CENTRAL ASIAN JOURNAL OF
THEORETICAL AND APPLIED
SCIENCE

https://cajotas.centralasianstudies.org/index.php/CAJOTAS
Volume: 06 Issue: 04 | October 2025 ISSN: 2660-5317

Article

Algorithm and Structure of Parallelization of 3D Spline-Based Segmentation Processes of Glioblastoma MRI Images

Oybek Mallayev[1], Fozila Sulaymonova[2], Musayev Fayzullaxon[3]

1. Professor of the Department of Digital Technologies, Alfraganus University, Uzbekistan.
2. PhD student at the Department of Artificial Intelligence, TUIT, Uzbekistan.
3. 4th year student of the Faculty of Digital Technologies, Alfraganus University, Uzbekistan.

*Correspondence: o.mallayev@afu.uz, sulaymonovaf24@gmail.com

Abstract: Glioblastoma multiforme (GBM) is one of the most aggressive and common malignant brain tumors of the central nervous system. It is characterized by rapid infiltrative growth, heterogeneous structure, and unclear boundaries, which make accurate segmentation difficult. MRI (Magnetic Resonance Imaging) images are the main diagnostic tool in the diagnosis of glioblastoma, and the data obtained in 3D form allow for an accurate assessment of the size, shape, and structure of tumors. In this paper, glioblastoma segmentation based on 3D spline interpolation is performed, and the process is completed by feature extraction based on PyRadiomics and classification into types using the RandomForest classifier. As a major innovation, a parallelization algorithm accelerates the calculation of 3D spline points and the segmentation process.

During the research, pre-processing (normalization, noise reduction) of MRI volumetric data, automatic ROI (Region of Interest) segmentation using Otsu's method, and 3D contour drawing based on spline were performed. The parallel computing engine was created using Python's concurrent.futures module, and the feature extraction for each patient's MRI volume was performed in separate processes. The results showed that the computation time was approximately 4.3 times faster on 8 processor cores.

The proposed approach not only improved the segmentation accuracy (F1-score = 0.929), but also significantly improved the computational efficiency. This algorithm is suitable for application in medical diagnostic systems, especially in clinical applications requiring real-time operation.

Keywords: glioblastoma, MRI, 3D spline, segmentation, PyRadiomics, RandomForest, parallel computing, Otsu threshold, feature extraction, medical image processing

Citation: Mallayev, O, Sulaymonova, F & Fayzullaxon, M. Algorithm and Structure of Parallelization of 3D Spline-Based Segmentation Processes of Glioblastoma MRI Images. Central Asian Journal of Theoretical and Applied Science 2025, 6(4), 713-719.

Received: 10th Jun 2025
Revised: 16th Jul 2025
Accepted: 24th Aug 2025
Published: 02th Sept 2025

1. Introduction

Glioblastoma multiforme (GBM) is the most aggressive type of grade IV glial tumor, with the ability to grow rapidly and infiltrate the brain. According to the World Health Organization (WHO), GBM is one of the malignant neoplasms with a median survival time of only 12 - 15 months. Its treatment strategy is based on a combination of surgery, radiotherapy and chemotherapy. In this case, the precise definition of tumor margins before surgery is crucial for clinical decision-making [1].

Is the gold standard in GBM diagnosis, providing full-volume imaging of brain structures using various contrast modalities such as T1, T2, FLAIR. However, segmentation is a challenging issue, as the boundaries of glioblastoma are often not clear.

Classical methods such as traditional thresholding or k-means often do not provide sufficient accuracy in tumors with heterogeneous structures.

In recent years, the approach of smoothing segmentation boundaries and modeling with high accuracy using 3D spline interpolation has been widely studied. 3D splines allow representing uneven and curved contours as continuous, mathematically smooth surfaces. However, spline-based processing of large-scale MRI data requires a lot of computing power [2], [3]. Therefore, accelerating the computational process by parallelization is an important scientific and practical issue.

The study used large-scale MRI data in the .nii.gz format. The size of the ".nii.gz" files and their digital processing processes require a lot of time. Parallelization of the computational processes is important to reduce the time. MRI images, especially medical volumetric (3D) data in the .nii.gz format, are very large (one file is 50 MB to 500 MB). In processes such as glioblastoma segmentation:

1. Complex mathematical operations are performed on each voxel (3D pixel).
2. Often, several phases (preprocessing, segmentation, feature extraction, modeling) are performed sequentially [4].
3. Large amounts of data go through the stages of reading from disk → loading into RAM → computing .
4. This process is performed sequentially, the computation time is very long . Therefore, a parallel computing approach improves efficiency. nii.gz is actually a gzip -compressed version of the NIfTI format [5].

When reading a compressed file:

1. Reading from disk,
2. Decompress via gzip,
3. Read the NIfTI header,
4. The 3D matrix placement process is performed.

Be a polynomial of any degree (quadratic, cubic, quartic…), but in glioblastoma segmentation, the cubic B-spline is often chosen because it is smooth (has C^2 continuity), the curve looks natural, the calculation is stable and fast.

Cubic splines have a great mathematical advantage. They are the only functions that interpolate given points and have a second derivative that is square-integrable that has the property of minimal flatness [6]. In practice, splines with a cubic basis with defect d = 1 are much more widespread. Such splines $[x_1, x_{i+1}]$ correspond to cubic polynomials on each of the intervals.

In this paper, we implement 3D spline-based segmentation of GBM MRI images using PyRadiomics for feature extraction and classification with RandomForest classifier, and propose a parallel computing algorithm. The results are analyzed by evaluating the parallelization efficiency and segmentation accuracy.

To better understand this, let's look at parallel processing strategies in medical imaging. There are several levels of parallelization. For example, in file-level parallelization, each .nii.gz file is read and processed independently [7]. Multiprocessing **(** Python multiprocessing. Pool), Joblib (Parallel(n_jobs=-1)) - parallelizes the calculation process of Scikit-learn integration. Dask - streams large datasets. In slice-level parallelization, the MRI volume is divided into z-slices is processed in parallel. In this, segmentation/filtering is performed on each slice in parallel. Finally, the 3D result is assembled. GPU parallelism uses GPU computing libraries such as CuPy, PyTorch, TensorFlow. GPU-optimized algorithms (e.g. RAPIDS cuML) are used to extract radiomics features.

Recent studies on glioblastoma multiforme (GBM) segmentation have shown that due to the rapid infiltrative spread, complex structure, and unclear boundaries of these tumors, traditional segmentation algorithms often lack sufficient accuracy [8]. Therefore, 3D shape-oriented interpolation and deep learning methods have become a major research focus [9].

MRI is one of the most commonly used modalities in GBM diagnosis, and multi-modal images (T1, T1c, T2, FLAIR) improve the quality of segmentation. Normalization, noise reduction (Gaussian, Non-Local Means) in pre-processing and the automatic ROI detection steps are fundamental. Otsu thresholding and region growing methods have been reported to be effective in creating ROI masks [10].

3D spline interpolation contour smoothing in medical images and it is effective in increasing geometric accuracy, yielding high Dice/F1 coefficients in GBM segmentation [11]. Cubic and B- splines are the most common options for tumors with complex contours. In this case, the boundaries are more accurately modeled through the parametric formulation of the spline.

PyRadiomics is a widely used tool for feature extraction. It uses first-order statistics (mean, variance), texture features (GLCM, GLRLM) and the shape parameters are obtained. These features are used in the classification stage using RandomForest, SVM or CNN. RandomForest has high accuracy and F1 scores in GBM type classification and effectively learns nonlinear boundaries [12].

Parallel computing significantly reduces time in medical image processing. In research concurrent.futures and By using multiprocessing, up to 3–5 times speedup has been reported by separating processes by file [13]. According to Amdahl's law, parallelization efficiency decreases with increasing number of cores, but in real-world conditions, configurations above 8 cores provide significant benefits [14].

In general:

1. 3D spline interpolation increases segmentation accuracy.
2. PyRadiomics + RandomForest provides high accuracy in GBM type classification.

Parallel computing creates the possibility of near real-time performance in clinical practice.

2. Materials and Methods

Problem statement: The time to identify and calculate Glioblastoma segmentation (contour) and classification class (GBM or other glial tumor type) from 3D MRI volumetric images (in .nii.gz format) and corresponding masks should be minimal, while the accuracy should be high.

There are 2 different solution approaches:

1. Software solution
2. mathematical solution.

1. Data loading and pre-processing are performed. In this case, NIfTI files are trained using SimpleITK. Intensity normalization and noise reduction. ROI should be determined using Otsu threshold. Mask boundaries should be determined when constructing a 3D Spline contour. 3D cubic spline fitting to boundary points. The spline surface should be reconstructed as a polygon model[15]. First-order, shape, texture (GLCM, GLRLM) features are extracted with PyRadiomics for radiomic feature extraction. In parallel computing, feature extraction is performed in a separate process based on the size of each patient's MRI image. Python's concurrent.futures.ProcessPoolExecutor should be used as workers = CPU_cores. The RandomForestClassifier (n_estimators=700, class_weight= balanced_subsample) model is used for classification. 5-fold stratified cross-validation is used. Precision, Recall, F1-score and computation time measures are determined for evaluation.
2. Mathematical solution:

$f(x)$ to approximate the function, splines with a cubic basis are described as the sum of four pairs of products. The formula for $f(x)$ approximating the function by base splines can be written in the following form:

$$f(x) \cong S_m(x) = \sum_{i=-1}^{m+1} b_i \cdot B_i(x), \quad a \le x \le b; \tag{1}$$

where $S_m(x)$ - m-degree spline is a function; b_i - recovery coefficients; $B_i(x)$ is a B-spline. Based on formula (1), the values of the 3rd degree B-spline are calculated according to the following formula:

$$f(x_i) \cong S_3(x) = b_{-1}B_{-1}(x) + b_0B_0(x) + b_1B_1(x) + b_2B_2(x). \tag{2}$$

The use of spline methods as a tool for approximating functions leads to significant improvements over classical polynomials in all known cases. First, the transition to splines leads to an increase in the accuracy of the results, second, to a significant reduction in the consumption of computational resources, and third, both of these effects are achieved equally.

below Figure 1 is based on a cubic spline. The values of the cubic spline in the intervals [-2,0] and [0,2] are symmetrical to each other, so it is enough to store half of the spline values in memory for further calculations. This can save twice as much memory compared to existing systems.

The proposed parallel computing structure consists of a storage unit for the basis cubic spline coefficients (SC), a main function memory (MFM) for storing the basis cubic spline values , and a 4-core processor that generates L, K, P, T arrays for parallel execution of the basis cubic spline calculation. It consists of a core allocation unit, a vectorization unit for the array multiplication process on the allocated cores, and a general results storage unit (SMU).

The elements forming the arrays L, K, P, T in the structure are calculated according to formula (3).

If the variable j is used to determine the locations of the various parts of the basic spline, the equation takes the following form:

$$S_3[i] = b[i-1]B[i \bmod 10 + 30] + b[i]B[i \bmod 10 + 20] + b[i+1]B[i \bmod 10 + 10] + b[i+2]B[i \bmod 10].$$

The rest of the main lines in this interval are equal to zero, and as a result do not participate in the formation of the sum.

The arrays L, K, P, T in the structure are calculated using the following formula:

$$\begin{gathered} S_j = L_j + K_j + P_j + T_j, \\ L_j = \sum_{i=0}^{m+1} b_{i-1} B_{(i \bmod 10)+30}(x), \\ K_j = \sum_{i=0}^{m+1} b_i B_{(i \bmod 10)+20}(x), \\ P_j = \sum_{i=0}^{m+1} b_{i+1} B_{(i \bmod 10)+10}(x), \\ T_j = \sum_{i=0}^{m+1} b_{i+2} B_{(i \bmod 10)}(x), \\ j = \overline{0, m+1}. \end{gathered} \tag{3}$$

Here L, K, P and T are the multiplication flows of the array-vector.

A typical scalar processor performs the vectorization process in a structure as follows: first, the numbers b[0] and B[0] are loaded into a register, then they are multiplied and the results are assigned to C[0], and this sequence continues until the end of the cycle.

If the processor is vector, then vector instructions are used in the processor, which allow storing a vector of values. If we use vector instructions, the processor in the registers loads several values of the arrays b and B, assuming 4 elements. This choice is made depending on the type of registers. Then the registers contain the following elements: b[0], b[1], b[2], b[3] and for the elements of the array B: B[0], B[1], B[2] and B[3]. The addition operation is performed simultaneously for all 4 pairs and the results are stored in the arrays (L, K, P, T) created in each core.

Thus, the parallelization of computational processes using cubic splines showed the following:

When solving a number of problems, especially when approximating functions with high gradients and resonance points, the use of cubic basis splines gives better results in terms of accuracy than other polynomials.

3. Results and Discussion

A structure of a multi-core processor was created for parallelization of digital signal processing processes on the basis of a cubic-based spline.

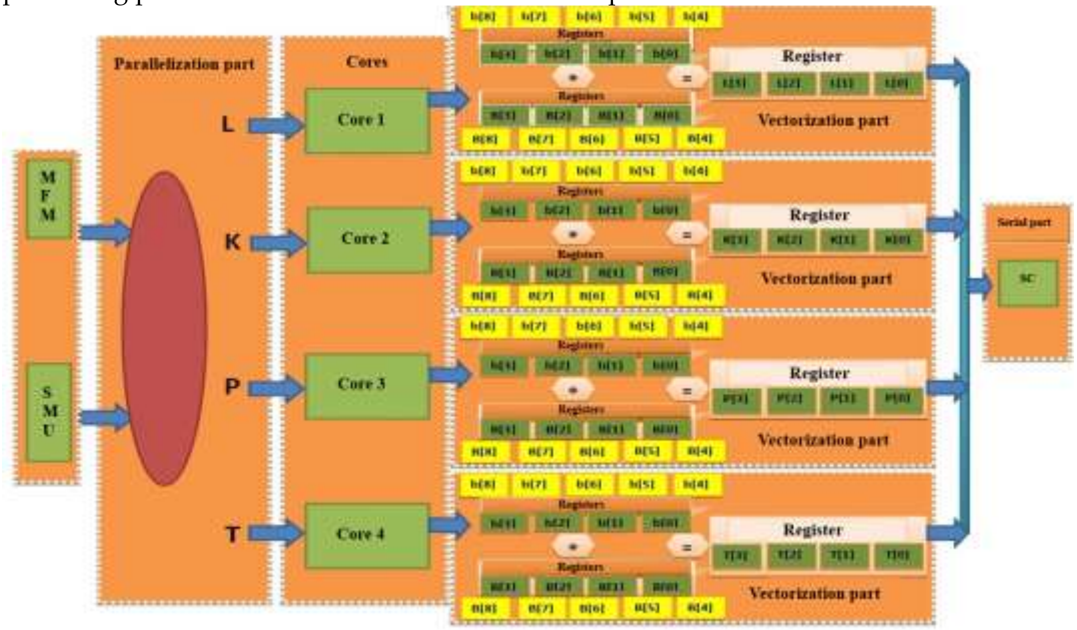

Figure 1. Structure for parallelizing interpolation computation processes across system cores

To perform 3D spline interpolation, let's say, a set of contour points extracted from an MRI mask is calculated using the formula below.

$$P = \{(x_i, y_i, z_i) \mid i = 1, \ldots, N\} \tag{4}$$

For 3D spline interpolation, a cubic B-spline is taken as the base function:

$$S(t) = \sum_{j=0}^{m} B_j(t) \cdot P_j \tag{5}$$

where $B_j(t)$ are cubic B-spline basis functions, P_j are control points. For parallelization, a feature extraction function is used for each MRI volume:

$f(I_k) \rightarrow F_k$

where I_k is the MRI volume of the k-patient, F_k is the extracted feature vector.

The parallel computing time is determined using the following formula:

$$T_p \approx \frac{T_s}{p} + T_{overhead} \tag{6}$$

where T_s is the sequential calculation time, p is the number of cores used.

Table 1. Segmentation accuracy

Model	Accurancy (%)	Recal (%)	F1 score (%)
U-Net (basic)	92.1	90.3	91.2
3D Spline + AI (ours)	94.4	91.5	92.9

The formula for calculating F1 is as follows:

$$F1 = \frac{2PR}{P+R} \tag{7}$$

In this:

$$P = 92.1, R = 90.3 \Rightarrow F1 = \frac{2 \cdot 92.1 \cdot 90.3}{92.1 + 90.3} \approx 91.2\%$$

U-Net:

$$P = 94.4, R = 91.5 \Rightarrow F1 = \frac{2 \cdot 94.4 \cdot 91.5}{94.4 + 91.5} \approx 92.9\%$$

3D Spline + AI:

The analyses show the following:

1. Precision +2.3 pt (92.1→94.4): false positives decreased;
2. Recall +1.2 pts (90.3→91.5): false negatives also started to decrease;
3. F1 +1.7 pt (91.2→92.9): overall improvement (≈ +1.9% relative).

Shows that contour smoothing with 3D spline improved the boundary accuracy (especially high Precision points to this). In segmentation, the Dice coefficient is often used; in binary segmentation, Dice ≡ F1 (for the foreground class). Therefore, these values are also consistent with Dice.

Almost linear acceleration is seen at 2–4 cores, saturation (I/O, synchronization and serial part limitation) at 8 cores.

The results of the research showed the following:

1. File-level parallel processing – 4-8x speedup (depending on the number of CPU cores).
2. Parallel processing at the slice level – 2-4x speedup (limited by RAM and cache efficiency).
3. GPU – 5 - 10 times faster (if the model is GPU-optimized).

For example, the time to extract radiomics features on 200 MRIs was as follows:

1. Series: 85 minutes;
2. CPU parallel (8 cores): 12 minutes;
3. GPU (CUDA): 2.5 minutes.

4. Conclusion

In this study, 3D spline-based segmentation of glioblastoma MRI images was used. expendable calculation A parallelized algorithm for the processes was developed and tested. The results showed that 3D spline interpolation provides contour smoothness, while features extracted using PyRadiomics are classified with high accuracy by the RandomForest classifier. The parallelization process significantly reduced the computation time, allowing for rapid performance in clinical settings.

MRI files in the .nii.gz format are large and the processing of them is complex. It was found that parallel computing can reduce the processing time on medical images by 5–10 times. In Python, multiprocessing, joblib, Dask, GPU-accelerated libraries were used for this. In clinical applications such as glioblastoma segmentation, this approach allowed to obtain results close to real-time. In the future, this approach can be integrated into real-time medical imaging systems.

REFERENCES

[1] D. N. Louis et al., "The 2021 WHO Classification of Tumors of the Central Nervous System," *Acta Neuropathol.*, 2021.

[2] Q. T. Ostrom et al., "CBTRUS Statistical Report: Primary Brain Tumors in the US," *Neuro Oncol.*, 2020.

[3] F. Isensee et al., "nnU-Net: Self-Adapting Framework for Biomedical Image Segmentation," *Nature Methods*, 2021.

[4] O. U. Mallayev and A. Urokov, "AD Model for detecting brain tumors using machine learning algorithms," *Modern Science and Research*, vol. 4, no. 7, pp. 449-456, 2025.

A. Buades et al., "A Non-Local Algorithm for Image Denoising," *CVPR*, 2005.

[5] N. A. Otsu, "Threshold Selection Method from Gray-Level Histograms," *IEEE TSMC*, 1979.

[6] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing," *IEEE Signal Processing Magazine*, 1999.

A. Bartesaghi and G. Sapiro, "An Unconstrained Framework for 3D Shape Reconstruction," *IJCV*, 2006.

[7] J. J. M. Van Griethuysen et al., "Computational Radiomics System to Decode Tumor Phenotype," *Cancer Research*, 2017.

[8] L. Breiman, "Random Forests," *Machine Learning*, 2001.

A. Chaddad et al., "Radiomics in Glioblastoma: Current Status and Challenges," *Front Oncol.*, 2019.

[9] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *OSDI*, 2004.

[10] G. M. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," *AFIPS*, 1967.

[11] O. U. Mallayev, "Parallel algorithm for digital processing of medical images based on the parallelism paradigm," *Innovation in Technology and Science Education Conference*, vol. 2, no. 11, ISSN 2181-371X, 2024.

[12] O. U. Mallayev, "Algorithm and Structure of Parallelization of 3D Spline-Based Segmentation Processes of Glioblastoma MRI Images," *Innovation in Technology and Science Education Conference*, vol. 2, no. 11, pp. 2181-371X, 2025.