A Rule-Based Approach to Stroke Lesion Analysis from CT Brain Images

Milan Matešin, Sven Lončarić
Department of Electronic Systems and Information Processing
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
E-mail: milan.matesin@fer.hr, sven.loncaric@fer.hr

Damir Petravić
Department of Neurology
University Hospital Rebro
Kišpatićeva 12, 10000 Zagreb, Croatia
E-mail: damir.petravic1@zg.tel.hr

Abstract

This paper presents a method for automatic segmentation and labeling of computed tomography (CT) head images of stroke lesions. The method is composed of three steps. The first step is automatic determination of head symmetry axis, with possibility of manual improvement of result, if necessary. Symmetry axis calculation is based on moments. In the second step the seeded region-growing (SRG) algorithm is used to segment input image into number of regions having uniform brightness. Features of these regions, such as brightness, area, neighborhood and relative position to symmetry axis are used to create facts for a rule-based expert system. Based on created facts and pre-defined rules as input, the rule-based expert system is used in the third step to label regions as background, scull, gray/white matter, CSF and stroke. Experimental results have been conducted and have demonstrated the feasibility and accuracy of the proposed method.

1. Introduction

Segmentation and labeling of various tissue types in medical imagery are important objectives in study of anatomical structures, diagnosis or pre-operative planning. These tasks are difficult as they require complex application-specific knowledge. In addition, image artifacts such as partial volume effects and noise make this task problematic [1]. Analysis of brain images has been an important task in recent years and has received a lot of attention in literature. In particular, analysis of brain stroke lesions has shown to be difficult. Stroke lesions appear somewhat darker (hypodense) in CT images, but often with no distinct boundary, so even experienced radiologists do not always agree about the extent of the lesion. Another problem for computer system is finding the position of the stroke. It is obvious that some knowledge must be used to successfully label these images.

A number of authors have used intelligent systems for segmentation of medical images [2]. Research on brain image segmentation using rule-based expert systems has been presented in [3, 4, 5, 6]. Expert systems have also been used for segmentation of lung boundaries [7].

In this paper we propose a new technique for analysis of brain stroke from CT images. The proposed method for stroke analysis consists of three main steps. The rule-based expert system used for region labeling uses the knowledge about symmetry properties of the brain. For this purpose symmetry axis of the brain is determined in the first step of the procedure. In the second step, the image is segmented into regions having uniform brightness. The segmentation step utilizes the seeded region growing algorithm. The obtained regions are labeled in the third step of the analysis. Labeling procedure is realized using a rule-based expert system. Facts for the expert system contain various properties of the segmented image regions. A set of rules is defined that contains high-level knowledge about the anatomical structure and organization of CT brain image containing stroke lesions.

The paper is organized as follows. The method for computation of brain symmetry axis is described in Section 2. Segmentation using SRG algorithm is described in Section 3. Rule-based system is discussed in Section 4. A discussion of results and conclusion are provided in the last section.

2. Finding symmetry axis

Input to our system are 256-valued grayscale CT images of human head. Segmentation and labeling are focused on individual slices. Before any other processing can take place, the system must determine the symmetry axis of the head. From original slice, the head region is extracted by means of thresholding. The region consists of the skull and internal brain structures. The center of mass $(\bar{x}, \bar{y})$ and the angle of symmetry axis (Φ) are computed for the segmented binary image $S(x, y)$ using moments:

$$m_{p,q} = \sum_{(x,y)\in S} x^p \cdot y^q$$

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}}, \quad \bar{y} = \frac{m_{0,1}}{m_{0,0}}$$

$$\mu_{p,q} = \sum_{(x,y)\in S} (x-\bar{x})^p \cdot (y-\bar{y})^q$$

$$\Phi = \frac{1}{2}\arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right)$$

Where $m_{p,q}$ is $(p+q)$-th order moment, and $\mu_{p,q}$ is $(p+q)$-th order central moment.

The operator can modify the result by adjusting generated coordinates of center of mass and angle of symmetry axis.

3. Seeded region growing

Seeded region growing algorithm (SRG) is first described in [8]. It is different from ordinary region growing, in which the general procedure is to compare one pixel to its neighbor. If a criterion of homogeneity is satisfied, the pixel is said to belong to the same class as one or more of its neighbors. That pixel then becomes new seed. The result is controlled by tuning homogeneity parameters.

In SRG, algorithm is controlled by choosing a number of pixels, called seeds. No additional tuning is necessary. In our case, every seed is in its own set, A_1, A_2, ..., A_n. Four-neighbors of seed points (with some additional information) are then put in so-called "sequentially sorted list" (SSL). When a pixel (x) is being put in SSL, the algorithm determines if that pixel has some neighbors that already belong to some set. If so, for each of those sets A_i the difference $\delta_i(x)$ is computed:

$$\delta_i(x) = |\, g(x) - \text{mean}\,(A_i)\,|$$

where $g(x)$ is gray level value of pixel x, and mean(A_i) is the mean value of brightness of all pixels belonging to set A_i. Now, the minimum of $\delta_i(x)$ is found, call it $\delta(x)$. Pixel coordinates, minimum value, and region label are saved to SSL. SSL is sorted in descending order of differences. The pixel with the smallest difference is first in the list, and so on. In addition to this, when adding pixel to SSL, the algorithm updates running mean of the winning set.

After these initial operations, the process evolves inductively from the seeds. In each step one pixel is taken from SSL (first one, with smallest difference) and added to the appropriate set. Neighbors of that pixel that are still unlabeled are found and put to SSL. The algorithm iterates until all pixels are labeled.

The algorithm for implementation of SRG can be expressed as follows:

1. Label seed points according their initial grouping.
2. Put neighbors of seed points in the SSL.
3. While the SSL is not empty:
4. Remove first point y from SSL
5. Label y to appropriate set.
6. Put all unlabeled neighbors of y to SSL.

The key-point in SRG algorithm is the fact that SSL is sorted. Because of that, those pixels that are undoubtedly part of some region are first labeled and have opportunity to give its neighbors the same label. But if pixels are on region boundaries (and because of that have large differences), they are labeled last.

In our method, we use 26x26=676 seed points organized in a rectangular grid. One half of seed points are on the left side of symmetry axis, the other half is on the right side. It is important that the number of seeds is an even number, so we can later compare each region with its symmetric region. The more the regions, more facts will be generated and process will take more time and memory. Seeds are positioned in the head area (we know where the head is from the first step) and rotated appropriately.

The original image with computed symmetry axis is in Figure 1. The result of SRG segmentation is shown in Figure 2.

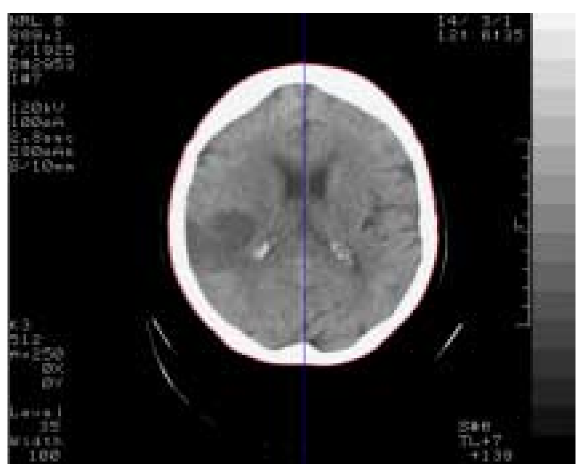

Figure 1. Original image

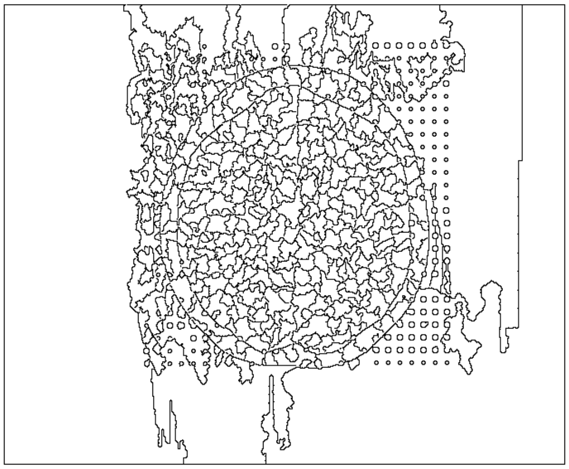

Figure 2. Result of SRG segmentation

4. Rule-based labeling

From the result of SRG segmentation, a number of facts are created. Each fact represents one region. The facts are saved in a file and loaded into expert system shell CLIPS [9]. Each fact has 9 slots, each representing some region property. The slots are described in Table 1.

Table 1. Information slots of facts

slot name	description
id	region identification number (1-676)
label	how region is labeled (scull/CSF…)
neigh	ids of neighbors
intensity	brightness (verybright, bright, …)
area	area in pixels of region
x	x coordinate of region
y	y coordinate of region
side	on which side of sym. axis is region
includes	which regions are merged with this one

A sample fact could look like this:

```
(region (id 501) (label unknown)
(neigh 488 502 513)
(intensity bright)(area 43)
(x 7)(y 12)(side 0)(includes 501))
```

"id" is region identification number. Regions are numbered from 1-676 (total number of regions).

"label" is region label. Every region on start has label "unknown". Other possible labels are "background", "scull", "brain", "csf", "maybestroke", "normal" (gray/white matter), "stroke".

"neigh" is list of neighbor identification numbers. Neighborhood in this case is 4-neighborhood.

"intensity" is description of region brightness. There are five possible values: "verybright", "bright", "medium", "dark", "verydark". From several pictures and their histograms, four thresholds are chosen to delineate these classes.

"area" is area in pixels of given region.

"x" and "y" are x and y region coordinates. Coordinate system is presented in Figure 3. (for 10x10=100 seeds; vertical black line representing symmetry axis).

"side" is "1" for left, "0" for right.

"includes" is list of all regions that are merged with this one in process of connecting "stroke" regions. At the beginning, "includes" consists just of region's id.

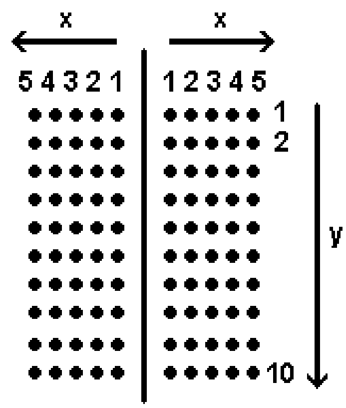

Figure 3. Coordinate system

Every expert system works on facts based on rules. Rules are in "IF-THEN" form. If left-hand-side (LHS) of the rule is satisfied by some fact(s) ("activated"), that rule "fires" and executes actions on right-hand-side (RHS). Actions can include printing to output (file or screen), retracting or asserting new facts or modifying existing one.

There are several stages of rule-based labeling. In each stage, only selected subset of all facts can be activated and fired. This is controlled by "salience" which is a kind of priority in execution of rules in expert system shell. When some rule fires, the agenda (stack that hold information about which rules can fire) is updated with new activations. As long as there are some rules with higher salience, rules with lower salience can't fire. In CLIPS 20001 different levels of salience can be defined. In our program we use 6 levels of hierarchy (and executed in 5 stages) as depicted in Figure 4.

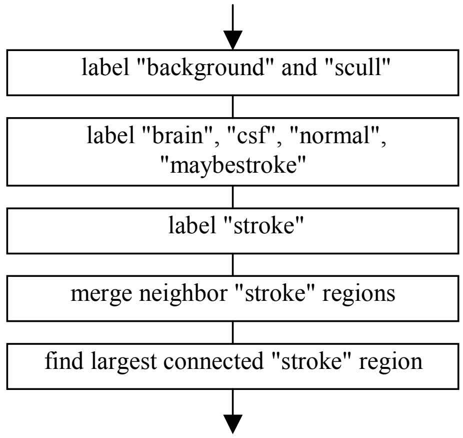

Figure 4. Stages in rule-based labeling

Here are presented some rules from first 3 stages:

1. Rules for "background":
   1.1. Largest "verydark" region is "background".
   1.2. "verydark" or "dark" region, that has "background" region for neighbor, is "background" itself.
   1.3. Region that has only one neighbor, and that neighbor is "background", is "background" itself.
2. Rules for "scull":
   2.1. Largest "verybright" region is "scull".
   2.2. "verybright" region, that has "scull" region for neighbor, is "scull" itself.
3. Rules for "brain", "csf", "normal", "maybestroke":
   3.1. Largest "dark|medium|bright" region that has "scull" for neighbor, is "brain".
   3.2. "brain" region that is "dark" is "csf"
   3.3. "brain" region that is "medium" is "maybestroke"
   3.4. "brain" region that is "bright" is "normal"
4. Rules for "stroke":
   4.1. "maybestroke" region that has no symmetric "maybestroke" region is "stroke".

For example, rule 4.1 in CLIPS syntax looks like this:

```
(defrule rule-stroke "Rule for stroke"
(declare (salience –20))
?fact1 <- (region (label maybestroke) (id ?id) (x ?x1)
                    (y ?y1) (side ?side1))
(not (region (label maybestroke) (side ~?side1)
            (x ?x2&:(< (abs (- ?x2 ?x1)) 2))
            (y ?y2&:(< (abs (- ?y2 ?y1)) 2))))
=>
            (modify ?fact1 (label stroke))
)
```

As we can see, "symmetric" means "all regions on other side, that have x or y coordinate different from x (or y) coordinate of candidate's region no more than one".

The rule for merging neighboring "stroke" regions into one is:

"IF there exist 2 facts (regions), id1 and id2, that are "stroke" and that are neighbors THEN retract fact for id1, add it's area to area of id2, add it's "include" field to "include" field of id2, tell all neighbors of id1 that from now on they have id2 for neighbor, and tell id2 that from now on it has neighbors of id1 for neighbors."

After merging all regions that can be merged, expert system finds largest "stroke" region and report to calling procedure of which areas it consists of ("includes" slot) and what is it's area ("area" slot). In Figure 5 segmented stroke region is shown in black colour.

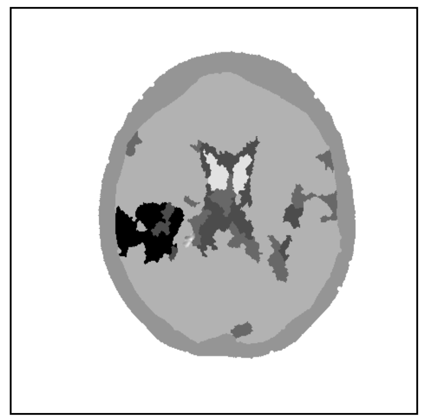

Figure 5. Labeled image

5. Discussion and conclusion

On a PC with 600 MHz Duron processor, the procedure takes less than 30 seconds per slice, thanks to heap implementation of SSL. The expert system is very fast. It performs labeling in just few seconds, for several hundred facts and a dozen of rules. Initial experimental results are encouraging.

If there is no need for manual improvement of symmetry axis, the whole process is fully automatic, but relies on some assumptions. The first assumption is that the largest "white" area is scull, for finding symmetry axis. The second assumption is that "seed grid" is dense enough to "hit" region of interest – stroke lesion. As we could see in several experiments using real data, this is true in most cases. The third assumption is that brightness thresholds are well chosen.

The seeded region growing algorithm has itself certain limitations. Because data in SSL is not updated as every new pixel enters set, it is somewhat incorrect, but very fast. With analysis of medical images, that is not great disadvantage, because there are not anyway clear boundaries. The advantages of SRG are that it is fast, robust and free of tuning parameters once all seeds are placed. Regions produced by SRG closely follow natural boundaries of objects in image. If we had wanted to achieve the same property using classic split-and-merge algorithm, we would have needed several thousand regions.

Use of expert system provides flexibility in experimenting with rules. This system can be easily upgraded by adding new rules for the expert system.

Plans for further work include addition of interactivity in the process of stroke segmentation. The interactivity is necessary due to complexity of stroke segmentation. It will enable users (physicians) to interactively influence the result of segmentation to best fit their expert knowledge about the stroke.

6. References

[1] J.S. Duncan, N. Ayache, "Medical Image Analysis: Progress over Two Decades and the Challenges Ahead", IEEE Transactions on Pattern Analysis and Machine Intelligence, January 2000, vol. 22, no. 1, pp. 85-106.

[2] A. Dhawan, S. Juvvadi, "Knowledge-Based Analysis and Understanding of Medical Images", Computer Methods and Programs in Biomedicine, 1990, vol. 33, pp. 221-239.

[3] D. Ćosić, and S. Lončarić, "Rule-Based Labeling of CT Head Image", In Proceedings of the 6th European Conf. of AI in Medicine Europe (AIME97), Grenoble 1997.

[4] C. Li, D.B. Goldgof, and L.O. Hall, "Knowledge-Based Classification and Tissue Labeling of MR Images of Human Brain", IEEE Transactions on Medical Imaging, 1993, vol. 12, pp. 740-750.

[5] S.P. Raya, "Low-Level Segmentation of 3-D Magnetic Resonance Brain Images – A Rule-Based System", IEEE Transactions on Medical Imaging, 1990, vol. 9, pp. 327-337.

[6] M. Sonka, S.K. Tadikonda, and S.M. Collins, "Knowledge-Based Interpretation of MR Brain Images", IEEE Transactions on Medical Imaging, 1996, vol. 15, pp. 443-452.

[7] M.S. Brown, L.S. Wilson, B.D. Doust, R.W. Gill, and C. Sun, "Knowledge-Based Method for Segmentation and Analysis of Lung Boundaries in Chest X-Ray Images", Computerized Medical Imaging and Graphics, 1998, vol. 22, pp. 463-477.

[8] R. Adams, and L. Bischof, "Seeded Region Growing", IEEE Transactions on Pattern Analysis and Machine Intelligence, June 1994, vol. 16, no. 6, pp. 641-647.

[9] J. Giarratano, G. Riley, "Expert Systems, Principles and Programming", PWS Publishing Company, Boston, 1993.