DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA INFORMATICA

Image Processing and Artificial Vision

Automated 3D segmentation of brain tumor using visual saliency

Docente:

Prof. A. Guerriero

Studente:

Salvatore Bufi

Anno Accademico 2020/2021

Table of contents

List of figures

Introduction

The large size of medical image data, the complexity of Features Of Interest (FOIs), and the necessity to process these on time, both accurately and efficiently, are making the job of doctors and radiologists increasingly difficult. Therefore, it has become essential to develop automated delineation of Regions Of Interest (ROIs) and Volumes Of Interests (VOIs) to assist and speedup medical image understanding.

Medical experts manually segment different regions of interest (ROIs) for detection, diagnosis and planning of treatment. Automated medical image analysis, on the other hand, overcomes human bias and can handle large volumes of data. Variation in blood flow (perfusion) within a tumor causes variation in imaging features like necrosis and contrast. It was observed that regions of tumor that are poorly perfused on contrast-enhanced T1C -weighted images may exhibit areas of low (or high) water content on T2-weighted images and low (or high) diffusion on diffusion-weighted FLAIR (Fluid-Attenuated Inversion Recovery) images. Thus high (or low) cell densities can coexist in poorly perfused volumes, with the creation of perfusion-diffusion mismatches. Regions having poor perfusion and high cell density are of clinical interest, because they contain cells which are likely to be resistant to therapy. This highlights the utility of superimposing multiple channels of MR imaging, like FLAIR, T 2, and contrast enhanced T 1 C components, in identifying and extracting heterogeneous tumor region(s).

Humans can easily identify the salient (or relevant) parts of an image mainly due to the attention mechanism of the human visual system. Computational models of saliency take images as input and generate a topographical map of how salient or attention grabbing each area of the image can be to a human observer.

Algorithms employing the bottom-up strategy detect saliency by using low-level features, like colour, intensity, orientation. Those using the top-down strategy include some learning from the training data involving the position or shape of a salient object.

A visually salient region is typically rare in an image and contains highly discriminating information. This concept is, therefore, expected to have a major bearing towards the fast identification of an ROI or tumor from a medical image.

In this project Multi-sequence MR images are integrated to generate pseudo-colored MRI for efficiently detecting the whole tumor region. A bottom-up saliency detection strategy, incorporating spatial distance between image patches, is used to highlight the salient region(s) in the pseudo-colored MR image. Next saliency map is used to identify and segment the object (tumor) using Superpixel algorithm and the result are improved using the region growing algorithm.

Pseudo-Coloring

Digital colour images are often constructed from three stacked color channels viz. red, green and blue (RGB). These can be decomposed to three gray scale images, in six ways, and recomposed back to the RGB image. For example, let us consider three gray-scale images A, B, C , and let A be assigned to Red, B to Green, and C to Blue channel. Then the six combinations are ABC, ACB, BAC, BCA, CAB, CBA . I assign the three MR sequences (FLAIR, T 2, T 1 C) to RGB for generating a "pseudo-color"MR image, figure 2. This yields a 48 bit MR image containing more information than a single-channel gray-scale image. Thereby a colour MR image is capable of detecting and displaying the whole tumor region as the ROI from such pseudo-colour projections. Figure 1 shows sample MRI sequences and Figure 2 illustrates the corresponding RGB converted MR images.

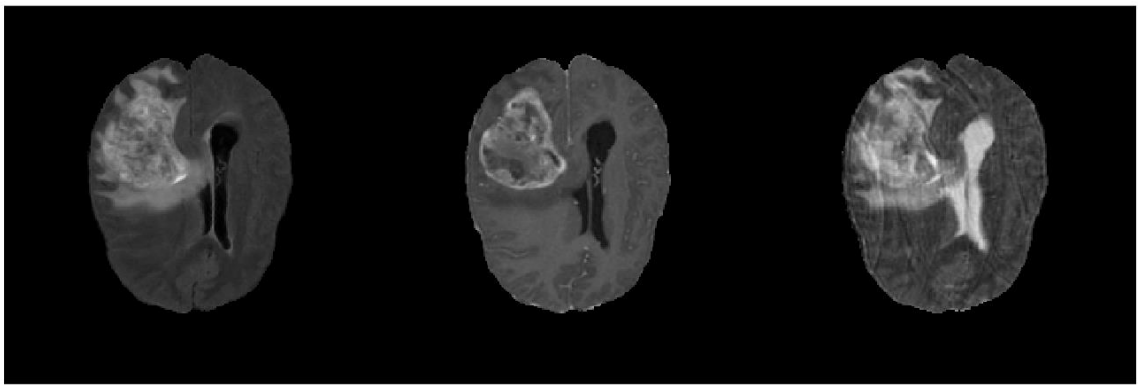

Figure 1: Sample MRI sequences: FLAIR, T1C, T2 left to right.

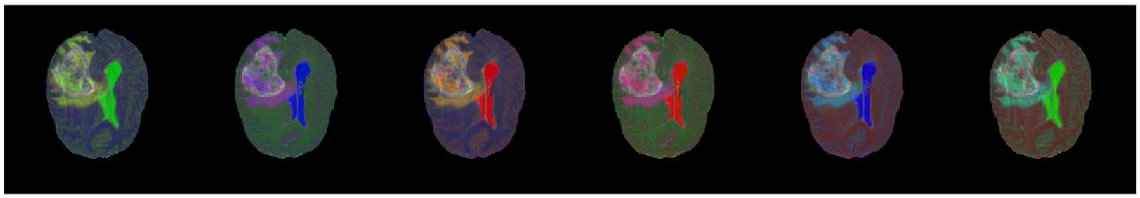

Figure 2: Six corresponding pseudo-colored brain MR sequences. (1)Flair-T2-T1c (2)Flair-T1c-T2 (3)T2-Flair-T1c (4) T2-T1c-Flair (5) T1c-Flair-T2 (6) T1c-T2-Flair.

Since saliency detection algorithm depends on the center-surround difference of a region with its neighbours, based on the pixel color values, a perceptually uniform colour space that decorrelates luminance from chrominance information becomes desirable. Therefore, I choose to transform the

RGB space to the International Commission on Illumination (CIE)- recommended CIE−L ∗a ∗b∗ domain, with the Euclidean distance ΔE representing the color difference. Here:

$$\Delta E = ||Vi - Vj||_2$$

$$Vi = L_i^*, a_i^*, b_i^*$$

$$Vj = L_j^*, a_j^*, b_j^*$$

Converting an RGB image to L ∗a ∗b ∗results in the separation of luminosity (layer L ∗) and chromaticity (layer a ∗, which indicates the position of the colour along the red-green axis, and layer b ∗, which represents the position of the colour along the blue-yellow axis).

Tumor location with saliency

The salient region of an image is defined by one (or more) very important piece(s) of the composition, which make it stand out from its surroundings. An L ∗a ∗b ∗image (of size M ×N) is first transformed to a square image of size w × w. Since the database can contain images of different sizes, these need to be converted to one uniform size (here w = 256).

In order to give less importance to the additional pixels, that do not belong to the original image, I decided to extend the image as shown in Figure 3.

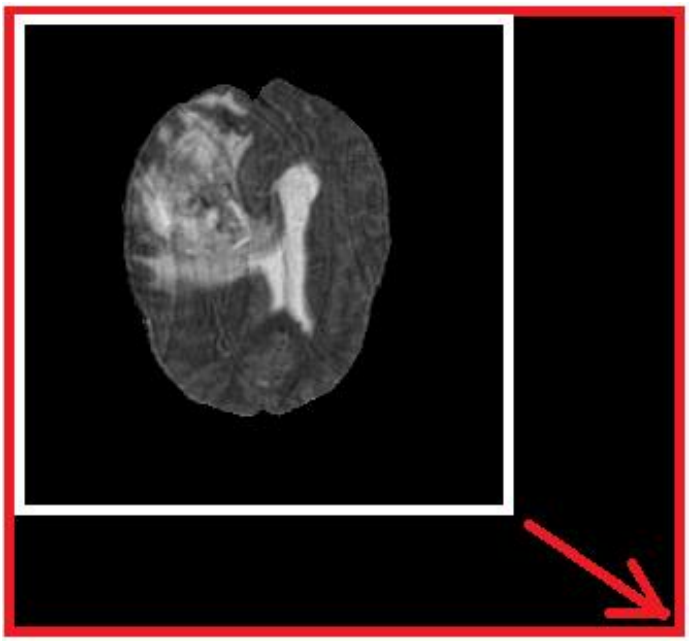

Figure 3: Image resized to 256 x 256 pixels

Then it is decomposed into several non-overlapping blocks Ri (or patches) of size k ×k pixels (where w is a multiple of k), with each being represented by its mean L ∗a ∗b ∗values.

The number of patches (w/k × w/k) correspond to the number of pixels in the saliency map. Let the i-th patch of the image I(Ri), 1 ≤ i ≤(w/k × w/k), be represented by its mean L∗a∗b∗ color values as:

$$R_i^{\bar{L}^*} = \frac{\sum R_i^{L^*}}{k \times k} \qquad R_i^{\bar{a}^*} = \frac{\sum R_i^{a^*}}{k \times k} \qquad R_i^{\bar{b}^*} = \frac{\sum R_i^{b^*}}{k \times k}$$

Next the saliency of each patch is calculated with respect to all other patches in the image based on color differences. The color difference between a pair of patches is defined as the Euclidean distance between the corresponding mean color values of L ∗a ∗b ∗.

Therefore, for patch Ri, the saliency S(Ri) is calculated as the sum of the color difference between $R_i^{\bar{L}^*}, R_i^{\bar{a}^*}, R_i^{\bar{b}^*}$ and $R_j^{\bar{L}^*}, R_j^{\bar{a}^*}, R_j^{\bar{b}^*}$ as shown in the next equation.

$$S(R_i) = \sum_{j \neq i} \frac{1}{1 + d(R_i, R_j)} \times \sqrt{(R_i^{\bar{L}^*} - R_j^{\bar{L}^*})^2 + \left(R_i^{\bar{a}^*} - R_j^{\bar{a}^*}\right)^2 + \left(R_i^{\bar{b}^*} - R_j^{\bar{b}^*}\right)^2}$$

While most salient patches are typically concentrated around spatially adjacent areas, the other patches (which are normally not salient) may be distributed anywhere over the entire image. If a region is salient then the probability for its surrounding regions to be also salient is larger, while the probability of those regions located farther away being salient becomes smaller.

So, the influence of adjacent regions can be considered to be more important when computing the saliency of a region.

Therefore, I incorporate the spatial distance between patches as another important factor while computing image saliency.

In the process, I consider the difference of the mean L∗a∗b∗color values between any two blocks, and the spatial distance between them. This is represented by the denominator on the previous equation where d(Ri , Rj) is the spatial distance between the patches Ri and Rj of the image, and x and y refers to the coordinate of the current patch :

$$d(R_i, R_j) = \sqrt{(\bar{x}_{R_i} - \bar{x}_{R_j})^2 + (\bar{y}_{R_i} - \bar{y}_{R_j})^2}$$

A saliency map can be considered to be a probability map, with the intensity of a pixel indicating its chance of belonging to the tumor region in the original image. While a saliency value 0 (pure black, when considered as a gray scale image) indicates least importance, an intensity of 1 (pure white) corresponds to highest importance. Although the saliency map for a larger patch can help in accurately locating a salient object, its resultant blurring causes disappearance of most details.

Saliency maps, depicting the saliency strength at every pixel over different scales, involving varying sizes of the patches (with k = 4, 8, 16, 32, of sizes 4×4, 8 ×8, 16 ×16 and 32 ×32) are provided in Figure 4 (post-rescaling to their original sizes).

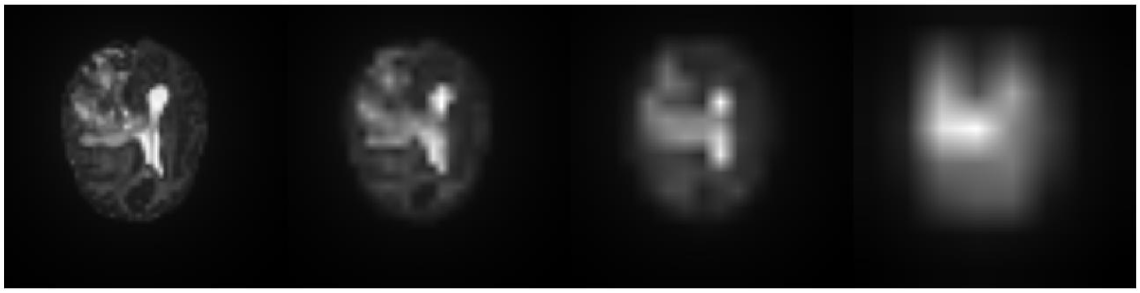

Figure 4: Saliency map of pseudo-colored MR image at different scales. Patch size of (1) 4 x 4, (2) 8 x 8, (3) 16 x 16, (4) 32x32.

We observe that as the contour of the ROI gets gradually blurred, with increasing patch size k, the position of the salient region becomes clearer. Here the block size k relates to the resolution of the saliency map.

Re-scaling is performed to bring back the saliency maps to the original image size (M × N) using Bilinear interpolation. Let $\hat{S}^k$ denote the interpolated image at its original size, as generated from the saliency map S^k at scale k.

Since the properties of a region depend on the pixels within it, saliency prediction is governed by both its size and scale. Thus this algorithm is simultaneously employed over multiple scales, for capturing the salient region(s) in the MR image at different levels of resolution. Those region(s) consistently highlighted over different resolutions are deemed to be the ones most likely to be salient.

Therefore we superimpose these saliency maps, corresponding to the different scales, for computing the final map. For example, the integrated map over the four scales of Figure 4 contains all important information and is depicted in Figure 5.

The final saliency map is now computed as:

$$S = \sum_{k=4,8,16,32} r^k \times \hat{S}^k$$

Where r^k is the weight corresponding to the saliency map at size k. In order to make each map give an equal contribution, in this project I have chosen:

$$r^k = {}^1/_4 \qquad \forall\, k$$

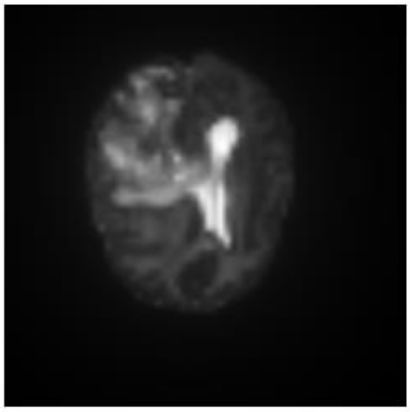

Figure 5: Final saliency map - Superimposed saliency map corresponding to Figure 4

Finally a 25 ×25 mean filter is applied to smoothen the saliency map S, in order to help focus on the core region within the actual ROI in the resized image. This is depicted in Fig. 6 and acts as the reference map for subsequent segmentation.

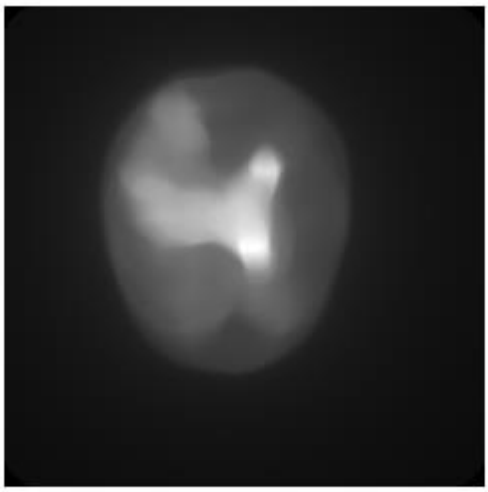

Figure 6: Final saliency map - smoothened version.

Modification for 3D saliency map

Generation of saliency maps is next extended over the 3D stack of such 2D MR image slices, each encompassing the three MR sequences.

However for the slices having too small foreground, as compared to the background, this saliency detection algorithm can produce false positive results.

This is because saliency is computed based on the mean color difference between patches, thereby highlighting the small foreground regions in such 2D slices.

In order to circumvent this problem all background pixels, in each of the three sequences, are now replaced by their mean image intensity values.

I decided to calculate the average intensity value without considering the background pixels; therefore the sum of the intensities of the pixels is divided only by the number of pixels that have an intensity value other than zero, the results are shown in Figure 7.

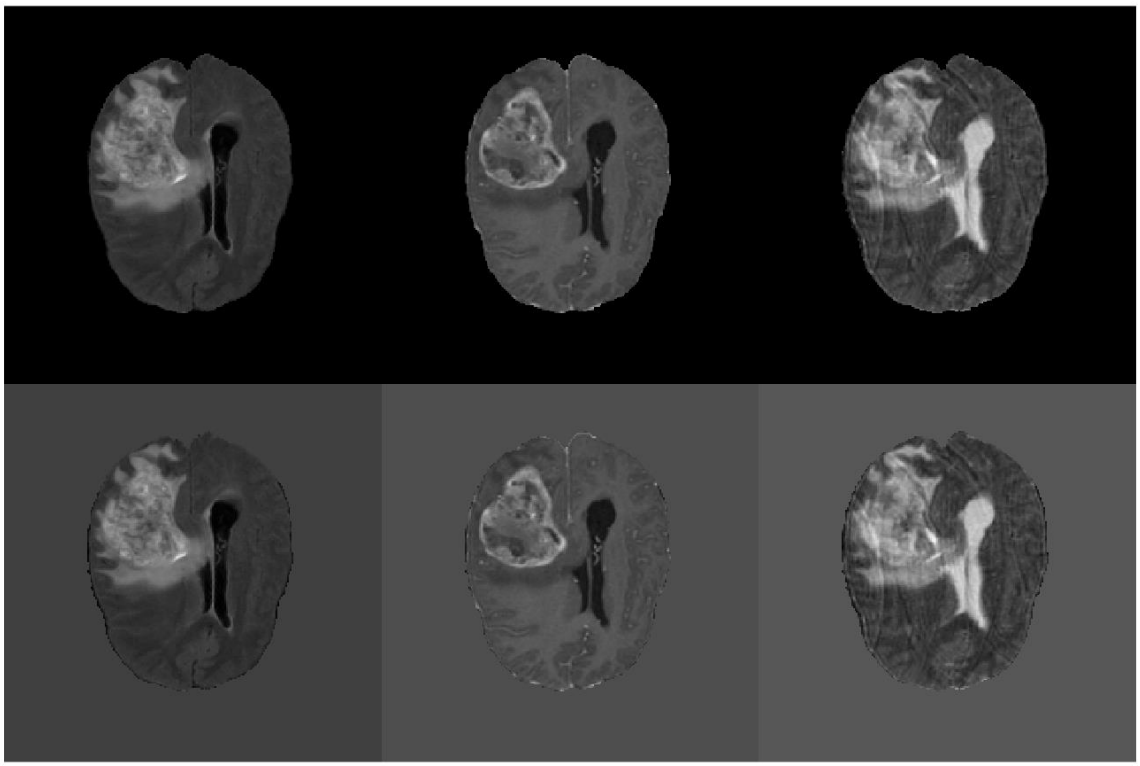

Figure 7: (1) First row MR slice (left to right): Flair, T1c, T2 ; (2) Second row: corresponding modified MR slices.

In this modified version the saliency detection algorithm effectively ignores such skewed Situations.

Main 3D Object detection

According to the object-based attention theory, human visual processing starts with the segmentation of an image into connected regions, which are also termed as proto-objects or pre-attentive objects .

Each proto-object is then assigned an importance value, which is called its saliency score. Saliency score is represented by the intensity of pixel in the saliency map obtained in the previous step.

Then the attention or focus reaches the most salient proto-object in the scene or image.

To identify the objects in the saliency map I used superpixel3 matlab function that implements the Simple Linear Iterative Clustering (SLIC) algorithm.

A superpixel can be defined as a group of pixels that share common characteristics, in this case, the SLIC algorithm generates superpixels by clustering pixels based on their color similarity (intensity distance) and proximity in the image (spatial distance) defined as following:

$$d_{intensity} = \sqrt{(l_i - lj)^2}$$

$$d_{spatial} = \sqrt{(x_i - xj)^2 + (y_i - yj)^2 + (z_i - zj)^2}$$

$$D = \sqrt{\left(\frac{d_{intensity}}{m}\right)^2 + \left(\frac{d_{spatial}}{S}\right)^2}$$

Where l_i represent the intensity value of the pixel i and l_j the mean intensity value of cluster j; x y and z are the coordinates of the pixel i and the coordinates of the center of the cluster j.

m is the compactness parameter and it determines the relative importance of the intensity distance and the spatial distance in the overall distance metric. A lower value makes the superpixels adhere to boundaries better, making them irregularly shaped. A higher value makes the superpixels more regularly shaped. S is the maximum spatial distance S = √(N/K)

The SLIC algorithm takes as input a desired number of approximately equally-sized superpixels K.

K initial cluster centers are sampled on a regular grid spaced S pixels apart each ones having an area of N/K where N is the number of pixels in the input image. The centers are moved to seed locations corresponding to the lowest gradient position in a 3x3x3 neighborhood.

Next, in the assignment step, each pixel i is associated with the nearest cluster center, using di distance D specified above, whose search region overlaps its location. Since the expected spatial extent of a superpixel is a region of approximate size S × S x S, the search for similar pixels is done in a region 2S × 2S x 2S around the superpixel center.

Once each pixel has been associated to the nearest cluster center, an update step adjusts the cluster centers to be the mean of all the pixels belonging to the cluster.

The L2 norm is used to compute a residual error E between the new cluster center locations and previous cluster center locations. The assignment and update steps can be repeated iteratively until the error converges when a maximum number of iterations are performed.

The result of this algorithm is shown in Figure 8, on the left side there is a single slice of the saliency map and on the right side there is the result of applying superpixel on that slice.

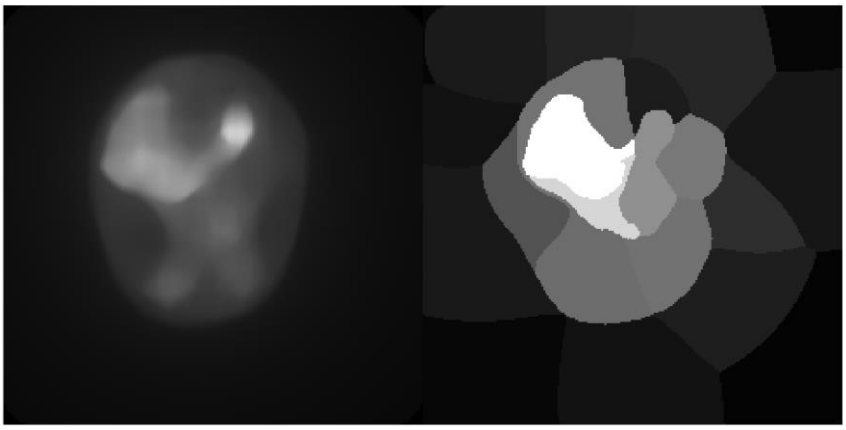

Figure 8: superpixel's results

Then, only the object corresponding to the maximum intensity value is extracted and it corresponds to the main proto-object.

Morphological operations

The main proto-object is used for generating foreground and background seed region on the MR images.

The foreground seed region is computed by shrinking the size of the proto-object using morphological operation, with a sphere structuring element.

Next the proto-object is enlarged by applying the morphological dilation operation, with a circular structuring element using a pair of radii r1 < r2 . Let dilation using r 1 and r 2 result in enlarged proto-objects ‘α’ and ‘β’ respectively. The pixels within the region β−α and outside β are selected as the background pixels.

Now the marked foreground and background pixels serve as the 3D label map of the image, for subsequent region growing based segmentation.

We want to label the pixels within the region between the foreground pixels and α, as shown in grey color in figure 9.

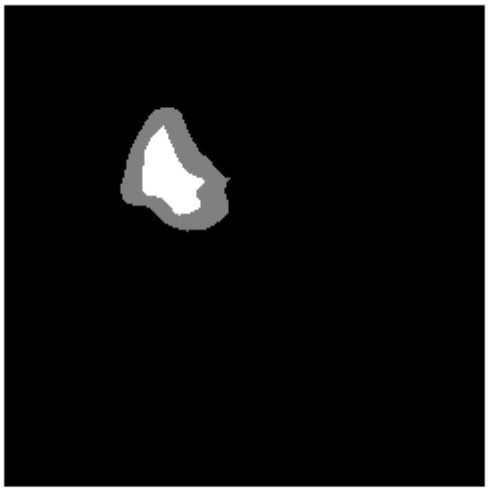

Figure 9: Foreground pixels (white), background pixels (black), pixels to label (grey)

Automated segmentation

Finally the 3D label map is used for automated segmentation of the tumor region.

I choose to use the Grow-Cut (GC) algorithm which is reported to be good for tumor delineation in medical images. Grow-Cut typically requires the user to insert seed points, both on the object and the background regions (constituting the user-specified label map), based on which the algorithm iteratively expands the ROI following the principles of cellular automata (CA). My algorithm, on the other hand, automatically generates the 3D label map based on saliency for subsequent Grow-Cut segmentation.

Since the label map entails no user interaction, it is termed the unsupervised or automated version of Grow-Cut (AGC).

Cellular automaton is generally an algorithm discrete in both space and time, that operates on a lattice of sites p 2 P µ Zn (pixels or voxels in image processing). A cellular automaton is a triplet

A = (S;N;f), where S is an non-empty state set, N is the neighborhood system, and f is the state transition 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 previous time step t.

Here the state of each cell is given by a 3-tuple (l, θ, I), where l is the label, θ∈ [0, 1] is its strength and I is the pixel intensity. The 3D label map obtained from the morphological operations is used as the initial label map.

The strength of all foreground pixels are assigned to cell strength 1, while the background pixels are set to 0. This defines the initial state of the cellular automaton. At iteration t + 1 cell label l t+1 p and strength θt+1 p are updated by the following algorithm:

AGC state update

Procedure Update(l^t, θ^t)
// for each cell
For $p \in P$ **do**
    // copy previous state
    $l_p^{t+1} = l_p^t$;
    $\theta_p^{t+1} = \theta_p^t$;
    // neighbors try to attack current cell
    For $q \in N(p)$ **do**
    If $g(||I_p - I_q||_2) \times \theta_q^t > \theta_p^t$ ***then***
        $l_p^{t+1} = l_q^t$;
        $\theta_p^{t+1} = g(||I_p - I_q||_2) \times \theta_q^t$;
    end if
    end for
end for
end procedure

Here g is a monotonically decreasing function, bounded by [0, 1], and it is defined as follows:

$$g(x) = 1 - \frac{x}{\max(I)}$$

The result of this phase is shown in the figure 10.

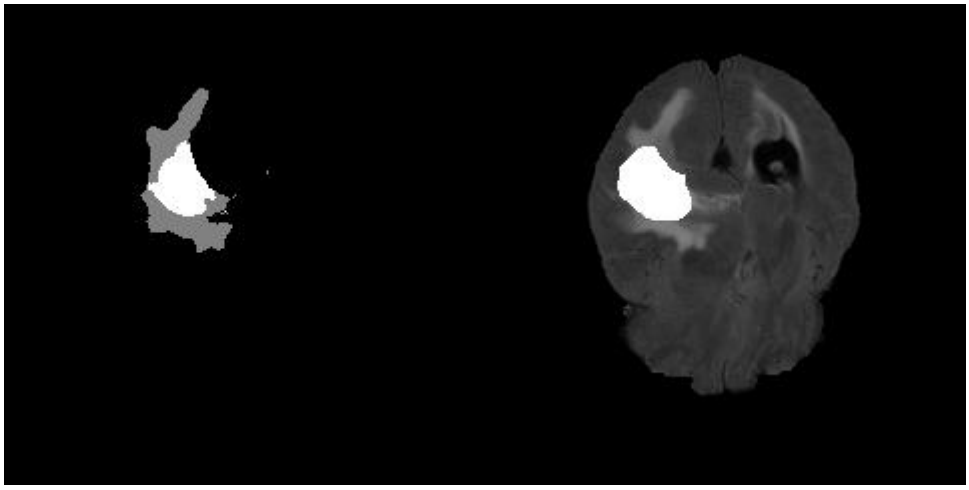

Figure 10: Tumor expert delineation (left side), tumor obtained from AGC (right side)