

Image Segmentation Using Cellular Automata: A Technical Survey

Priyanka Shotrya¹, Sanjeev Bhardwaj²

¹M.Tech Scholar, I.F.T.M University, Moradabad, India

²Deptt. Of Computer Science & Engg., I.F.T.M University, Moradabad, India

priyankashotrya@gmail.com, sanjeevmbd@gmail.com

Abstract—Image segmentation is an integral part of image processing applications like medical images analysis and photo editing. In this paper, we have discussed the most commonly used Image Segmentation techniques using Cellular Automata. The segmentation technique to be chosen depends on the properties of the image being segmented. We have also discussed different scenarios in which different segmentation techniques can be best employed.

Keyword—cellular automata, segmentation, cellular learning automata,

1. Introduction

In its simplest meaning Segmentation means division of an image into different objects or regions. In segmentation, the area of our concern is labeled as foreground and rest of the image is labeled background. Manual segmentation of is a time-consuming and difficult job. Mostly, the task is done by marking the object of our interest, but this approach does not guarantees accurate results. We need a segmentation method that segments an image automatically without much interaction of the user and provide accurate results. This automated method makes use of Cellular Automata. The underlying concept behind using Cellular Automata for automatic segmentation using is that an image is composed of pixels that can be treated as cells . The segmentation of an image is based on the properties of the image such as intensity and other intensity derived properties. The segmentation of the image also depends on the value of the intensities of the neighborhood of the pixel, i.e. the classification of a pixel as foreground or background pixel also depends on the neighboring pixels.

2. Cellular Automata

Cellular automata (CA) were originally introduced by Ulam [1] and Von Neumann [2] with the purpose of obtaining models of biological self-reproduction. Later on, Amoroso and Fredkin and Cooper proposed a simple replicator based on parity or modulo-two rules. Finally actual work in this field started only after Stephen Wolfram developed the CA theory. These days cellular automata is used in a large number of fields as it can be used as a discrete model for many processes. Cellular Automata is currently used in branches derived from physics, chemistry, biology, economics, and information systems . Cellular Automata is

also used in Digital Image Processing which include image denoising, edge detection, restoration, enhancement, segmentation, compression, feature extraction and pattern recognition. Cellular automaton is a discrete dynamical system. Space, time, and the states of the system are discrete. Each point in a regular spatial lattice, called a cell, can have any one of a finite number of states. A local transition function is defined to update the state of a cell. The local transition function takes the present state of the cell and state of the neighborhood cells as input and outputs the next state of the cell. Another specialty of Cellular Automata is that the updation phase is applied to the whole lattice of cells simultaneously. Thus the state of the entire lattice advances in discrete time steps.

A (bi-directional, deterministic) cellular automaton is a triplet $A = (S; N; \delta)$, where S is an non-empty *state set*, N is the neighborhood system, and $\delta: S^N \rightarrow S$ is the local transition function (rule). This function defines the rule of calculating the cell's state at $t + 1$ time step, given the states of the neighborhood cells at time step t [3]. Commonly used neighborhood systems N are the von Neumann and Moore neighborhoods:

Von Neumann Neighborhood

A diamond-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0, y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid. Mathematically, the Von Neumann neighborhood of range r is defined as:

$$N_{(x_0, y_0)} = \{(x, y): |x - x_0| \leq r, |y - y_0| \leq r\}$$

Von Neumann neighborhoods for $r=1$ is shown in fig 1.

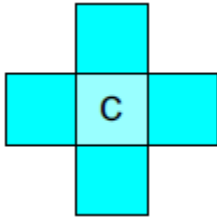


Fig 1 Von Neumann neighborhood

A square-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0, y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid. Mathematically, the Moore neighborhood of range r is defined as:

$$N_{(x_0, y_0)} = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\} \quad 2$$

Moore neighborhoods for $r=1$ is shown in fig. 2.

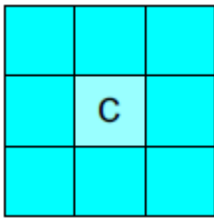


Fig 2 Moore Neighborhood

3. Optimum Global Thresholding, Otsu's Method

The underlying principle in this method is that by selecting an optimized threshold value for classification of the pixels of an image, the pixels can be labeled with a large intensity variation. Hence, the emphasis in this method is laid on finding an optimum threshold value that can be used for segmentation.

Otsu's algorithm can be presented as follows [4]:

1. Compute the normalized histogram of the input image by using following equation:

$$P_i = n_i / MN \quad 3$$

Where n_i is the number of pixels with intensity i , MN is total number of pixel.

2. Set through possible threshold as shown in equation :

$$T(k) = k, \quad 0 < k < L-1 \quad 4$$

Where L is the maximum intensity level in the image .

3. Compute the cumulative sum by using equation :

$$P(k) = \sum_{i=0}^k P_i \quad 5$$

4. Compute the cumulative means by using equation:

$$m(k) = \sum_{i=0}^k (iP_i) \quad 6$$

5. Compute the global intensity mean by applying equation:

$$m_G = \sum_{i=0}^{L-1} (iP_i) \quad 7$$

6. Compute the between class variance by using equation:

$$\sigma_B^2 = \frac{(P(k).(1-P(k)))^2}{P(k).(1-P(k))} \quad 8$$

7. Compute the Global variance by using equation:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 P_i \quad 9$$

8. Obtain the Otsu threshold as in equation:

$$K = \max(\sigma_B^2) \quad 10$$

9. Obtain the separability measure by using equation:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} \quad 11$$

In certain types of images, especially when there is a presence of dark contrast in the foreground formal Otsu's algorithm is not efficient as shown in fig 3. Hence, it needs to be modified.

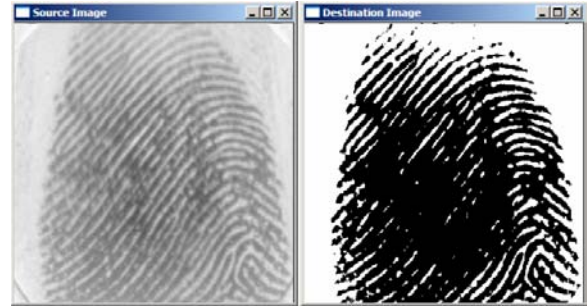


Fig 3 Segmentation using original Otsu's Algo[4]

Modified Otsu's Method: In this section we will discuss a new version of Otsu method. The developed algorithm seeks locally in small regions of the image and find corresponding optimal threshold for every region. The modification is illustrated as follow:

1. Apply original Otsu to the image and save the obtained segmented image into I_1 .
2. Subdivide I_1 into no overlapping rectangles.
3. Subdivide the original image with the same previous subdivision and save into I_2 .
4. Calculate the mean gray level for each subdivision in I_1 .
5. Check in every partition the mean in I_1 .
6. If the mean for I_1 partition is 90% white, assign the whole partition as background.
7. Otherwise, apply again original Otsu's global method in the subdivision in the I_2 to get the optimal threshold to this partition. Fig 4 Shows the results of modified Otsu method.



Fig 4 Segmentation Using Modified Otsu Method

4. Grow Cut Segmentation

In this scheme, the user specifies certain image pixels (seed pixels) that lie within the object. These seed pixels guide the segmentation process and provide user interaction so that the user gets the desired results. In “Grow-Cut” Segmentation, the seed pixels are labeled foreground pixels and the segmentation process starts with these pixels and tries to merge all the other pixels that lie within the same object.

Cellular Automata is used to assign labels to the pixels of the image. With each iteration of the transition function, some pixels are labeled and the process is carried out till all the pixels are labeled as either foreground or background. The user interactivity of this method helps in correcting the method in areas which are difficult to segment but there is no need to specify any seed pixels in regions which can be segmented easily.

Some important advantages of this method are:

1. Capable of solving moderately hard segmentation tasks.
2. Works with images of any dimension $N \geq 1$.
3. Performs multi-label image segmentation (the computation time is not directly affected by the number of labels).
4. Is extensible, allowing construction of new families of segmentation algorithms with specific properties.
5. Interactivity - as the segmentation is refined with each iteration.
6. The algorithm is simple in both understanding and implementation.
7. Using cellular automata allows fast parallel implementation.

Method Details:

A digital image is a two-dimensional array of $k \times m$ pixels. An unlabelled image may be then considered as a particular configuration state of a cellular automaton, where cellular space P is defined by the $k \times m$ array set by the image, and initial states for $\forall p \in P$ are set to:

$$I_p = 0, \theta_p = 0, \vec{C}_p = RGB_p;$$

12

where triplet (I_p, θ_p, C_p) is defined as the label I_p of the current cell, ‘strength’ of the current cell q_p , and cell feature vector $\sim C_p$, defined by the image. Without loss of generality we will assume $q_p \in [0;1]$, and RGB_p is the three dimensional vector of pixel’s p color in RGB space. The final goal of the segmentation is to assign each pixel one of the K possible labels [6].

When user starts the segmentation by specifying the segmentation seeds, the seeded cells labels are set accordingly, while their strength is set to the seed strength value. This sets the initial state of the cellular automaton.

At iteration $t + 1$ cell labels I_p^{t+1} and strengths q_p^{t+1} are updated as follows:

Code 1 Automata evolution rule

```
// For each cell...
for  $\forall p \in P$ 
  // Copy previous state
   $I_p^{t+1} = I_p^t$ ;
   $\theta_p^{t+1} = \theta_p^t$ ;
  // neighbors try to attack current cell
  for  $\forall q \in N(p)$ 
    if  $g(\|\vec{C}_p - \vec{C}_q\|_2) \cdot \theta_q^t > \theta_p^t$ 
       $I_p^{t+1} = I_q^t$ 
       $\theta_p^{t+1} = g(\|\vec{C}_p - \vec{C}_q\|_2) \cdot \theta_q^t$ 
    end if
  end for
end for
```

where g is a monotonous decreasing function bounded to $[0,1]$, e.g.

$$g(x) = 1 - \frac{x}{\max\|\vec{C}\|_2};$$

13

The results of this method on a colored image are shown in fig 5:

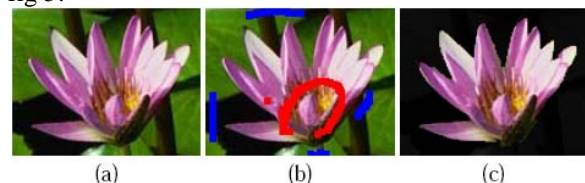


Fig 5 (a) Source Image (b) User specified Seeds (c) Segmentation results

5. Image segmentation using Label Priors: This method uses the prior label knowledge obtained from the seed pixels labeled by the user to guide the segmentation process. Here a 3-D image is considered so a six cell Von-Neumann neighborhood is taken given by-

$$N(\mathbf{v}) = \|\mathbf{v} - \mathbf{q}\|_1 = \sum_{i=1}^3 |v_i - q_i| = 1 \quad \mathbf{q} \in \mathbf{R}^3 \quad 14$$

6. Cellular Learning Automata-Based Color Image Segmentation using Adaptive Chains

This method uses soft and hard segmentation processes. In the soft segmentation process, a cellular learning automaton analyzes the input image and brings together the pixels that are enclosed in each region to generate a soft segmented image. Adjacency and texture information are encountered in the soft segmentation stage. Soft segmented image is then given as input to the hard segmentation process which outputs the final segmentation result. By Cellular Learning Automata, we mean that the CA adapts to its environment after sometime.

Cellular Learning Automata

A learning automata (LA) is an automaton that interacts with its environment in order to improve its performance. It is connected to the environment in a feedback loop [7].

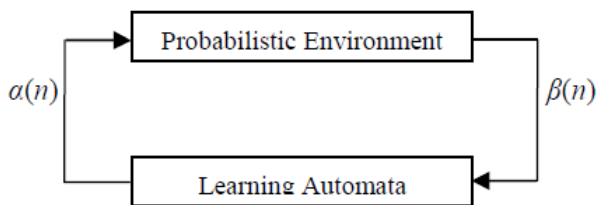
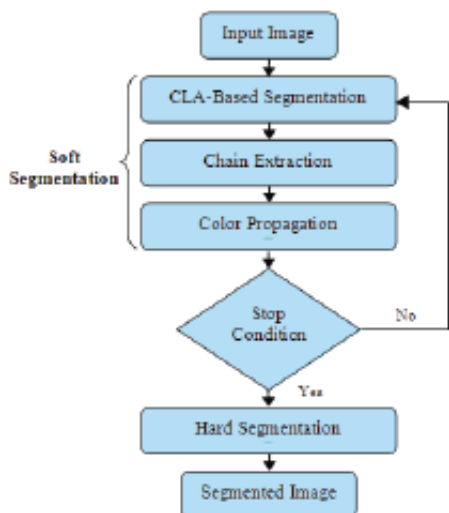


Fig 6 Feedback connections of Automata and its environment

The automata takes an action that results in a response (reinforcement signal) from the environment. Response can either be in terms of a reward or a penalty. The automaton uses the reinforcement signal and its knowledge base to decide its next move or action. The automaton modifies its action probability vector depending upon the reinforcement signal received.

A CLA consists of a large number of simple learning agents. A learning automaton is assigned to each cell that determines the state of the cell on the basis of its action probability vector.

Flowchart for the Method:



Method Details:

The input image is first processed by CLA. In this method a 2-D learning automata with dimensions similar to the input image is created. Each cell of CLA is allocated with a dynamic structure LA. Each learning automaton carries out eight events each corresponding to eight neighbors of the central cell which has a selection probability. Selection probability of each event shows the similarity of the central cell to its neighbors. Selection of an event by central cell means that the central cell in input image and the selected neighbor lie in the same region.

First, for each pixel $I(x,y)$ at location (x,y) of input image "I", Calculate the Euclidean distance of pixel:

$$D_i = I(x,y) - I(x',y') \tag{15}$$

where $I_i(x',y')$ is the i th adjacent pixel at location (x',y') . The rule to reward or penalise the central cell is- If

$c \times Di(x,y) \leq Dm(x,y)$ [7], then reward else penalise. At the end of first step the pixels that lie in the same segment propagate their intensity information together and hence attain a stable state. After this an algorithm is applied for extracting chain of pixels which are similar. For each cell in the input image a chain of the cells which resemble a given cell is computed. The cells in the chain are then updated with a weighted average value of all the cells in the chain. The chain propagation stops if the current cell gets a punishment or if a cycle is formed.

The algorithm for extracting the chain is as follows [7].

Extract_Chain (x, y)

```

Set current cell to (x,y).
Assign pixels_list to an empty set.
Repeat
  Add current cell to pixels_list.
  Set alpha to selected action in current cell.
  Set current cell to (x',y') which (x',y') is
  the neighborhood of current cell
  corresponds to action alpha.
Until (LA in current cell is penalized or a cycle is
detected in pixels_list)
Return pixels_list.
End Extract_Chain
  
```

For each cell in the input image a chain of the cells which resemble a given cell is computed. The cells in the chain are then updated with a weighted average value of all the cells in the chain [7].

$$I'(x,y) = \frac{\sum_{i=1}^n L_i(x,y) | (w_i \times I\{L_i(x,y)\})}{\sum_{i=1}^n L_i(x,y) | w_i} \tag{16}$$

In the second phase a threshold value Θ is selected which is equal to the maximum distance between pixels of a segment. If the distance between two adjacent pixels is less than this threshold value, these two pixels are considered to belong to the same segment otherwise they are considered to belong to different segments. After this the average of the gray scale values is considered to be the gray scale value of the entire segment.

9. Watershed Segmentation:

This method can be explained by a metaphor based on behavior of water in a landscape. When it rains, drops of water falling in different regions will follow the landscape downhill. The water will end up at the bottom of valleys. In this case, the depth of the lakes in valleys is not considered. For each valley there will be a region from which water drains into it. In other words, each valley is associated with a catchment basin, and each point in the landscape belongs to exactly one unique basin. A grey level image may be seen as a topographic relief, where the grey level of a pixel is interpreted as its altitude in the relief.

In this method a valued graph $G = (V, E, f)$ is taken where, $f: V \rightarrow R$ is the value function on vertices. The algorithm used is [8]:

1. For all minima: s_i with $i \in [1; k]$
2. $\lambda(s_i) = 0$ and for all $v \neq s_i, \lambda(v) \leftarrow -\infty$
3. label $(s_i) \leftarrow i$ and for all $v \neq s_i, \text{label}(v) \leftarrow 0$
4. for all $p \in V$
5. $U^t = \min_{q \in N_p} \{ \lambda^t(q) + f(p) \}$
6. $\lambda^{t+1}(p) = \min [\lambda^t(p), U^t]$
5. label^{t+1}(p) = label { min [$\lambda^t(p), U^t$] }
6. End for

10. Conclusion

This paper reports a detailed survey of the various segmentation algorithms of CA. The survey also provides a vivid sketch of the different theoretical developments which have taken place over the years in the CA research field. These developments have established the immense potential of CA in modeling different applications, thus spreading the appeal of cellular automata over a wide cross-section of researchers.

11. Acknowledgement

The authors would like to thank Prof. Anand Prakash Shukla Associate Professor, KIET, Ghaziabad for their kind guidance and support.

References:

- [1] Ulam, S. Some ideas and prospects in biomathematics. Ann. Rev. Biophys. Bioengin. 1: 277-291, 1963.
- [2] Von Neumann, J. Theory of Self-Reproducing Automata. University of Illinois Press, IL, 1966.
- [3] Popovici, A., and Popovici, D. Cellular automata in image processing. In Fifteenth International Symposium on Mathematical Theory of Networks and Systems. 2002.
- [4] C. Ren, Y. Yin, J. Ma, and G. Yang, "A Linear Hybrid Classifier for Fingerprint Segmentation," IEEE Fourth International Conference on Natural Computation, Jinan, China, 2008..
- [5] Thesis on fingerprint segmentation by Diala Jomaa, 2009.
- [6] Vladimir Vezhnevets, Vadim Konouchine "GrowCut" -- Interactive Multi-Label N-D Image Segmentation By Cellular Automata Graphics and Media Laboratory, Moscow State University, Russia, 2005.
- [7] Ahmad Ali Abin, Mehran Fotouhi, Shohreh Kasaei, "Cellular Learning Automata-Based Color Image Segmentation using Adaptive Chains", 14th International CSI Computer Conference IEEE, 2009.
- [8] Claude K., Nicolas P. "A Cellular Automaton for Ultra-Fast Watershed Transform on GPU" IEEE, 2008.