

Step 4 Calculate \( V(t+1) \).
\[
\forall 1 \leq i \leq c , v_i = \sum_{k=1}^{n} \mu_{ki} x_k / \sum_{k=1}^{n} \mu_{ki} \] \( \quad (4) \)

Step 5 Choose an appropriate matrix norm to compare \( V(t+1) \) and \( V(t) \). If \( \| V(t+1) - V(t) \| \leq \varepsilon \), end; otherwise \( t = t + 1 \) and go to Step 3, where \( \varepsilon \) is a positive and small enough real number.

III. SMALL-WORLD OPTIMIZATION ALGORITHM

In the 1960s, social psychologist Stanley Milgram performed a small-world experiment of tracing out short paths through the social networks of the United States [5]. He asked a few hundred people in Omaha to forward a letter to a “target” stranger in Boston through personal contacts, and ended up with 60 completed chains of letters that averaged six places. He asked a few hundred people in Omaha to forward a letter to a “target” stranger in Boston through personal contacts, and ended up with 60 completed chains of letters that averaged six places. He asked a few hundred people in Omaha to forward a letter to a “target” stranger in Boston through personal contacts, and ended up with 60 completed chains of letters that averaged six places. He asked a few hundred people in Omaha to forward a letter to a “target” stranger in Boston through personal contacts, and ended up with 60 completed chains of letters that averaged six places.

Inspired by the mechanism of small-world phenomenon, Du firstly proposed small-world algorithm (SWA) for function optimization [13]. He took the optimization as a process that information transmits from candidate node (i.e. candidate solution) to optimal node (i.e. optimal solution) in network (i.e. searching space). The SWA includes local short-range searching operator and random long-range searching operator.

The two-dimensional space searching of SWA is schematically shown in Fig. 1. From the figure, we can see that a candidate node moves to optimal node through local short-range search and random long-range search, and the transition information is the solution of optimization model during the search.

Figure 1. Two-dimensional space searching principle of SWA

Without loss of generality, the following global optimization model is considered:
\[
J = \min_{x \in \mathbb{R}^q} f(x) , \quad x = \{x_1, x_2, \ldots, x_q \} \quad (5)
\]

Where, \( f(x) \) is the objective function, \( q \) is the dimension of vector \( x \). \( x_i \in [d_i, u_i] \).

According to [13], let \( S \) be the \( N \)-dimensional transmission node population. \( \forall \) node \( s \in S \), \( s = a_1 a_2 \ldots a_m \) is the binary coding of \( q \)-dimensional variable \( x \). \( \forall \) \( a_i \), \( 1 \leq i \leq q \), \( a_i = a_i^1 a_i^2 \ldots a_i^m \), \( a_i^j \in \{0,1\} \), \( 1 \leq j \leq l_i \), \( l_i \) depends on the binary coding precision \( \sigma \). \( a_i \) is called the coding of variable \( x_i \), and described as \( a_i = e(x_i) \), where \( e(\bullet) \) is the coding manner. \( x_i \) is called the decoding of node \( a_i \), and described as \( x_i = e^{-1}(a_i) \), where \( e^{-1}(\bullet) \) is the decoding manner.

Definition 1: \( \forall \) \( a_i \), \( 1 \leq i \leq q \), the decoding manner is defined as follows:
\[
x_i = e^{-1}(a_i) = d_i + (u_i - d_i) \frac{1}{2^l - 1} \left( \sum_{j=1}^{l_i} a_i^j 2^{j-1} \right) \quad (6)
\]

Definition 2: \( \forall s_i \in S \), the \( 1 \) neighborhood set of the node \( s_i \) can be defined as follows:
\[
\zeta^1(s_i) = \{ s_j \mid 0 < \| s_i - s_j \| \leq 1 , s_j \in S \} \quad (7)
\]
Where, $||\cdot||$ is the Hamming distance.

**Definition 3:** $\forall s_i \in S$, the non-1 neighborhood set of $s_i$ can be defined as follows:

$$\zeta^1(s_i) = \{s_j \mid ||s_i - s_j|| > 1, s_j \in S\}$$  (8)

The SWA inspired by the mechanism of small-world phenomenon includes local short-range searching operator $\Psi$ and random long-range searching operator $\Gamma$.

Supposing $R_i(k) \subseteq \zeta^1(s_i(k))$ is the local searching of node $s_i(k)$, the main action of $\Psi$ is transmitting the information from node $s_i(k)$ to the node $s_i(k+1)$, which is the nearest node to the optimization model in $R_i(k)$. The process can be described as follows:

$$s_i(k+1) \leftarrow \Psi(s_i(k))$$

$$= \{s_i^* \in R_i(k) \mid f(e^{-1}(s_i^*))\}$$

$$= \min_{s_i(k) \in R_i(k)} (f(e^{-1}(s_i(k))))$$  (9)

$\forall s_i(k+1) \in \zeta^1(s_i)$, the main action of $\Gamma$ is transmitting the information from $s_i(k)$ to $s_i(k+1)$ at given preset long-range searching probability $p_c$. The process can be described as follows:

$$s_i(k+1) \leftarrow \Gamma(s_i(k))$$

$$= \{s_i^*(k) \mid p < p_c : s_i^*(k) \in \zeta^1(s_i(k)), p = \text{rand}()\}$$  (10)

The concrete operating processes of $\Psi$ and $\Gamma$ are shown in [13].

### IV. Chaos Optimization Algorithm

In SWA, the main operation of $\Psi$ is randomly searching a node in $\zeta^1(s_i(k))$. The operating manner is so easy that the local searching efficiency of SWA is decreased. To improve the searching efficiency, the chaos optimization is used in SWA. First, making use of the characteristics of ergodicity and randomness of chaotic variables, the initial population is generated by logistic mapping; secondly, the local search of individual is performed by chaos disturbance after local short-range search.

Considering the Logistic Equation is more convenient and the calculated amount is less than other chaos iteration equations, we use the following (11) to generate initial population $S(0)$.

$$z(k+1) = \eta z(k)(1 - z(k))$$  (11)

Where, $\eta$ is the chaos attractor, the chaos space is $[0, 1]$ when $\eta = 4$. $z(k)$ is the chaos variable at $k$th iteration, $z(k) \in (0,1)$ and $z(k) \notin \{0.25,0.5,0.75\}$. Let $\nu$ be the total iteration times of mapping.

The chaos disturbance is defined as follows:

$$\beta(k) = (1-\alpha)\beta^* + \alpha \beta(k)$$  (12)

Where, $\beta^* = \Theta(x^*)$ is the optimal chaos variable, $x^*$ is the current optimal solution of optimization model, $\Theta(\bullet)$ is the mapping from solution space to chaos space, let $\Theta^{-1}(\bullet)$ be the inverse mapping, namely $x(k) = \Theta^{-1}(\beta(k))$, $\beta(k)$ is the chaos variable at $k$th iteration, $\beta^*(k)$ is the chaos variable after chaos disturbance, $\alpha$ is the adjustment coefficient and defined as follows:

$$\alpha = 1 - \left|\frac{k-1}{\nu}\right|^\nu$$  (13)

Where, $p$ is an integer and $p = 2$ in this paper, $k$ is the iteration times. Let $\gamma$ be the total iteration times of chaos disturbance.

### V. FCM Based on Chaos Small-World

According to the objective function $J(U,V)_m$, we can know that the final target of fuzzy clustering is to obtain the clustering center $V$ and fuzzy matrix $U$ of finite set $X$. Since $V$ and $U$ are interrelated, we can take either of them as the optimization variable. Considering the computation cost, we choose to code on $V$ by binary in this paper. The objective function $J(U,V)_m$ is taken as the fitness function of optimization problem.

Based on the above fuzzy cluster method, small-world optimization algorithm and chaos optimization algorithm, we can depict a novel FCM based on chaos small-world as follows:

**Step1** Initialization: binary coding precision $\sigma$, clustering center number $c$, weight coefficient $m$, population size $N$, neighborhood size $l$, long-range searching probability $p_c$, total iteration times of mapping $\nu$, total iteration times of chaos disturbance $\lambda$, maximal generation $k_{\text{max}}$, $k \leftarrow 0$.

**Step2** Extract datum.

**Step3** Generate initial population of clustering centers $S(0)$ according to (11).
Step 4 Execute the optimization of population \( S(k) = \{ s_1(k), s_2(k), \ldots, s_N(k) \} \) based on small-world optimization algorithm.

Step 4.1 \( S'(k) \leftarrow S(k) \);

Step 4.2 Finish random long-range search according to given \( p_c : s'_i(k+1) \leftarrow \Gamma(s'_i(k)) \), \( s'_i(k) \in S'(k) \), \( 1 \leq i \leq N \);

Step 4.3 Finish local short-range search:
\[
\xi'_i(k+1) \leftarrow \psi(s'_i(k+1)), 1 \leq i \leq N;
\]

Step 4.4 If \( J(e^{-1}(s'_i(k+1))) < J(e^{-1}(s_i(k))) \) then \( s_i(k) \leftarrow s'_i(k+1), 1 \leq i \leq N \).

Step 5 Execute the optimization of \( S(k) \) based on chaos disturbance.

Step 5.1 Choose \( \text{fix}(\mu \cdot N) \) individuals whose fitness values are smaller in \( S(k) \), and form the population \( S'(k) = \{ s_1(k), s_2(k), \ldots, s_l(k) \} \), \( l = \text{fix}(\mu \cdot N) \), where, \( \text{fix} \) is the round function;

Step 5.2 Transforming from coding space to chaos space:
\[
\forall s_i(k) \in S'(k), 1 \leq i \leq \text{fix}(\mu \cdot N), x(k) \leftarrow x(s_i(k));
\]
\[
\beta(k) \leftarrow \Theta(x(k));
\]

Step 5.3 Execute chaos disturbance to \( \beta(k) \) according to (12) (13).

Step 6 Judge whether the terminating condition is satisfied. If not, go on the following process, otherwise output the optimal cluster centers and end.

Step 7 \( k \leftarrow k + 1 \), and go to Step 4.

VI. SIMULATION RESULTS AND ANALYSIS

To verify the validity of our proposed algorithm in image edge detection, two images of 256×256 pixels, namely Lena and Cameraman, are tested with MATLAB 7.0 on an Intel Pentium IV 2.99GHz computer with 512MB RAM. We compare the detection results among FCM, GAFCM and CSWFCM. In CSWFCM, \( m = 2, N = 30, l = 1, p_c = 0.8, \sigma = 10^{-5}, \nu = 300, \lambda = 30 \) and \( k_{\text{max}} = 50 \). In GAFCM, the population size is 30, crossover probability is 0.8, mutation probability is 0.1, and maximal generation is 50. The parameters of FCM are the same as the corresponding parameters in CSWFCM. During the image edge detection, four values are chosen for the cluster number \( c \), namely 2, 3, 4 and 5. Considering the randomness, each algorithm is tested for twenty times.

Table I is the performance comparison of edge detection among three algorithms. From the table we can see that with the increase of clustering center number \( c \), the average and best fitness \( J(U,V)_m \) both decreases, which indicates that the fuzzy clustering effects are better and better. Aiming at any certain clustering center number \( c \), the fitness of CSWFCM is better than FCM and GAFCM. The differences of average and best fitness are not obvious when \( c = 2, 3 \) and 5, however, when \( c = 4 \), FCM always traps into local minimum value, GAFCM and CSWFCM can avoid the deceptive problem and get better results. All in all, through table I, we can see that CSWFCM can converge to a global optimal value and get better edge detection effect than other two methods to some extent.

<table>
<thead>
<tr>
<th>Image</th>
<th>( c )</th>
<th>( J(U,V)_m )</th>
<th>FCM</th>
<th>GAFCM</th>
<th>CSWFCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>2</td>
<td>Average 1418.04</td>
<td>1407.70</td>
<td>1403.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 1417.97</td>
<td>1407.26</td>
<td>1403.26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Average 893.11</td>
<td>885.02</td>
<td>880.36</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 892.18</td>
<td>881.27</td>
<td>880.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Average 883.11</td>
<td>559.01</td>
<td>530.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 856.73</td>
<td>543.86</td>
<td>528.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Average 412.83</td>
<td>382.37</td>
<td>372.43</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 384.90</td>
<td>370.59</td>
<td>368.61</td>
<td></td>
</tr>
<tr>
<td>Cameraman</td>
<td>2</td>
<td>Average 1006.56</td>
<td>996.03</td>
<td>995.21</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 1002.75</td>
<td>995.32</td>
<td>995.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Average 531.30</td>
<td>514.39</td>
<td>506.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 525.71</td>
<td>507.81</td>
<td>500.98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Average 537.66</td>
<td>404.76</td>
<td>394.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 497.67</td>
<td>398.712</td>
<td>393.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Average 319.14</td>
<td>304.16</td>
<td>299.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Best 304.56</td>
<td>300.31</td>
<td>294.80</td>
<td></td>
</tr>
</tbody>
</table>

In order to see the edge detection results in reality, we take the Lena image as an example and see the differences of edge detection among three algorithms when \( c = 3 \) and \( c = 4 \). Fig. 2 is the original Lena image. Fig. 3 and Fig. 4 are the comparisons of edge detection about Lena image among three algorithms when \( c = 3 \) and \( c = 4 \) respectively.
Figure 3. Comparisons of edge detection about lena image among three algorithms when $C = 3$

Figure 4. Comparisons of edge detection about lena image among three algorithms when $C = 4$
As the best fitness $J(U, V)_m$ about lena image of FCM, GAFCM and CSWFCM is close when $c=3$, we can see that edge detections based on above three algorithms all get good effects and the differences of edge detections aren’t obvious from Fig. 3. When $c=4$, it is obvious that the edge detections of GAFCM and CSWFCM are better than FCM for theirs global optimization capacity, and the detection of CSWFCM has more detail and clearer contour, such as the continuities of face and hat brim, than FCM and GAFCM.

![Figure 5. Convergence curves of best fitness about lena image of three algorithms when $c=3$](image)

![Figure 6. Convergence curves of best fitness about lena image of three algorithms when $c=4$](image)

Fig. 5 and Fig. 6 are the convergence curves of best fitness about lena image of FCM, GAFCM and CSWFCM when $c=3$ and $c=4$ respectively. When $c=3$, although the best fitness of three algorithms is close, the convergence speed of CSWFCM is quicker than FCM and GAFCM, and the convergence of FCM has fluctuation as shown in Fig. 5. When $c=4$, from Fig. 6, we can see that FCM traps into local minimum value, GAFCM and CSWFCM avoid local minimum and get good results to some extent, and the convergence speed of CSWFCM is quicker than GAFCM.

VII. CONCLUSIONS

In this paper, a novel fuzzy clustering methods based on chaos small-world algorithm is presented to solve the image edge detection, and the edge detection results of two images are compared with FCM and GAFCM. According to the simulation experiments, we can draw the following conclusions: (1) Inspired from the mechanism of small-world phenomenon, the small-world algorithm is an efficient optimization algorithm, and chaos optimization further improves the searching efficiency of SWA; (2) FCM based on the optimization of CSWA avoids the sensitivity to the initial value of clustering center and trapping into local minimum value; (3) The image processing results of CSEFCM are better at detail and clearness of contour than FCM and GAFCM; (4) CSWFCM has the higher convergence speed and stability than FCM and GAFCM.

ACKNOWLEDGMENT

The support of the project of mechanical manufacturing equipment key laboratory of Shan’xi province (Grant No. 03JF06) is gratefully acknowledged.

REFERENCES