

An application of the supervoxel-based Fuzzy C-Means with a GPU support to segmentation of volumetric brain images

Anna Fabijańska

Lodz University of Technology
Institute of Applied Computer Science
ul. Stefanowskiego 18/22
90-924 Lodz, Poland
Email: anna.fabijanska@p.lodz.pl

Jarosław Goćłowski

Lodz University of Technology
Institute of Applied Computer Science
ul. Stefanowskiego 18/22
90-924 Lodz, Poland
Email: jaroslaw.goclawski@p.lodz.pl

Abstract—In this paper the problem of segmentation of volumetric medical images is considered. The fast and effective segmentation is obtained by applying the proposed approach which combines the idea of supervoxels and the Fuzzy C-Means algorithm. In particular, Fuzzy C-Means is used to cluster supervoxels produced by the fast 3D region growing. Additional acceleration of the method is achieved with the support of graphical processor (GPU). The detailed description of the proposed approach is given. The results of applying the method to volumetric CT and MRI brain images and CT images of various phantoms are presented, analysed and discussed. The issues related to accuracy of the method, memory workload and the running time are also considered.

I. INTRODUCTION

ONE of the main challenges of recent medical image processing is the development of 3D image segmentation algorithms. These algorithms should be fast, efficient and accurate. Additionally, they should be easy to use and thus diminish the amount of user interaction required to extract the region of interest.

Although the problem of 3D image segmentation have been widely considered and numerous dedicated segmentation approaches have been proposed (e.g. [1], [2], [3]), it is still far from the satisfactory solution. This is caused mainly by the constant increase of the resolution of volumetric images acquired by computed tomography (CT) and magnetic resonance imaging (MRI) scanners. This manifests itself both, by the increase of the spatial resolution of single slices as well as the number of slices included into a scan. This in turn translates into the significant increase of the time and the memory workload required to perform segmentation of volumetric medical data.

Because of these reasons the existing approaches to image segmentation often cannot be directly used in everyday clinical routine. Therefore, recently a lot of effort have been put into the optimization and adaptation of popular segmentation approaches to fast and efficient processing of high resolution 3D medical images. This problem is also considered in this paper where Fuzzy C-Means (FCM) algorithm [4], [5] is adapted

to segmentation of three dimensional images of brain. This is obtained by processing so called supervoxels (i.e. blocks of connected voxels of similar intensity) instead of single voxels. In particular, the input image is firstly divided into a number of supervoxels which are next clustered using FCM approach. This kind of processing significantly reduces the memory workload required to perform image segmentation.

It should be also mentioned, that supervoxels used in this paper extend the idea of superpixels, which has been known for few last years. However, the existing approaches to image division into blocks of pixels of similar intensity (e.g. watersheds [6], mean-shift [7], SLIC superpixels [8], Turbopixels [9] or FH superpixels [10]) are mainly dedicated to 2D images and their extension into the third dimension is not explicit or significantly increases time and memory workload and thus reduces benefit obtained due to processing blocks of pixels instead of single pixels. The method incorporated in this paper for creation of supervoxels is simple and straightforward. It is based on the fast and efficient region growing and thus can be directly adapted to three dimensional images [11].

Additionally, the graphical processor (GPU) support is also proposed to diminish the running time of both image division into supervoxels and FCM segmentation. What is more, the proposed approach reduces a user interaction to minimum, since only indication of one point is required to select the region of interest.

The following part of this paper is organised as follows. Firstly, in Section II the proposed approach is described in details. This is followed in Section III by the presentation and discussion of the results provided by the introduced method. These include both: the test performed on the CT and MRI brain scans, as well as the tests performed on various phantoms. Finally, Section IV concludes the paper.

II. THE PROPOSED APPROACH

A. Building supervoxels in the image space

The main idea behind the introduced approach is to use Fuzzy C-Means algorithm to cluster supervoxels produced by the fast region growing [11], [12]. The supervoxels divide an image into blocks of similar intensity. Each of the supervoxels is built starting from a randomly selected seed voxel still not assigned to any region. The growing is performed with regard to the allowable difference in pixel intensities ΔI_{MAX} and the maximum size V_{MAX} limiting the supervoxel size.

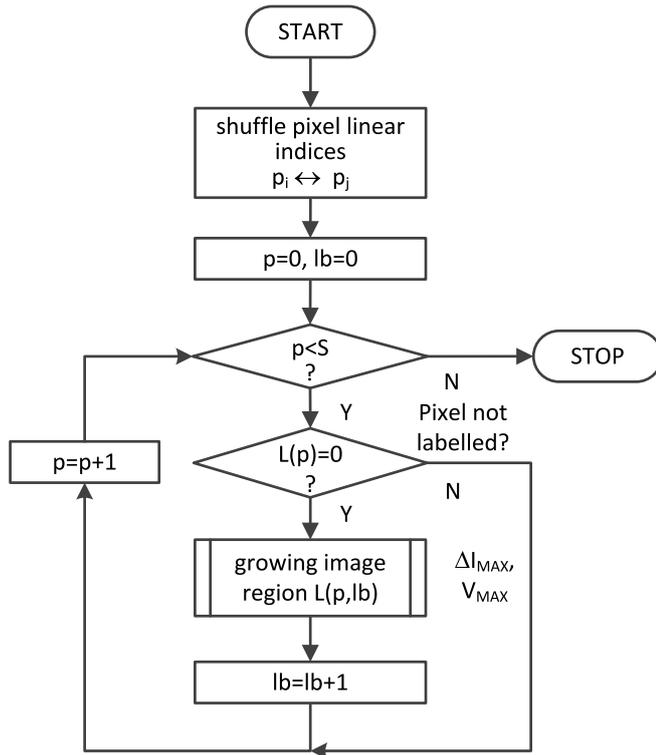


Fig. 1. Algorithm creating the label image L of three dimensional regions from the input intensity image I , $L(p)$ - the label assigned to the voxel $p \in [0, S)$, lb - the currently used label number.

The algorithm of image division into supervoxels shown in Figure 1 initially shuffles the voxel linear indices p of the input image I in the range $[0, S)$, where S is the number of voxels in the image. It speeds up the selection of unlabelled seeds. The indices randomly swapped in pairs can later be sequentially searched providing random selection of region seeds at the pixels which remain unlabelled. The procedure of growing image region around the seed p is shown in the Algorithm 1. It uses the queue Q_S of the next seed pixels and the queue of seed candidates Q_C . The queue Q_S is initially loaded with the primary seed pixel p , selected as shown in Figure 1.

For each pixel from Q_S the queue Q_C is created from its nearest 6 neighbours, assuming that their intensities fit in the range $I(p) \mp \Delta I$. Labelling the Q_C voxels increases the cumulated region volume V_R , explicitly limited to the value V_{MAX} . Exceeding volume limit V_{MAX} and empty voxel

queue Q_S stops the region labelling process. Supervoxels are assigned label numbers as consecutive non negative integers. The supervoxel intensities are computed after the labelling process as the means of all original voxel intensities in the region.

Algorithm 1 The algorithm of limited region growing around the randomly selected non-labelled image pixel p

Input: $I, lb, V_{MAX}, \Delta I_{MAX}$
Output: L

```

1:  $Q_S \leftarrow p, Q_C \leftarrow \emptyset$ 
2:  $V_R \leftarrow 0$ 
3: while  $Q_S \neq \emptyset$  do
4:    $p \leftarrow Q_S$ 
5:   foreach  $q \in N_B(p)$  do
6:     if  $\Delta I < \Delta I_{MAX}$  then
7:        $L(q) \leftarrow lb$ 
8:        $Q_C \leftarrow q$ 
9:        $V_R = V_R + 1$ 
10:    end if
11:    if  $V_R \geq V_{MAX}$  then
12:      return
13:    end if
14:  end foreach
15:   $Q_S \leftarrow Q_C$ 
16: end while
  
```

The results of image division into supervoxels are shown in Figure 2, where different colours represent different supervoxels. In particular, Figure 2a presents a sample CT brain slice, while the remaining subfigures show the corresponding slice after CT volume division into supervoxels of the increasing size V_{MAX} . All the results were obtained for the constant allowable difference in voxels intensity ΔI_{MAX} equal to 20. Due to the limited number of colours, they repeat for different supervoxels.

B. Fuzzy C-means specific solution

The segmentation of CT or MRI images with Fuzzy C-means (FCM) method [4] allows to assign any voxel with the intensity $v_i, i \in [1, N_V]$ to a certain post-segmentation class (region) with the membership degree $u_{ij}, j \in [1, N_C]$. The final deterministic assignment selects the class of the highest membership degree (probability) for each voxel v_i . FCM segmentation can be understood as an optimisation method minimizing the objective function G_m given in Equation (1).

$$G_m = \sum_{i=1}^{N_V} \sum_{j=1}^{N_C} u_{ij}^m \|v_i - c_j\|^2, \quad (1)$$

where $m > 1$, N_V is the image vector size, N_C is the given number of clusters (regions) of different intensities, u_{ij} is the membership degree of the voxel v_i to the region j , c_j denotes the intensity of j -th cluster centre and $\|\cdot\|$ represents the distance between the voxel v_i and the cluster centre c_j .

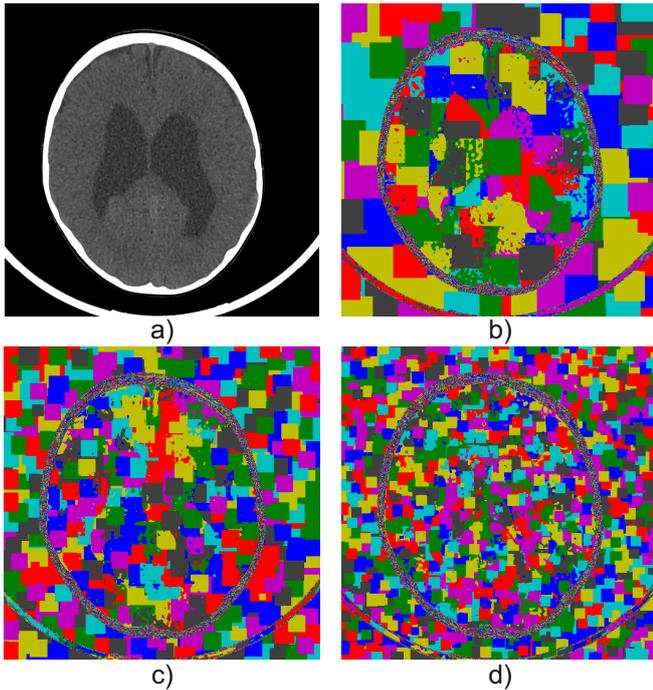


Fig. 2. The result of image division into supervoxels shown on a sample CT brain slice; a) original slice; b) $\Delta I_{MAX} = 20$, $V_{MAX} = 5000$; c) $\Delta I_{MAX} = 20$, $V_{MAX} = 10000$; d) $\Delta I_{MAX} = 20$, $V_{MAX} = 100000$.

The execution of the FCM algorithm relies on the interactive correction of the class centres c_j based on the memberships u_{ij} . Next iteration class centres imply further modifications of u_{ij} until stopping u_{ij} changes or achieving the limit of iterations. Assuming the array of class centres $C = [c_j]$, $U = [u_{ij}]$ - the array of membership degrees and $m = 2$ the following computations are applied iteratively:

$$c_j^{(k)} = \frac{\sum_{i=1}^{N_V} u_{ij}^2 v_i}{\sum_{i=1}^{N_V} u_{ij}^2}, \quad (2)$$

$$u_{ij}^{(k)} = \left(\sum_{k=1}^{N_C} \frac{\|v_i - c_j\|^2}{\|v_i - c_k\|^2} \right)^{-1}, \quad (3)$$

The iteration stops when $\|U^{(k)} - U^{(k-1)}\|_\infty < e_{MAX}$. The number of intensity classes N_C must be defined a priori.

In the case of 3D CT or MRI images including tens of millions of voxels the FCM clustering process may be unacceptable slow. Therefore the authors proposed its acceleration in two ways:

- by using smaller number of supervoxels,
- by applying parallel computations with GPU.

The pseudocode of FCM algorithm with CUDA GPU computing [13], [14] is presented in Algorithm 2. The command *par foreach* built inside means a *for* loop executed in parallel by GPU processors. Capital letter symbols in the code refer to host data buffers, bold font of capital letters denotes arrays

Algorithm 2 FCM segmentation algorithm using GPU parallel computing. N_V –the number of supervoxels, N_B –the number of GPU memory blocks, N_t –the number of GPU threads per block, N_C –the number of classes, b –the current block number, t –the thread number in a block, `sync()` –the thread synchronization.

Input: $V[N_V]$, N_C , e_{MAX} , k_{MAX}

Output: $C[N_C]$, $U[N_V \times N_C]$

```

1:  $\mathbf{V} \leftarrow V$ 
2:  $\mathbf{U}_- \leftarrow \text{random}([N_V \times N_C])$ 
3:  $\mathbf{U}[N_V \times N_C] \leftarrow \{0\}$ 
4: alloc  $\mathbf{P}[N_B \times N_C]$ ,  $\mathbf{Q}[N_B \times N_C]$ 
5: alloc  $\mathbf{C}[N_C]$ 
6:  $k \leftarrow 0$ 

7: repeat
8:   swap_addresses( $\mathbf{U}$ ,  $\mathbf{U}_-$ )
9:    $\triangleright$  kernel function #1
10:  par foreach  $i \in [0, N_V]$  do
11:    alloc  $\mathbb{S}[N_t \times N_C]$ 
12:     $\forall j \in [0, N_C)$ ,  $\mathbb{S}(t, j) \leftarrow \mathbf{U}(i, j)^2 \cdot \mathbf{V}(i)$ 
13:    sync()
14:     $\forall j \in [0, N_C)$ ,  $\mathbf{P}(b, j) \leftarrow \text{reduction}(\mathbb{S}(t, j))$ 
15:     $\forall j \in [0, N_C)$ ,  $\mathbb{S}(t, j) \leftarrow \mathbf{U}(i, j)^2$ 
16:    sync()
17:     $\forall j \in [0, N_C)$ ,  $\mathbf{Q}(b, j) \leftarrow \text{reduction}(\mathbb{S}(t, j))$ 
18:  end foreach
19:   $\triangleright$  kernel function #2
20:  par foreach  $j \in [0, N_C)$  do
21:     $\mathbf{C}(j) \leftarrow \frac{\sum_{b \in [0, N_B)} (\mathbf{P}(b, j))}{\sum_{b \in [0, N_B)} (\mathbf{Q}(b, j))}$ 
22:  end foreach
23:   $\triangleright$  kernel function #3
24:  par foreach  $i \in [0, N_V)$  do
25:     $v \leftarrow \mathbf{V}(i)$ 
26:    foreach  $j \in [0, N_C)$  do
27:       $w \leftarrow \|v - \mathbf{C}(j)\|^2$ 
28:       $s \leftarrow 0$ 
29:       $\forall k \in [0, N_C)$   $s \leftarrow s + \|v - \mathbf{C}(k)\|^{-2}$ 
30:       $\mathbf{U}(i, j) \leftarrow 1/(w \cdot s)$ 
31:    end foreach
32:  end foreach
33:   $\triangleright$  kernel function #4
34:   $e \leftarrow \|\mathbf{U} - \mathbf{U}_-\|_\infty$ 
35:   $k \leftarrow k + 1$ 
36: until ( $e < e_{MAX}$ )  $\vee$  ( $k \geq k_{MAX}$ )
37:  $\mathbf{C} \leftarrow \mathbf{C}$ 
38:  $\mathbf{U} \leftarrow \mathbf{U}$ 

```

allocated in the GPU memory, double stroked font symbols represent GPU shared memory arrays.

The algorithm input data are: the supervoxel vector image $V[N_V]$, the assumed number of intensity classes N_C , the norm e_{MAX} of maximum acceptable error and k_{MAX} – the maximum number of iterations for Equation (2) and Equation (3). The output data consists of the host vector $C[N_C]$ of output class centres and the host array of each supervoxel membership degree $U[N_V \times N_C]$. The \mathbf{U} array is at first allocated in the graphic card memory and randomly initialized in the probability range of $[0, 1]$. The GPU memory also includes local temporal arrays \mathbf{P} and \mathbf{Q} to store component parts of the numerator and denominator in Equation (2) corresponding to N_B blocks, each of the size $N_t = 256$ GPU threads.

The kernel function #1 for each thread computes the monomials in the numerator of Equation (2) and copies them to the GPU shared memory \mathbb{S} to later evaluate partial sums \mathbf{P} and \mathbf{Q} by the process of reduction.

The kernel function #2 adds the partial sums and computes for each thread the values c_j given in Equation (2).

The kernel function #3 completely fulfils the formula in Equation (3), because it sums the relatively small number N_C of intensity classes.

The segmentation error e in a current computing cycle is evaluated as the maximum distance norm of supervoxel membership degrees in two successive iterations. The error value is copied to the host memory to make the decision of stopping iterations.

The array of membership degrees is allocated in the GPU memory in two copies \mathbf{U} and \mathbf{U}_- , which addresses are swapped instead of copying data between \mathbf{U} and \mathbf{U}_- in every iteration cycle (Algorithm 2, line 8).

The FCM output array U includes N_C column images of the degrees of membership to a particular class. The class of the highest membership in each row of U is assigned to the output vector of intensity supervoxel classes as in Equation (4).

$$V(i) = \max_j(U(i, j)), \quad i \in [0, N_V), \quad j \in [0, N_C). \quad (4)$$

This output vector image $V[N_V]$ is then reshaped to the original matrix form $I[Y \times X \times Z]$ after its reverse mapping from supervoxels to voxels. In the image I of N_C intensity classes (labels) only a single region is identified, which belongs to a certain class determined by the voxel marker that was selected interactively. The identification can be fulfilled by the flood fill spatial expansion covering the whole region around the marked voxel.

The complete algorithm sequence of brain image segmentation starts with GPU averaging lowpass filter to reduce data noise coming from the CT or MRI acquisition systems. The noise is gained in particular when setting low doses of radiation during brain examinations in children. The filter fulfils the formula given in Equation (5).

$$J(x, y, z) = \frac{1}{UVW} \sum_{u=-\frac{U}{2}}^{\frac{U}{2}} \sum_{v=-\frac{V}{2}}^{\frac{V}{2}} \sum_{w=-\frac{W}{2}}^{\frac{W}{2}} I(x+u, y+v, z+w), \quad (5)$$

where $[V \times U \times W]$ – the cube of image data averaging with the odd numbers U, V, W . The segmentation is finalized with the operation of morphological opening (Equation (6)).

$$J_B = (I_B \ominus S(R_X, R_Y, R_Z)) \oplus S(R_X, R_Y, R_Z), \quad (6)$$

where $S(R_X, R_Y, R_Z)$ denotes an ellipsoidal structuring element with the radii R_X, R_Y, R_Z in the particular space directions, I_B and J_B are the input and output binary images of a selected region in the original brain image. The ellipsoidal structuring element S represents an ellipsoid mask mapped into the discrete space of image voxels with the radii R_X, R_Y, R_Z respectively in X, Y, Z space directions. Using the ellipsoid instead of the ball shape enables mapping different image resolutions in space directions into the voxel space (in particular, the spacing between slices). Post processing morphological operations allow controlled smoothing of the borders in the extracted brain region.

III. RESULTS

A. Tests on brain images

The results of applying the proposed segmentation approach to sample volumetric CT and MRI brain images are shown in Figures 3 and 4 respectively. In both figures the top panel shows the original brain slices with the region of interest indicated by the green square marker. In the middle panel the corresponding segmentation results overlaid on the input slice are presented. Finally, the segmentation results are visualised in 3D in the bottom panel. In both figures, cases are numbered from the left to the right. In the case of CT images the image division into supervoxels was performed for the maximal volume of supervoxel equal $V_{MAX} = 7500$ and maximal intensity difference $\Delta I_{MAX} = 40$. For FCM segmentation the number of classes was set to $N_C = 6$, while maximal number of iterations k_{MAX} was set to 400 (with $e_{MAX} = 0.001$). In the case of MRI images image division into supervoxels was performed for $V_{MAX} = 5000$ and $\Delta I_{MAX} = 40$. The number of classes N_C ranged from 8 (for case 1) to 6 (for the remaining cases). As previously, the maximal number of iterations k_{MAX} was set to 400 (with $e_{MAX} = 0.001$). In both cases the parameters were tuned manually, to obtain the subjectively best results. Prior to segmentation all images were subjected to preprocessing (Gaussian filtration).

The time and memory workload required to perform image segmentation in the considered CT and MRI datasets is summarised in Table I and Table II respectively. In both tables the first column indicates the case ID. This is followed by image resolution given in the second column. The third

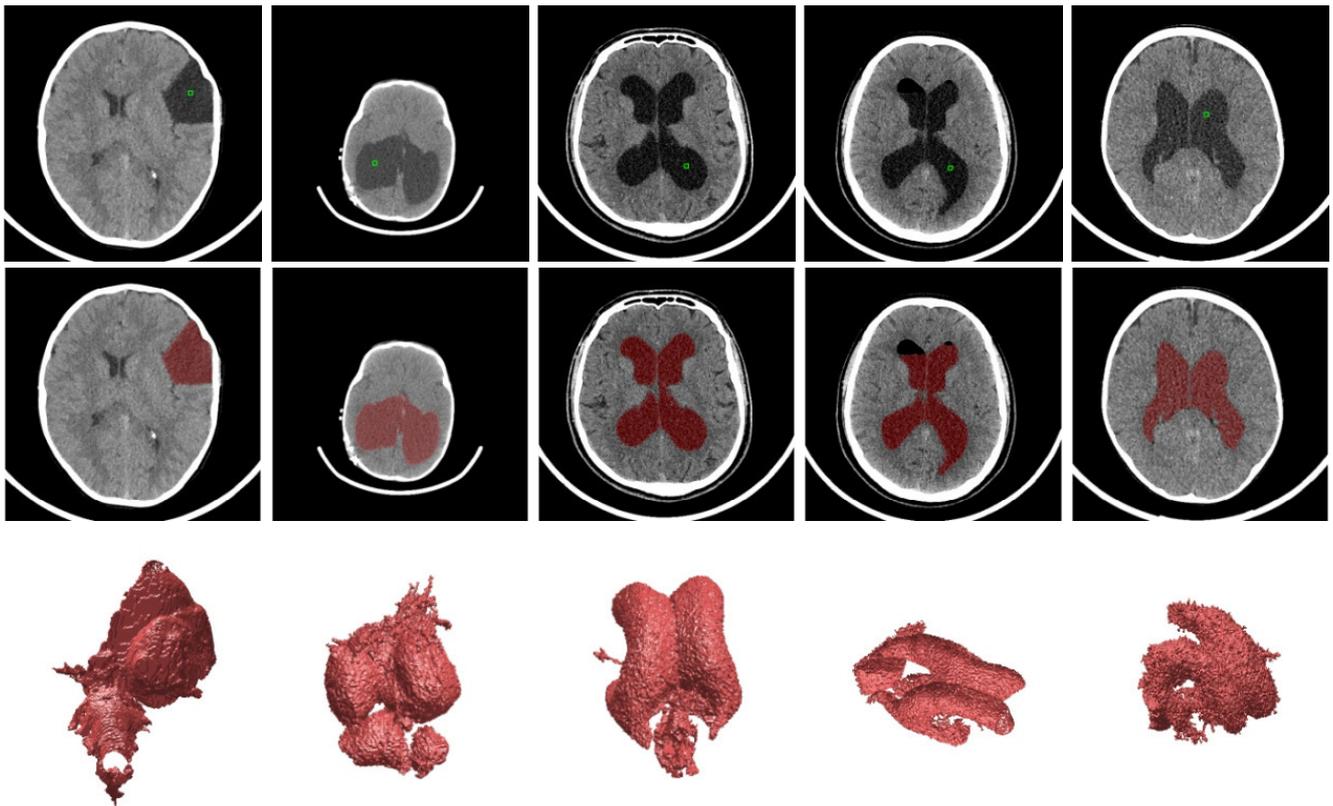


Fig. 3. The results of applying the proposed approach to sample CT brain images; top row - original images with a region of interest indicated by green square marker; middle row - the results overlaid on a sample slices; bottom row - the results shown in 3D. Cases are numbered from left to right.

column shows the level of data reduction r due to image division into supervoxels. In particular, it means that the number of supervoxels was $r\%$ lower than the number of voxels. The fourth column shows the GPU memory workload given in MB, while columns five, six and seven present time T_0 of preprocessing (filtration), time T_1 of image division into supervoxels and time T_2 of FCM execution respectively. All times are given in milliseconds. The tests were performed on a computer with Intel Core i7 3.6 GHz processor, 32 GB RAM memory and graphical card Nvidia GeForce Titan (6 GB).

TABLE I
THE TIME AND MEMORY WORKLOAD OF THE PROPOSED METHOD - THE RESULTS FOR CT DATASETS.

ID	Image size [px.]	r [%]	GPU [MB]	T0 [ms]	T1 [ms]	T2 [ms]
1	512×512×115	95.02	140.9	426.2	5410.2	560.5
2	512×512×112	96.59	80.4	435.3	9313.5	288.8
3	512×512×216	97.03	314.0	548.9	15366.1	818.9
4	512×512×220	97.24	293.9	538.8	18518.3	1087.7
5	512×512×202	97.34	344.4	544.6	17022.1	1103.9

From the Figures 3 and 4 it can be seen that the proposed

TABLE II
THE TIME AND MEMORY WORKLOAD OF THE PROPOSED METHOD - THE RESULTS FOR MRI DATASETS.

ID	Image size [px.]	r [%]	GPU [MB]	T0 [ms]	T1 [ms]	T2 [ms]
1	512×448×25	88.15	88.6	0	1750.5	446.5
2	512×512×22	88.97	83.3	311.3	1326	406.4
3	512×512×21	86.23	75.5	323.8	1405.4	403.5
4	256×256×23	88.54	19.0	306.7	340.2	101.9
5	256×256×20	87.10	18.5	0	360.7	89.1

approach was successful in segmenting indicated regions of interest both CT and MRI images. Based on visual assessment it can be concluded that the borders of ventricular fluid and cysts were properly determined and details of its shape were captured by the image segmentation algorithm. The accuracy of object shape determination is sufficient for further quantitative analysis.

In the case of CT images it was possible to run the algorithm with a uniform setting of parameters, especially the number of classes N_C considered during FCM segmentation. In the case of MRI images, to obtain better quality results it was necessary

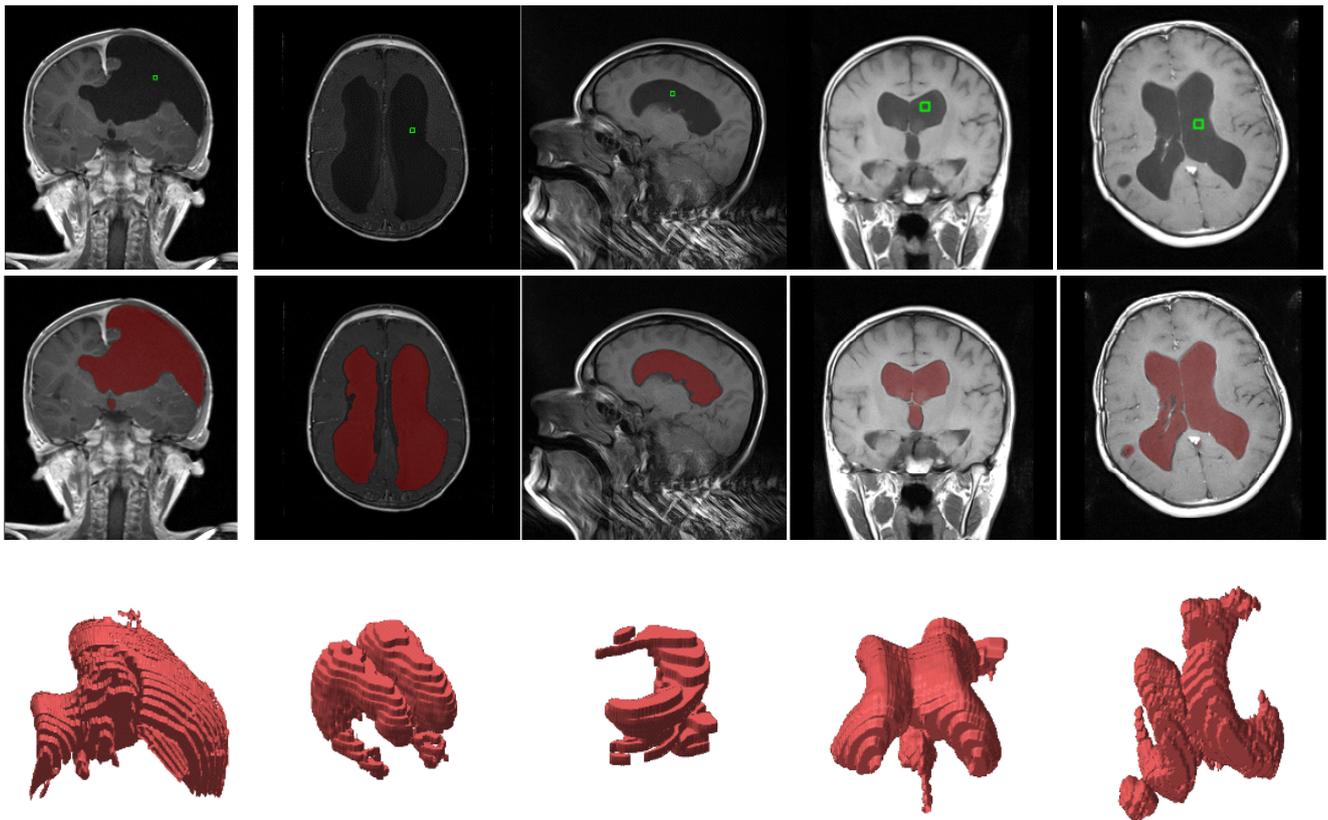


Fig. 4. The results of applying the proposed approach to sample MRI brain images; top row - original images with a region of interest indicated by green square; middle row - the results overlaid on a sample slices; bottom row - the results shown in 3D. Cases are numbered from left to right.

to tune the number of classes in the case of one considered dataset. Additionally, in cases 1 and 5 preprocessing was not applied since it deteriorated segmentation results.

From the Tables I and II it can be seen, that the proposed approach is fast and efficient. For all the considered cases the running time of the method wasn't longer than 20 seconds including preprocessing, image division into supervoxels and FCM segmentation. This is a very good result, regarding that the considered datasets consisted of up to 220 slices of resolution 512×512 pixels each. The corresponding time of FCM segmentation performed with respect to single voxels last hours. This acceleration is obtained due to significant reduction of clustered data. In the considered cases, processing of supervoxels instead of voxels diminished the number of data points clustered by FCM by on average 96.6% for CT images and 88.7% in the case of MRI images.

The memory workload of the proposed approach was also significantly reduced due to application of supervoxels. When run on the original datasets the FCM required several gigabytes of RAM memory to cluster all the voxels. As it can be seen from Tables I and II the improved method in the worst case used less than 350 MB of GPU RAM memory. In the case of volumetric images this is a very good results which makes

the method available for a common use on a standard PC computer.

TABLE III
THE INFLUENCE OF SUPERVOXELS AND GPU COMPUTING ON THE FCM SEGMENTATION TIME IN CT DATASETS. T_2 - THE EXECUTION TIME WITH SUPERVOXELS AND GPU, T_2' - NO SUPERVOXELS, T_2'' - NO SUPERVOXELS OR GPU.

ID	Image size [px.]	T_2 [ms]	T_2' [ms]	T_2'' [ms]
1	$512 \times 512 \times 115$	560.5	9594	186873
2	$512 \times 512 \times 112$	288.8	7301	147686
3	$512 \times 512 \times 216$	818.9	21934	540873
4	$512 \times 512 \times 220$	1087.7	19562	595971
5	$512 \times 512 \times 202$	1103.9	24196	552163

In tables III and IV different variants of FCM segmentation times are compared for CT and MRI images respectively. The column T_2 refers to the proposed method applying both supervoxels and parallel GPU computing. Two next columns describe the same cases omitting supervoxel creation (T_2') and also GPU computing (T_2''). The exclusion of supervoxels increases CT segmentation time $17 \div 27$ times (on average

TABLE IV
THE INFLUENCE OF SUPERVOXELS AND GPU COMPUTING ON THE FCM SEGMENTATION TIME IN MRI DATASETS. T2 - THE EXECUTION TIME WITH SUPERVOXELS AND GPU, T2' - NO SUPERVOXELS, T2'' - NO SUPERVOXELS OR GPU.

ID	Image size [px.]	T2 [ms]	T2' [ms]	T2'' [ms]
1	512×448×25	446.5	9813	84817
2	512×512×22	406.4	11576	90933
3	512×512×21	403.5	9126	119840
4	256×256×23	101.9	2652	26286
5	256×256×20	89.1	3198	28595

≈ 22 times) and additional removing of GPU computing extends it up to 330 ÷ 660 times (on average ≈ 510 times). For the MRI images the FCM execution is slowed down on average 27 and 258 times in the two algorithm variants with T2' and T2''. The acceleration with supervoxels is achieved at the expense of their preparation time. Therefore real speed-up of the algorithm in this approach is only 1.2 and 28 times for the tested CT images or respectively 6 and 56 times for MRI cases. The example CT images of larger size than the example MRI require more effort to prepare supervoxels. Hence, their use only gives less profit on time, which can be still enhanced during subsequent parallel computations.

B. Tests on phantoms

The accuracy of the introduced supervoxels based FCM segmentation approach was assessed using three different physical phantoms including: phantom of head, phantom CIRS 045 of prostate [15] and phantom CIRS 062 [16] used for calibration of computed tomography scanners. All these phantoms were scanned using a CT scanner. The resulting images were next subjected to segmentation using the proposed approach. The aim of the segmentation was to extract characteristic objects contained within phantoms. Finally, the volumes of these objects were determined based on segmentation results and compared with the volumes given in phantom specification.

The phantom of head was prepared using the 3D printing technique. The phantom consists of two main parts: the skull made from a material of the density 2.1-2.3 g/cm³ and the brain made from a material of the density 1.02-1.04 g/cm³. In the brain there is a hole of a shape similar to ventricular system and volume equal to 41.69 cm³. The hole was filled with water imitating the cerebrospinal fluid. The head phantom is presented in Figure 5. In particular, Figure 5a shows the general view of the phantom, while Figure 5b presents a sample CT slice. Additionally, in Figure 6 the shape of the ventricular system of the phantom is presented.

The phantom CIRS 045 is dedicated to prostate brachytherapy. Inside it contains three cysts of the increasing volumes (namely: 4 cm³, 9 cm³ and 20 cm³). The general view of the phantom is shown in Figure 7a, while a sample CT scan is presented in Figure 7b.

Finally, the phantom CIRS 062 is shown in Figure 8. The phantom consists of two rings with inclusions of equal size.

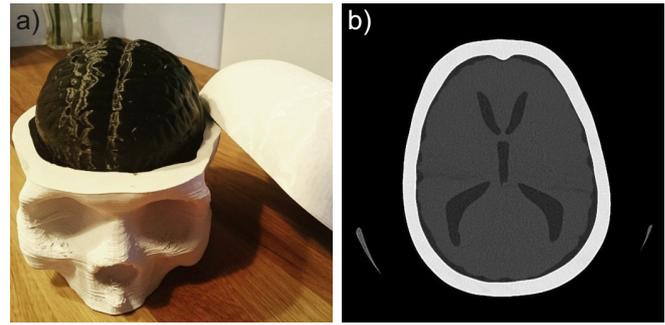


Fig. 5. The brain phantom; a) the general view; b) a CT scan.

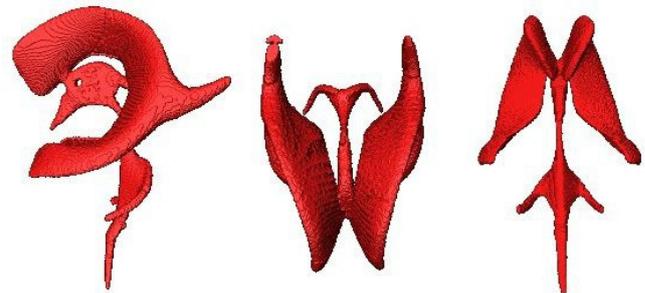


Fig. 6. The shape of the ventricular system within the brain phantom.

Each inclusion corresponds with different tissue and thus exhibit different attenuation of X-rays (see Fig. 8b).

The results of applying the introduced segmentation approach to images of phantoms are shown in Figure 9. In the case of head phantom, the proposed approach was used to extract the ventricular system (see Fig. 9a). From prostate phantom CIRS 045 the largest cyst was extracted (see Fig. 9b). Finally, from the phantom CIRS 062 a randomly selected inclusion was extracted using the proposed approach (see Fig. 9c). For each case, the segmentation result is both: shown in 3D and overlaid on a sample slice.

The results of comparison between the real and the determined volumes of the considered regions are summarised in Table V. In particular, the considered phantom is indicated in the first column. The real region volume V₀ is given in the fifth column, and followed by the determined volume V in the sixth column. The corresponding relative error of volume

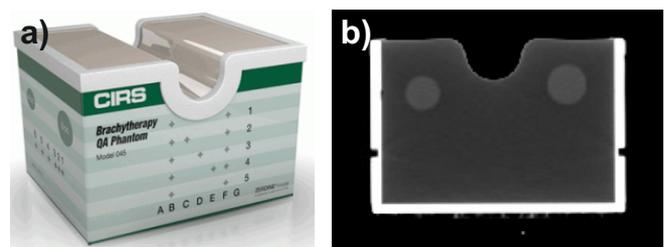


Fig. 7. Phantom CIRS 045; a) the general view (©CIRS Tissue Simulation & Phantom Technology); b) a sample slice from a CT scan.

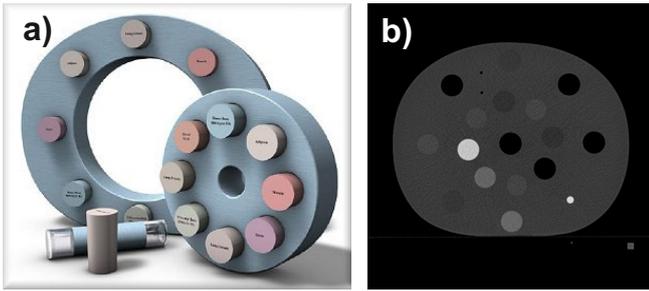


Fig. 8. Phantom CIRS 062; a) the general view (©CIRS Tissue Simulation & Phantom Technology); b) a sample slice from CT scan.

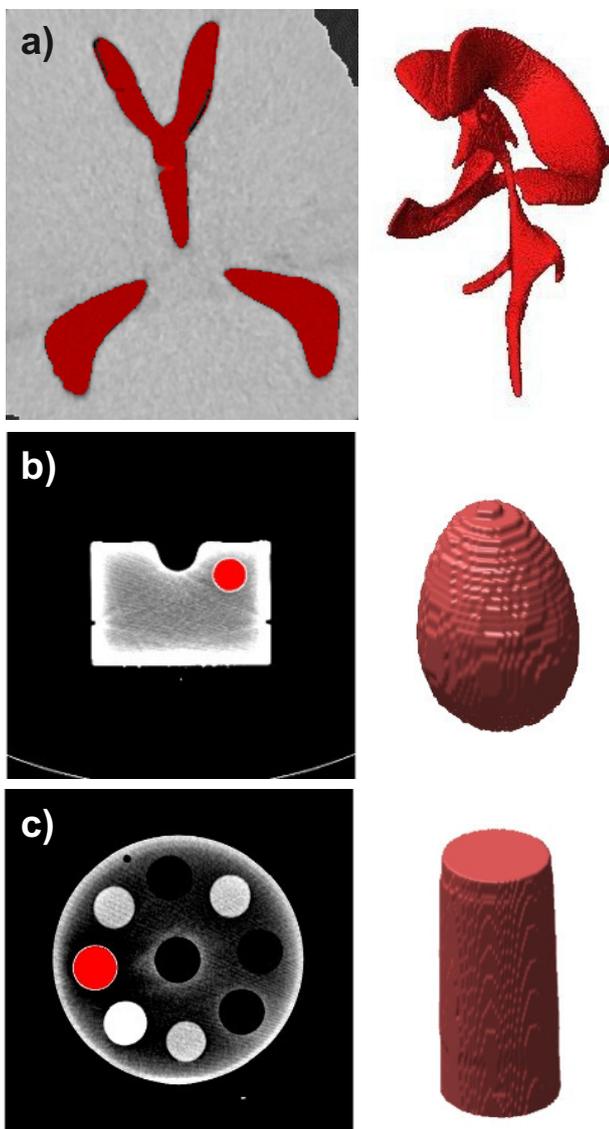


Fig. 9. The results of supervoxel based FCM segmentation applied to the considered phantoms; a) brain phantom; b) CIRS 045; c) CIRS 062.

determination is shown in the last column. Additionally, for each phantom image resolution, time of image division into supervoxels T_0 and time of FCM segmentation T are given in columns two, three and four respectively.

TABLE V
THE ASSESSMENT OF SEGMENTATION ERROR.

phantom	Image size [px.]	T_0 [s]	T [s]	V_0 [cm ³]	V [cm ³]	dV [%]
head	512×512×515	36.88	1.66	41.69	40.77	-2.20
CIRS 045	512×512×61	3.85	0.42	20.00	17.44	-12.80
CIRS 062	512×512×55	3.46	0.41	42.44	53.75	1.41

From Figure 9 it can be easily seen that the proposed segmentation approach successfully extracted objects of interest from the considered phantoms. The shape of the segmented objects correspond with the real one. This is especially visible in the case of the ventricular system segmented from the phantom of head (see Fig. 9a). In this case all important anatomical details are clearly visible in object after segmentation. The shape of objects extracted from the CIRS phantoms also correspond with the description given in phantoms specification.

The relative error of volume determination equals on average 5.5% (see Tab. V). It is worth highlighting, that in the case of the ventricular system extracted from the phantom of head the error is only -2.2%. This is a very good result, especially having in mind the complex shape of the ventricular object.

IV. CONCLUSIONS

In this paper the fuzzy C-means method (FCM) was specifically adapted to the segmentation of volumetric brain images. In particular CT images include from tens to hundreds of millions of voxels to analyse within a reasonable time. Although the FCM method has the advantage of the same functionality in two and three dimensions, the algorithm execution time is proportional to the number of image voxels and can exceed large values for hundreds of CT slices. The proposed reduction of input data size based on supervoxels varies from 85% to 95% in the tested examples at the expense of several seconds for creating supervoxel regions. For the assumed parameters the segmentation accuracy of about 2% tested for the head phantom still remains acceptable. The above data compression and GPU parallel computing limit processing time to single seconds, practically without any extra cost for hardware equipment. Standard CUDA compatible NVIDIA graphic card of computing capability 3 ÷ 3.5% will be sufficient in this case, because the GPU memory workload falls below 0.5 GB for 200 CT slices. The bottle neck of such graphic computing is the maximum acceptable size of shared memory per block (up to 8 kB) implying the limited class number to preserve the parallel computation efficiency. Additionally in some medical cases establishing a priori the number of segmentation classes to extract desired objects is a very intuitive task. Nevertheless the proposed approach make the modified FCM method very promising for its very fast processing of bulk data and dimensionality independence.

ACKNOWLEDGEMENTS

The authors would like to thank Mikołaj Kopernik's Hospital in Lodz (Poland) for providing CT images of Electron Density Phantom CIRS 045 and Brachytherapy Phantom CIRS 062.

REFERENCES

- [1] D. D. Burdescu, L. Stanescu, M. Brezovan, C. S. Spahiu, "Efficient Volumetric Segmentation Method", *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, pp. 659–668, 2014, <http://dx.doi.org/10.15439/978-83-60810-58-3>
- [2] K. Xiao, A. E. Hassaniien, N. I. Ghali, "Medical Image Segmentation Using Information Extracted from Deformation", *Proceedings of the 2011 Federated Conference on Computer Science and Information Systems*, pp. 157–163, 2014.
- [3] M. R. Ogiela, T. Hachaj, "Automatic segmentation of the carotid artery bifurcation region with a region-growing approach", *Journal of Electronic Imaging*, vol. 22(3), 033029, 2013, <http://dx.doi.org/10.1117/1.JEI.22.3.033029>
- [4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [5] R. Nock, and F. Nielsen, "On weighting clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(8), pp. 1–13, 2006, <http://dx.doi.org/10.1109/TPAMI.2006.168>
- [6] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33(7), pp. 1384–1399, 2011, <http://dx.doi.org/10.1109/TPAMI.2010.200>
- [7] W. Tao, H. Jin, and Y. Zhang, "Image segmentation based on mean shift and normalized cuts", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37(5), pp. 1382–1389, 2007, <http://dx.doi.org/10.1109/TSMCB.2007.902249>
- [8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state of the art superpixel methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(11), pp. 2274–2282, 2012, <http://dx.doi.org/10.1109/TPAMI.2012.120>
- [9] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31(12), pp. 2290–2297, 2009, <http://dx.doi.org/10.1109/TPAMI.2009.96>
- [10] P. F. Felzenszwalb, and D. P. Huttenlocher, "Efficient graph-based image segmentation" *International Journal of Computer Vision*, vol. 59(2), pp. 167–181, 2004, <http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77>
- [11] A. Fabijańska, and J. Goławski, "The segmentation of 3D images using the random walking technique on a randomly created image adjacency graph" *IEEE Transactions on Image Processing*, vol. 24(2), pp. 524–537, 2015, <http://dx.doi.org/10.1109/TIP.2014.2383323>
- [12] W. Pratt, *Digital Image Processing*, 4th ed. Los Altos, California: John Wiley & Sons Inc., 2007.
- [13] J. Sanders and E. Kandrot, *Cuda by Example. An Introduction to General Purpose GPU programming*, NVIDIA Corporation, 2011.
- [14] N. Wilt, *The CUDA Handbook. A Comprehensive guide to GPU Programming*, Addison-Wesley, Upper Saddle River, NJ, 2013.
- [15] Brachytherapy QA Phantom CIRS 045, <http://www.cirsinc.com/products/all/71/brachytherapy-qa-phantom>
- [16] Electron Density Phantom CIRS 062, <http://www.cirsinc.com/products/all/24/electron-density-phantom>