



*MRI brain segmentation, CT tissue segmentation,  
Cellular Automaton, image processing,  
medical informatics*

Rafał Henryk KARTASZYŃSKI \*, Paweł MIKOŁAJCZAK \*\*

## **CABRS – CELLULAR AUTOMATON BASED MRI BRAIN SEGMENTATION**

In this article new approach to the MRI brain segmentation is presented. It is based on the Cellular Automaton (CA). The method is truly interactive, and produces very good results comparable to those achieved by Random Walker or Graph Cut algorithms. It can be used for CT and MRI images, and is accurate for various types of tissues [2]. Discussed here version of the algorithm is developed especially for the purpose of the MRI brain segmentation. This method is still in the phase of development and therefore can be improved, thus final version of the algorithm can differ from the one presented here. The method is extensible, allowing simple modification of the algorithm for a specific task. As we will also see it is very accurate for two-dimensional medical images. Three-dimensional cases require some unsophisticated data post processing [5], or making some modifications in the manner in which the automaton grows into the third dimension from the two-dimensional layer.

### **1. INTRODUCTION**

The method is based on cellular automata, firstly introduced by Ulam and von Neumann in 1966 [7]. It can be used to solve difficult segmentation problems, furthermore it is multilabel – segments many object simultaneously (computation time does not depend on the number of labels). It is interactive: requires the user to provide starting points for the algorithm (not many seeds are needed and their entering is not laborious), but in turn enables him to observe the segmentation process and make modifications in it. Interactivity is very important for physicians who like to have some (often large) influence on medical images processing. Furthermore, a radiologist will be able to place seed points very accurately and in characteristic places of a specific organ (the reason why will be explained later), and will check the correctness of segmentation afterwards.

### **2. CELLULAR AUTOMATON**

A cellular automaton is a discrete model studied in the computability theory, mathematics, and theoretical biology [4]. It consists of an infinite, regular grid of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the

---

\* Department of Computer Science, Maria Curie-Skłodowska University, Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland; e-mail address: hatamoto@goblin.umcs.lublin.pl

\*\* Department of Computer Science, Maria Curie-Skłodowska University, Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland; e-mail address: mikfiz@goblin.umcs.lublin.pl

state of a cell at time  $t$  is a function of the states of a finite number of cells (neighbourhood) at time  $t-1$ . These neighbours are a selection of cells relative to the specified cell and do not change. Though the cell itself may be in its neighbourhood, it is not usually considered a neighbour. Every cell has the same rule for updating, based on the values in this neighbourhood. Each time the rules are applied to the whole grid a new generation is produced.

Formally, cellular automata is triplet

$$A = (S, N, \delta), \quad (1)$$

where

$S$  – not empty states set

$N$  – neighbouring system

$\delta$  – transition function describes way of calculation cell state in time  $t+1$  basing on state of its neighbours in time  $t$ .

State of cell  $p$ , on the other hand also consists of three values:

$$S_p = (l_p, \theta_p, I_p), \quad (2)$$

where

$l_p$  – current cell label

$\theta_p$  – cell strength, being real number and we may assume that

$I_p$  – is intensity (value) of pixel (or voxel) in image corresponding to cell  $p$ .

### 3. APPLICATION OF CELLULAR AUTOMATON FOR SEGMENTATION

In our case cellular automaton is a sheet of graph paper (for two dimensional images), where each square is a cell, and each cell corresponds to a pixel of image being segmented. In three dimensional cases we would have a set of two dimensional sheets placed one on another. Because this segmentation algorithm is multi-label each cell state consists of one of the labels of areas we are segmenting plus neutral territory label ( $L$  possible labels). During the evolution of automaton other cells slowly conquer neutral territory. Obviously, each cell has eight neighbours on the same plane, and if we are dealing with a three dimensional case, there are also eighteen ones on the planes above and beneath. In a more general instance we may use, for example, von Neumann's:

$$N(p) = \left\{ q \in Z^n \quad : \quad \|p - q\|_1 := \sum_{i=1}^n |p_i - q_i| = 1 \right\} \quad (3)$$

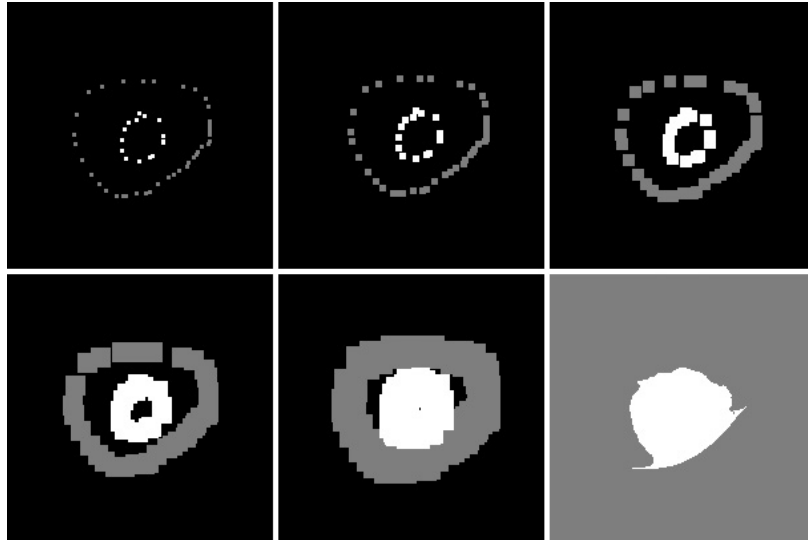
or Moor's

$$N(p) = \left\{ q \in Z^n \quad : \quad \|p - q\|_\infty := \max_{i=1..n} |p_i - q_i| = 1 \right\} \quad (4)$$

neighbouring system.

The algorithm requires the user to provide starting points (seeds) for segmentation. In the simplest case two kinds of seeds should be given, i.e. seeds corresponding to the object we are

segmenting and corresponding to its surroundings (see Fig. 2, 4). Of course, more than two classes can be provided, thus segmentation of several objects will be done.



*Fig. 1. Evolution of automaton segmenting tumour from Fig.4.:  
black – neutral territory, white – organ labelled bacteria, grey – background labelled bacteria. Evolution time steps are as follows (from upper left): 1, 3, 6, 11, 20, 40 (lower right).*

Let us use a biological metaphor to supply intuitive explanation of automata evolution. An image (two or three dimensional) is a discrete universe (set of pixels / voxels) which has to be labelled in order to extract its part. Adding starting seed points is equal to the creation of new bacteria's colonies matching image parts, pixels not marked as seeds are considered as neutral territory. The labelling process can be treated as a struggle for domination of  $L$  different types of bacteria, grouped in several colonies (seeds). Because time in this automaton is discrete, in each time step every bacterium tries to occupy the neighbouring pixel not belonging to its colony. After every step each colony grows, and soon every pixel of the image belongs to one of them. The evolution process is finished when, in one of the steps, no new cells have been conquered.

After the above explanations we can write a draft of program code realizing the automata evolution:

```

while not StopCondition do
begin //count automata state in time t+1 basing on its state in time t
  for each image pixel/voxel p
  begin //neighbours try to conquer cell p
    for each q being neighbour of p
    begin
      if  $g(|I_p - I_q|) \cdot \theta_q^t > \theta_p^t$  then
        begin //attack succeeded cell p has now label of cell q
          //and new strength
           $l_p^{t+1} = l_q^t$ 
           $\theta_p^{t+1} = g(|I_p - I_q|) \cdot \theta_q^t$ 
        end
      end
    end
  end
end

```

```

end;
end;
end;
end;

```

Where  $g$  is a monotonous decreasing function on  $[0, 1]$ . We have chosen a simple one  $g(x) = 1 - \frac{x}{\max I}$ , where we count the maximal density in the whole image or data set.

StopCondition can be, as mentioned before, case when during one time step no new cells are being conquered and the cell state ceases to change. Unfortunately this approach may lead to an execution of a lot of time steps, some of which are completely unnecessary. Because in most cases we wish to segment one organ which is a small part of the image, the better way is to narrow the calculations to a small box containing the interesting part of image, thus shortening the evolution process. Further improvement can be made by consideration of changes (to be more precise: lack of state changes) close to object boundaries. We put seed points (the ones belonging to the object and the ones belonging to its surroundings) near the boundary of the organ (on both the inner and outer side), so during the evolution, the situation on the organ boundary is quickly stabilized and the process can be stopped, and the interior of the object (if not yet conquered by cells corresponding to it) can be automatically filled. Another way to save time is to execute a fixed number of time steps. How many? It depends on the type of segmentation we perform (number of seed points, distance between outermost points, etc.), and can be estimated empirically. For example, results presented in this article (for two-dimensional cases) were achieved with forty time steps (no noticeable differences have been found between forty and, for example, one hundred steps).

#### 4. RESULTS OF TWO-DIMENSIONAL SEGMENTATION

We will now present some results of cellular automata segmentation. As it can be seen it is a very accurate method allowing to segment different tissues from various types of medical images. It should be stressed that accuracy of segmentation depends strongly on the appropriate choice of seed points (this will be explained later).

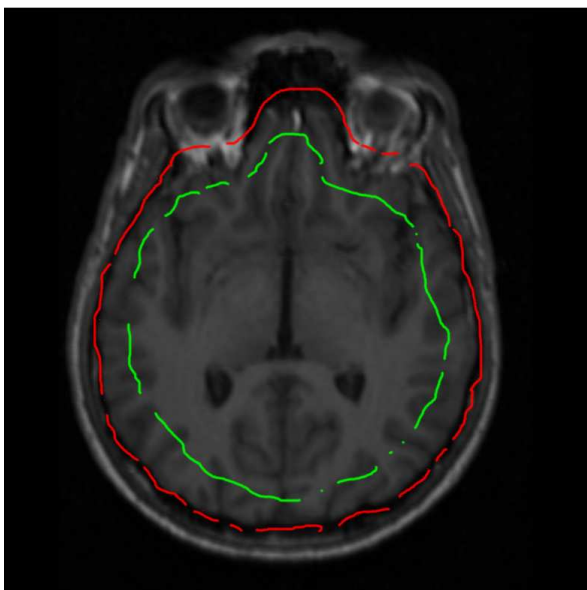


Fig. 2. Seeds for MRI brain segmentation

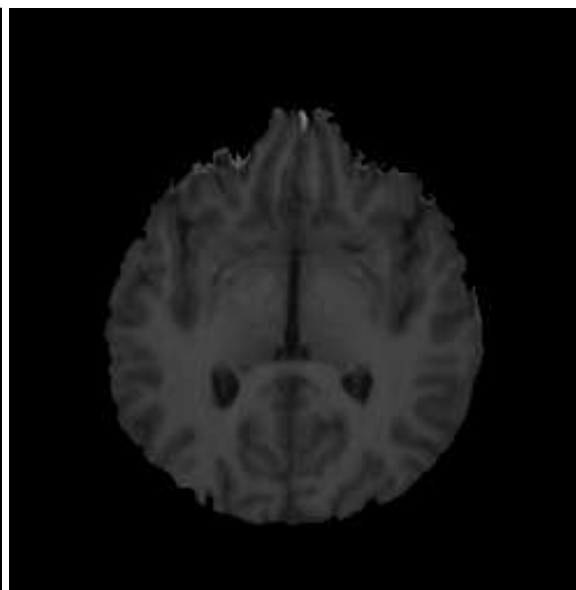


Fig. 3. Segmented MRI brain

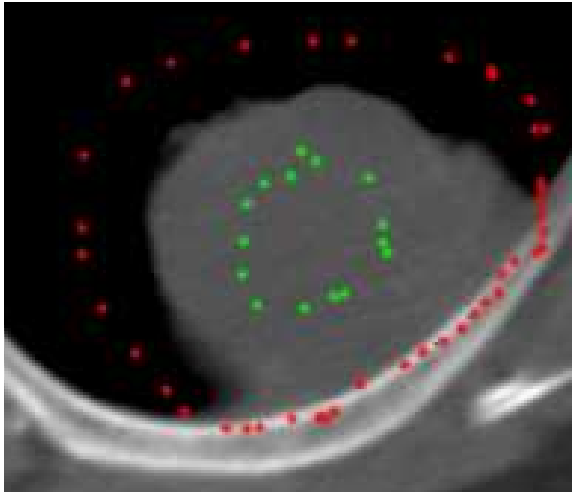


Fig. 4. Seeds for lung tumour segmentation

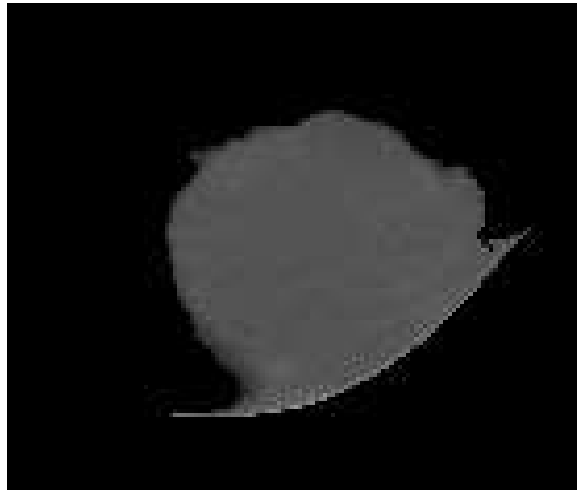
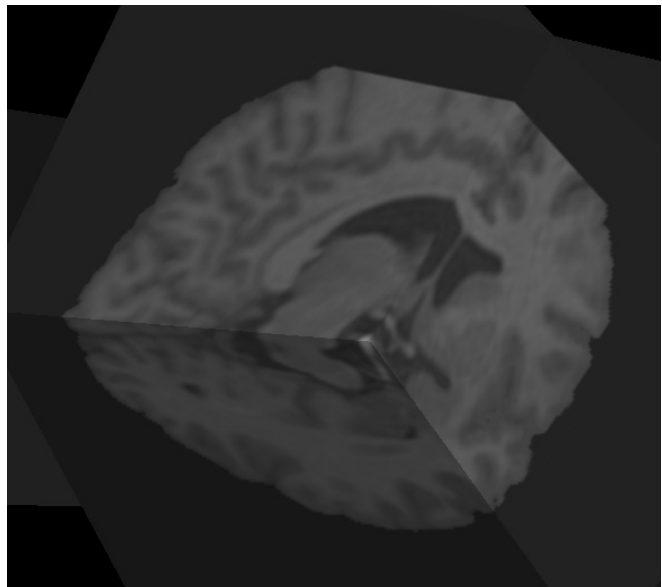
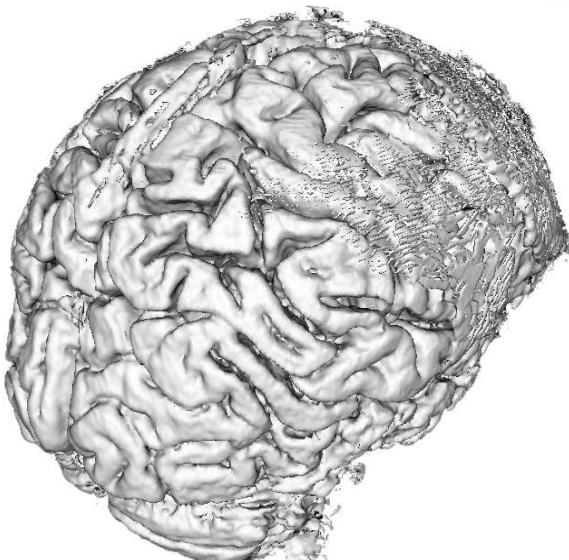


Fig. 5. Segmented lung tumour

## 5. RESULTS OF THREE-DIMENSIONAL SEGMENTATION

When we deal with three-dimensional data sets we would like to set seed points for one layer and let the automata segment the rest of the organ. The presented algorithm can be easily applied to two and three dimensions. If seeds for only one layer are given, 3D segmentation turns out to be rather accurate, though sometimes nearby tissues are recognized as a part of the segmented organ (see Fig. 6). This problem can be easily fixed by post processing of the data set. Applying morphological operations [6]: dilation and erosion filtering, destroys small connections between the main organ and the oversegmented tissue. Next connected components labelling [1][3] is performed to select the only organ of interest to us. As we can see in Fig. 7 such a processing is very effective. Apart from post processing of the data set, some pre processing (morphological operations) can also be necessary (thresholding at the Otsu level). The main difficulty may be finding a proper place to put the seed points at.



## 6. CONCLUSIONS AND FUTURE WORK

Let us firstly discuss the correct introduction of seed points. In the case of 2D this is rather simple and does not need additional explanation. One should remember to mark (as seeds) all (most of) characteristic pixels of the object and its surroundings – background. When this is done properly we may be sure that each point will grow in proper direction and the boundary will be correctly found. In 3D the same rules apply, but some further guidelines must be given. When choosing layers in the data set to place seeds in, we should choose ones which have the most characteristic features of the tissue we are segmenting and its surroundings. Seeds of the background ought to be placed not only near the boundary of the object, but also on different tissues surrounding it (even far away from the object). We must remember that we are dealing with a three-dimensional data set and tissues (organs) distant from the object on the current slice, can be in touch a few centimetres above. This fact should also be taken into consideration when selecting a place for seed points.

As we could see, on the effects of the two-dimensional segmentation, sometimes the segmented boundary is ragged (see Fig.4). When we need to capture the smallest detail, this is acceptable in most cases, but it may also be an unwanted artefact. To achieve smoother boundaries a slight alteration of the transition rule can be put forward. Let us call the cells of a different label than the examined one, the enemies of that cell. Now, cells having more enemies than E1 are prohibited from attacking their neighbours, and cells that have more than E2 enemies are automatically conquered by the weakest of their neighbours. Values E1, E2 control boundary smoothness and should have a value from 6 to 9 (no smoothing) for the Moor neighbouring system. This modification has not been tested for three-dimensional cases.

The presented algorithm is very accurate. Its drawback is a long execution time, but as we could see this can be levelled by making some simple modification. In the three-dimensional case post processing is required to erase artefacts which could sometimes appear during segmentation. Nonetheless, the method itself is very promising and should be developed further to improve its performance and find appropriate modifications for specific purposes. We will also accurately compare its performance with different segmentation approaches, such as Random Walker, Graph Cut or GrabCut. The first test shows a small difference in results, whereas the CA appears to be quicker and is a lot simpler in implementation.

## BIBLIOGRAPHY

- [1] BERTHOLD KLAUS PAUL HORN., Robot Vision., IMT Press, 1986.
- [2] KARTASZYŃSKI R., MIKOŁAJCZAK P., CATS – Cell Automata based Tissue Segmentation of two- and three-dimensional CT and MRI medical images, *Annales Informatica*, UMCS, Lublin 2006
- [3] HEIMANN, T., THORN, M., KUNERT, T., and MEINZER, H.-P., New methods for leak detection and contour correction in seeded region growing segmentation., In 20th ISPRS Congress, Istanbul 2004, *International Archives of Photogrammetry and Remote Sensing*, vol. XXXV, 317–322, 2004.

- [4] HERNANDEZ, G., HERRMANN, H. J., Cellular automata for elementary image enhancement, *CVGIP: Graphical Model and Image Processing* 58, 1, pp. 82–89, 1996.
- [5] RAFAEL C. GONZALEZ, RICHARD E. WOODS., *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.
- [6] SERRA. J., *Image Analysis and Mathematical Morphology*, Academic Press, 1982.
- [7] VON NEUMANN, J. *Theory of Self-Reproducing Automata*, University Of Illinois Press. Ed. And Completed by A. Burks., 1966