International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 42, No. 2, April, 2025, pp. 719-727
Journal homepage: http://iieta.org/journals/ts

MATLAB Simulation, and FPGA Implementation of the DRLSE Segmentation Algorithm

Fatma Z. Hamadi[1*], Mohamed L. Hamidatou[1,2], Latifa Hamami-Mitiche[1], Sabrina Ait Belkacem[1], Bouchra Bouzid[1]

[1] Departement of Electronics, National Polytechnic School (ENP), Algiers 16000, Algeria
[2] Systems Design Methods Laboratory, National High College of Data Processing, Algiers 16000, Algeria

Corresponding Author Email: fatma.hamidatou@g.enp.edu.dz

https://doi.org/10.18280/ts.420210

Received: 1 August 2024
Revised: 12 December 2024
Accepted: 17 March 2025
Available online: 30 April 2025

Keywords:
DRLSE algorithm, FPGA implementation, level sets, MATLAB simulation, medical image, segmentation

ABSTRACT

This work focuses on using level set curves for medical image segmentation through the DRLSE (Distance Regularization Level Set Evolution) algorithm, recognized for its effectiveness and adaptability. Traditional systems face limitations in computation time and efficiency when implementing this algorithm. To overcome these challenges, FPGA (Field-Programmable Gate Arrays) are used for their parallelism and low resource consumption. The objective is to optimize medical image segmentation by implementing the DRLSE algorithm on FPGA while ensuring efficient resource and computation time management. The Algorithm was first simulated in MATLAB and tested on a database of brain, breast, and other medical images, demonstrating its robustness and flexibility. The results validate the effectiveness of the DRLSE algorithm and highlight the advantages of the FPGA in terms of speed and precision. Despite the limited documentation on implementing DRLSE on FPGA Our approach is distinguished by the use of DDR memory, which provides increased capacity to overcome the limitations of BRAM memory. Parameter optimization ensures better performance and efficient management of hardware resources. This work underscores the potential of FPGA-based implementations for accelerating computationally intensive tasks like medical image segmentation while maintaining high accuracy and efficiency.

1. INTRODUCTION

The level set method is an advanced numerical technique developed by Osher and Sethian in 1988 [1], based on the evolution of a curve by solving partial differential equations and using the concept of implicit functions, allowing it to efficiently handle topological changes. The level set method has applications in various fields, including computer vision, medical image segmentation, and many others.

The evolution of the function ϕ(x,y) follows Eq. (1) of the level set curves [2]:

$$\frac{\partial\emptyset(x,y)}{\partial t} + F.|\nabla\emptyset| = 0 \quad (1)$$

where, F is a scalar function representing the speed of the curve's motion, and |∇ϕ| is the magnitude of the gradient of ϕ ensuring evolution occurs normal to the level set.

This formulation (2) allows the function ϕ to propagate along the level curve by adjusting its signed distance while maintaining the regularity of the curve.

The scalar speed function F depends on the local properties of the curve F_{prop}, external parameters related to the image gradient F_{curv}, as well as additional terms for curve propagation F_{adv}.

$$F = F_{prop} + F_{curv} + F_{adv} \quad (2)$$

The propagation term F_{prop}: This term controls the global movement of the curve. For example, in the context of image processing, F_{prop} can be related to the image intensity gradient, causing the curve to move towards regions of high contrast.

The curvature term F_{curv} : This component acts as a regularization force, encouraging the smoothness of the curve. It depends on the curvature κ, which is defined as Eq. (3):

$$F_{curv} = -\varepsilon\, k \quad (3)$$

with: $k = \nabla.\left(\frac{\nabla\emptyset}{|\nabla\emptyset|}\right)$

where, ∇ denotes the divergence operator. The inclusion of this term helps reduce irregularities in the shape of the curve.

The advection term F_{adv} in Eq. (4): Advection can also help the curve move more efficiently by guiding its propagation towards regions of interest while avoiding unwanted areas. This improves the convergence of the segmentation model.

$$F_{adv} = \vec{U}\,(x,y,t).\vec{n} \quad (4)$$

The level set associated with the segmentation process can then be written as Eq. (5):

$$\frac{\partial\emptyset}{\partial t} + F_0|\nabla\emptyset| + \vec{U}\,(x,y,t).\nabla\emptyset = \varepsilon.k.|\nabla\emptyset| \quad (5)$$

The term $F_0|\nabla\emptyset|$ is influenced by the term $g|\nabla I|$ [3], which represents a decreasing function of the image gradient amplitude I. The most commonly used function $g|\nabla I|$ is given by Eq. (6):

$$g(|\nabla I|) = \frac{1}{1+|\nabla G_\sigma * I|^P} \tag{6}$$

where, p can be 1 or 2, and Gσ is a Gaussian kernel with standard deviation σ convolved with the image intensity I.
This function (6) aims to stop the evolution of the active contour when it reaches the boundaries of the target object corresponding to strong image gradients. The construction of the distance function ϕ is crucial in the level set method; it must satisfy the condition $|\nabla\phi|=1$ while preserving its configuration as a signed distance function from the front C(t) throughout its evolution [4].

To maintain the stability of the curve while preserving equidistance between its levels ($\nabla\phi=1$), a reinitialization of the distance function is performed every n iterations [5]. This is done by solving the specific Eq. (7):

$$\frac{\partial\emptyset}{\partial t} = sign(\emptyset_0)(1-|\nabla\emptyset|) \tag{7}$$

2. DRLSE APPROACH

Medical image processing and segmentation are essential for diagnosis and therapeutic planning. Traditional algorithms can be limited by their computational demands and real-time performance. Level sets can develop irregularities during their evolution, losing their form as a signed distance function, which can lead to numerical errors and compromise stability. In this context, some studies [6-8] proposed a new formulation of level sets, with the DRLSE algorithm introducing a distance regularization term that stabilizes contour evolution and accelerates segmentation, while inherently maintaining regularity and eliminating the need for reinitialization. It is widely used for image segmentation and contour detection in complex images, producing faster and more accurate results. Optimizations, such as the use of adaptive coefficients and dynamic adjustments of terms, further enhance the accuracy and speed of the results [9]. This new approach uses an energy function composed of three terms [7], as shown in Eq. (8):

$$E(\emptyset) = \mu\, P(\emptyset) + \lambda\, L_g(\emptyset) + \alpha\, A_g(\emptyset) \tag{8}$$

With:
μ: Distance regularization parameter.
λ: Contour length regularization parameter.
α: Surface repulsion parameter.
The Regularization Term in Eq. (9) ensures that the function retains its distance function form throughout the evolution [6].

$$P(\emptyset) = \int \frac{1}{2}(|\nabla\emptyset|-1)^2\, dx\, dy \tag{9}$$

The Length Term in Eq. (10) helps position the active contour close to the object boundary of interest [8].

$$L_g(\emptyset) = \int g(x,y)\delta_\varepsilon|\nabla\emptyset|dxdy \tag{10}$$

where, δϵ is the Dirac function, defined as Eq. (11):

$$\delta_\epsilon(x) = \begin{cases} 0 & |x| > \epsilon \\ \frac{1}{2\epsilon}\left[1+\cos\left(\frac{\pi x}{\epsilon}\right)\right] & |x| \le \epsilon \end{cases} \tag{11}$$

with ϵ as the regularization parameter, and g is a weight function that adjusts the impact of the length term based on the specific application or the region of interest, guiding the curve evolution effectively.

The Area Term in Eq. (12) accelerates the evolution speed of the active contour when it is far from the object of interest, facilitating faster convergence.

$$A_g(\emptyset) = -\int g(x,y)\, H(-\emptyset)dxdy. \tag{12}$$

The Heaviside function H(−ϕ(x,y)) selects the region outside the contour.

The DRLSE model [10] incorporates a distance regularization term to stabilize the contour evolution function, thus improving segmentation speed and eliminating the need to reinitialize the function. The evolution becomes as Eq. (13):

$$\frac{\partial\phi}{\partial t} = \mu \mathrm{div}\left(d_p(|\nabla\emptyset|)\nabla\emptyset\right) + \lambda\delta_\epsilon(\emptyset)\mathrm{div}\left(g\frac{\nabla\emptyset}{|\nabla\emptyset|}\right) + \alpha g\delta_\epsilon(\emptyset) \tag{13}$$

μ, λ, and α are constants that control the influence of each term.
d_p is a function given by the formula (14):

$$d_p(s) \cong \frac{p'(s)}{s} \tag{14}$$

where, p is a potential function used for distance regularization [8].

For better distance regularization, the double-well potential function p_2 is used, expressed as Eq. (15).

$$P_2 = \begin{cases} \left(\frac{1}{2\pi}\right)^2(1-\cos(2\pi s)), \text{si} & s \le 1 \\ \frac{1}{2}\ (s-1)^2, \quad \text{si} & s \ge 1 \end{cases} \tag{15}$$

3. DRLSE ALGORITHM

We present the DRLSE flowchart describing its operation in Figure 1.

Our DRLSE algorithm starts with an image preparation step to make it suitable for segmentation processing, including pre-processing and user-selected parameters. This is followed by an initialization step of the level set function (LSF) as a binary function, where a rectangular region represents the initial contour to evolve. It is also possible to initialize multiple contours simultaneously by defining several initial regions. A main loop is used to evolve the level set curve over a predefined number of outer iterations. At each iteration of the outer loop, the function drlse_edge updates the LSF using the equations of the Distance Regularized Level-Set Evolution model. These equations incorporate several energy terms:
- **Weighted Length Term**: Favors smooth contours by minimizing the curve length.
- **Distance Regularization Term**: Maintains the regularity of the LSF based on the Euclidean distance.
- **Weighted Area Term**: Helps segment objects by favoring interior regions over edges.

A well-chosen potential function plays a crucial role in Level-Set segmentation. For example, a single-well potential is effective for modeling homogeneous regions, where contours are less complex and a clear separation between regions is desired. In contrast, a double-well potential is suited for models that need to distinguish both edges and more uniform interior regions. It helps stabilize the level set curve around complex contours by offering a more diverse energy landscape that can capture both abrupt transitions and smoother regions.

Calculation of the Distance Regularization Term with a Double-Well Potential:

The MATLAB function distReg_p2(phi) demonstrates how the distance regularization term is calculated using a double-well potential. This calculation uses gradients of the level set function (phi) to estimate local distance and applies a suitable regularization function $d_p(s)$ that promotes stability in the segmentation process.

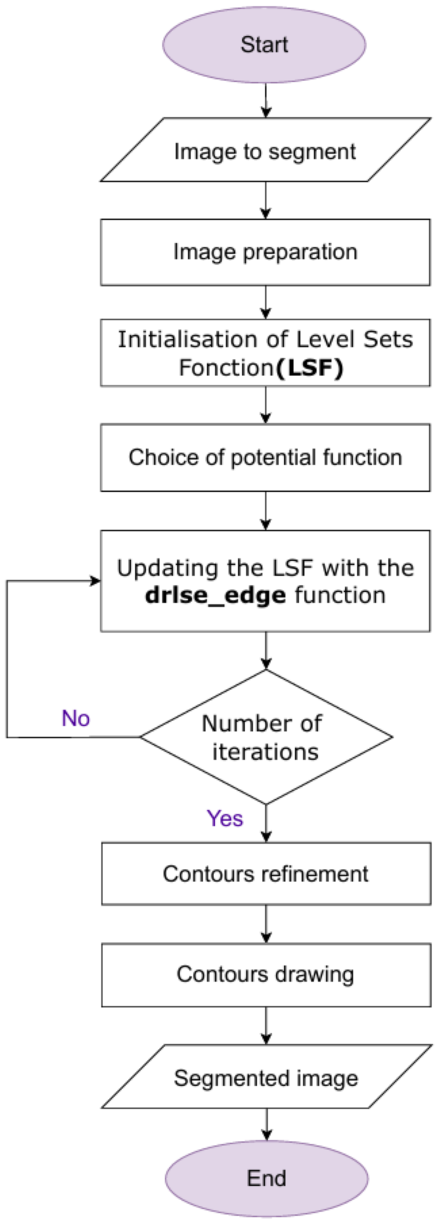

Start
Image to segment
Image preparation
Initialisation of Level Sets Fonction(**LSF**)
Choice of potential function
Updating the LSF with the **drlse_edge** function
No
Number of iterations
Yes
Contours refinement
Contours drawing
Segmented image
End

Figure 1. Flowchart of the DRLSE method

The Dirac function is used in the length term (Lg) to quantify the proximity of pixels to the contour. It is often based on the Heaviside function, which determines the boundaries of regions in the image by distinguishing pixels inside from those outside the segmented region.

The MATLAB function NeumannBoundCond(f) applies Neumann boundary conditions to the function f, meaning that the normal derivative of f is kept zero along the boundaries. This ensures that the values of f remain constant at the boundaries, stabilizing calculations and improving result accuracy in image processing and numerical simulation applications.

In summary, these concepts and functions are fundamental for the effective implementation of image segmentation methods based on Level-Sets [11], in particular to master regularization via DRLSE, boundary conditions, and contour precision in complex images [12].

4. FPGA IMPLEMENTATION

Our implementation is based on the PYNQ-Z2 FPGA board, a development platform using the Xilinx Zynq-7000 SoC. This board integrates both a Processing System (PS) and Programmable Logic (PL), providing a comprehensive solution for embedded hardware and software development [13, 14]. The model of the PC processor is Intel Core i7-3317U.1.70GHz. MATLAB runs on Windows10.

4.1 Implementation step

We present in Figure 2 the block diagram of the steps we followed during the implementation.

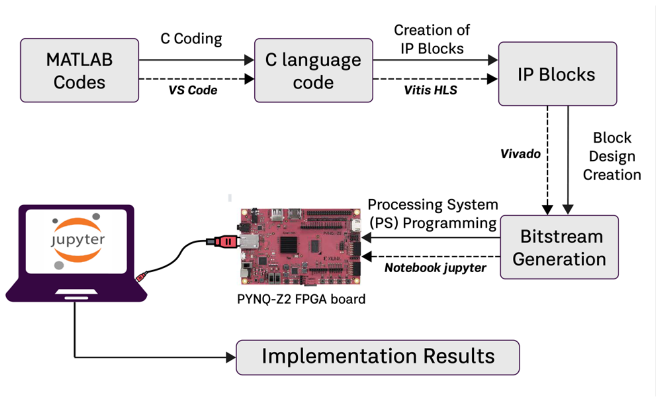


Figure 2. Simplified block diagram of the HLS approach

In the development process, we first write our program in C using Visual Studio Code (VS Code). We then run a simulation to verify its functionality. We compared the results with those obtained from MATLAB. Before generating VHDL code, we perform extensive testing and simulation of our C code in Vitis HLS, which allows us to identify and fix potential errors early in the development cycle [15]. Once the code is validated, we configure the necessary interfaces for the ports (e.g., AXI, BRAM, etc.). We then proceed to synthesize the code into a register transfer level (RTL) and ensure its accuracy by reviewing the generated reports. HLS directives (Pragma HLS) are used to optimize performance, resource utilization and latency.

The exported RTL file is used to generate IP blocks, which are then integrated into Vivado. This process is repeated for each IP block, ensuring continuous optimization of the system.

Using Vivado, we create a block design for the main system to be deployed on the FPGA, which includes the previously generated IPs [16]. Once the block design is validated, we create an HDL wrapper that converts the architecture diagram into an HDL program.

Once the block design is validated, it is converted to an HDL wrapper for synthesis and implementation. Vivado then optimizes the logic and resource utilization and generates detailed reports to ensure the quality of the design. Finally, a bitstream file is generated that configures the FPGA to execute the specified functions.

The block design is also exported as a TCL script and HWH file to allow the processor (CPU) to recognize the FPGA architecture and its various operations [17].

The final phase of the implementation takes place in Jupyter Notebook using Python. We prepare an SD card with the PYNQ image, then connect the FPGA board to the PC via Ethernet and USB to establish communication with the processor using the notebook's IP address [18].

The necessary files (bitstream, TCL, HWH and input data in .CSV format) are imported. An overlay is then created using the PYNQ. Overlay module, which allows control of the programmable logic (PL). The IP blocks are mapped to memory addresses using the MMIO class. Memory is allocated for each block, and input data is written to the appropriate buffer. A flush function is used to ensure the accessibility of the data, ensuring reliable and efficient transfer.

This process completes the implementation and ensures that the system operates correctly on the deployed FPGA.

4.2 Implementation of the DRLSE algorithm on FPGA

The first part of the implementation is subdivided into three distinct blocks, as shown in Figure 3: convolution, g computation, and gradient (g) computation. This subdivision allows for more efficient management of hardware resources by storing the outputs of each block in DDR memory and then duplicating them for use by subsequent blocks. Using DDR memory in an FPGA environment improves system performance by efficiently storing intermediate results, minimizing redundant computation, and optimizing parallel processing with seamless data flow. It also overcomes the limitations of internal memory, such as BRAM, which is often insufficient for complex tasks. In addition, DDR memory provides fast access to large data sets, enabling effective workload management and improved system adaptability.

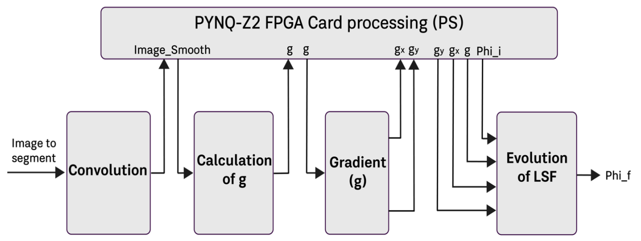


Figure 3. Diagram of the DRLSE algorithm

Each block corresponds to a top function, which we develop in the C language and simulate in VS Code, and to an IP block, which we create in Vitis HLS.

The "LSF Evolution" function takes the initial distance function ϕ_i, g, gx, and gy and produces ϕ_f as an output. During each iteration, ϕ_f is computed using an evolution equation that combines these data. The image contours (g) act as an external force that attracts the boundary of ϕ towards the detected contours in the image, allowing the LSF to follow the contours precisely. The gradients (gx) and (gy) are used to adjust the boundary to follow the intensity and texture changes in the image, ensuring accurate and detailed segmentation.

The block design provides a modular structure for the efficient implementation of the DRLSE algorithm on FPGA. Each block is designed to execute a specific function optimally, allowing for a robust and high-performance implementation of the algorithm in an FPGA environment. We rigorously test each step and verify the overall system's functionality to achieve performance and accuracy goals.

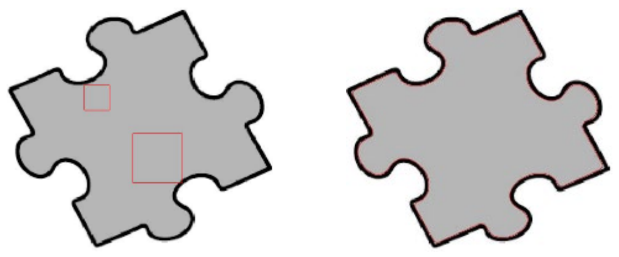
5. RESULTS AND DISCUSSION

We will present the results of the DRLSE model segmentation simulated on MATLAB for synthetic and medical images. We also discuss the implementation of the DRLSE algorithm on FPGA. The effectiveness of this method in segmentation of these images is evaluated by adjusting the parameters to observe their impacts on the segmentation results.

5.1 Results of the DRLSE model simulated on MATLAB for synthetic images

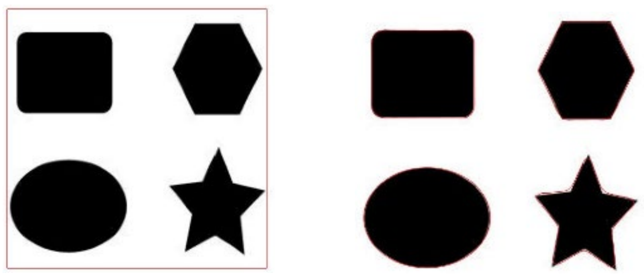
The DRLSE algorithm demonstrates successful segmentation of an image (Figure 4(b)) of a puzzle using two initial rectangular contours (Figure 4(a)) representing adjacent pieces of the puzzle.

We also tested the segmentation method on a synthetic image with several distinct geometric shapes. For initialization, we used a large outer contour that included all shapes (Figure 5(a)).



(a) Initial contours (b) Segmented image

Figure 4. Segmentation result with the DRLSE algorithm on a puzzle image



(a) Initial contour (b) Segmented image

Figure 5. DRLSE algorithm result on a synthetic image with multiple geometric shapes

Although the DRLSE algorithm is effective for segmenting concave and multiple objects, it may round off sharp corners, reducing precision for sharp anatomical structures (Figure 5(b)). Nevertheless, it remains promising and adaptable, particularly for medical segmentation needs.

5.2 Results of the DRLSE model simulated on MATLAB for medical images

Medical images present unique challenges due to their contrast variations, noise and complex anatomical structures. We illustrate the segmentation results of these images with the DRLSE algorithm, adjusting the weighting parameters (α, λ, and μ) seen in Eq. (6), to evaluate their impact on segmentation efficiency.

5.2.1 Adjustment of Parameters α, λ, and μ

Adjustment of Parameter α Figure 6 shows the results of the DRLSE algorithm on the image from the "Brain Tumor Data" database with different values of α, with 100 iterations.

For low values of α such as α=0.5, the contour evolves slowly, creating a conservative segmentation that closely follows the initial contours, ideal for capturing fine details. For medium values of α, such as α=1.5, these values accelerate contour evolution, allowing for fast segmentation suitable for medium-sized objects. A high value for α (α=2.5) intensifies contour evolution, allowing expansive or contractive segmentation suitable for very large objects.

The α parameter should be positive for outer contours and negative for inner contours to ensure accurate segmentation.

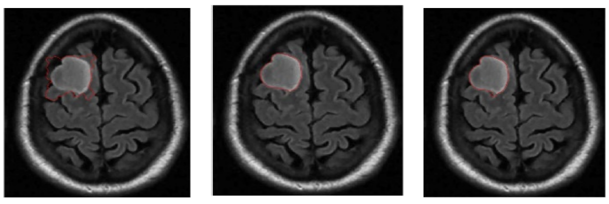


Figure 6. Results of the DRLSE algorithm on a medical image

Adjustment of λ Parameter Figure 7 shows the results of the DRLSE algorithm on a medical image with various values of λ with 100 Iterations. For low values of λ (λ=1), the contour follows the fine details and noise of the image, resulting in more irregular contours suitable for objects with complex edges. A moderate value such as λ=5 promotes smoother and more regular contours, suitable for objects with clear and well-defined edges. A high λ value (λ=8) produces very smooth and refined contours, particularly suited for segments that require maximum edge precision and significant noise reduction.

μ Parameter Adjustment Figure 8 shows the results of segmentation on a medical image by varying the values of μ and 100 Iterations.

For low values of μ, e.g. μ=0.02 distance regularization is minimized, allowing abrupt variations in the level set function. This can accelerate adaptation to local variations but risks introducing instability and artifacts.

For medium values of μ (μ=0.04), we offer a balanced compromise between stability and adaptability, ensuring a smooth level set function while allowing some flexibility to follow local variations.

A high μ value (μ=0.5) implies strong distance regularization, maintaining a very regular level set function and avoiding abrupt changes. This ensures robust convergence but limits the local adaptability of the algorithm.

The specific parameters and equations influence how the level set curve evolves, optimizing segmentation for more precise and detailed results.

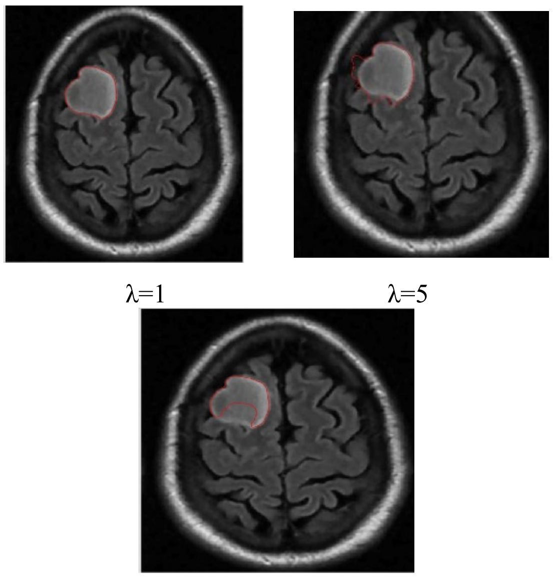


Figure 7. Segmentation results on a medical image

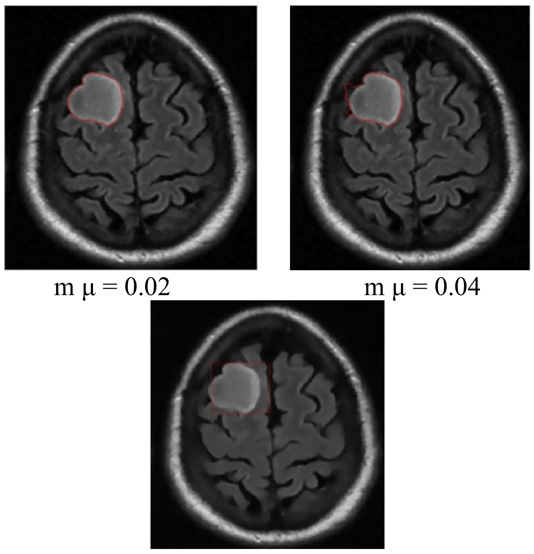


Figure 8. Results of the DRLSE algorithm on a medical image with different values of μ

5.3 Results on breast MRI images

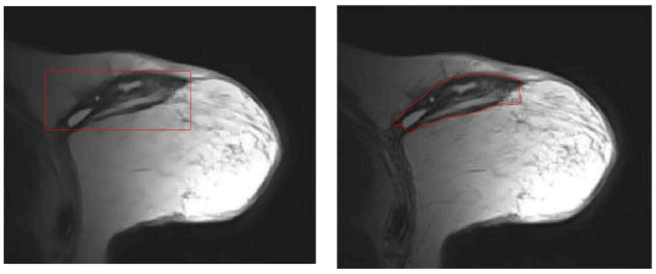


Figure 9. Segmentation result with the DRLSE algorithm

After several parameter adjustments, applying the DRLSE algorithm to this new medical image produced a less precise results in terms of segmenting the structures of interest. This observation prompted us to consider other approaches to improve segmentation quality. Figure 9 shows the segmentation result on breast MRI with α=1.5, λ=5, and μ=0.02.

5.4 Results on brain MRI images

To achieve accurate segmentation of medical images, we have carefully tuned the parameters α, λ, and μ, which play a critical role in the performance of the DRLSE algorithm. These parameters control key aspects of the level set evolution, such as the smoothness of the contour (α), the fitting to the object boundaries (λ), and the regularization of the level set function (μ).

Using an empirical approach, we iteratively adjusted these values to obtain the best segmentation results for the images shown in Figure 10. Specifically, for brain MRIs, we found that setting α=0.5, λ=6, and μ=0.02 provided optimal segmentation results. These values effectively balance the trade-off between contour smoothness and accuracy in detecting object boundaries.

The segmentation process required several iterations to refine these parameters, highlighting the importance of adapting them to the specific characteristics of the images being analyzed. The results confirm the robustness of the DRLSE algorithm in delimiting structures in medical images.

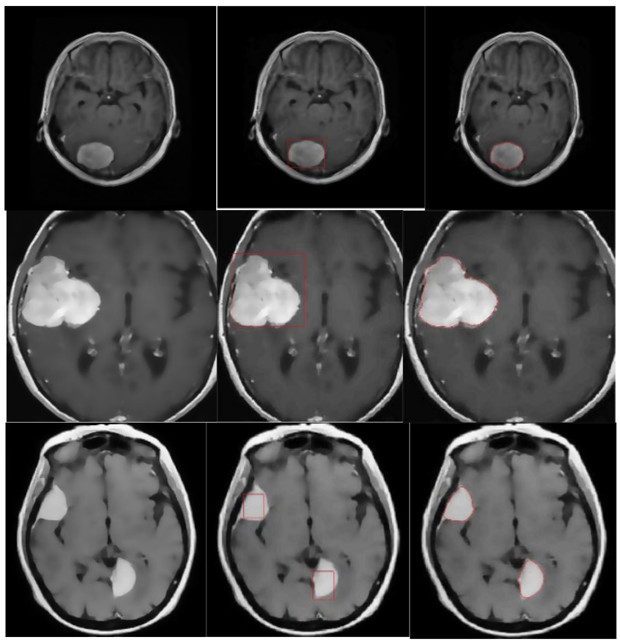


Figure 10. Segmentation result. with the DRLSE algorithm

5.5 Results of MRI and echographic images

The algorithm's effectiveness depends on the initial position of the level set curve and the defined parameters, with bad choices potentially leading to inaccurate segmentations. In cardiac MRIs, low contrast transitions complicate contour evolution, especially in poorly defined anatomical areas.

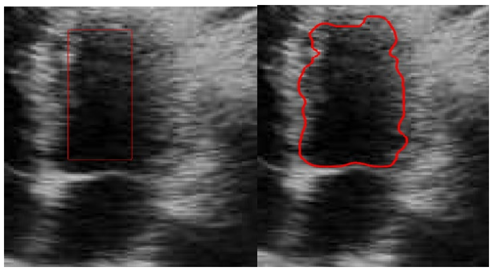
After several attempts, we successfully segmented some cardiac MRI and echographic images.

In Figure 11, we present the segmentation results applied to both MRI and ultrasound images, obtained using the DRLS (Distance Regularized Level Set) algorithm. The parameters used to segment MRI and ultrasound images were selected after several tests to achieve precise and rigorous segmentation that accurately defines the target contour.

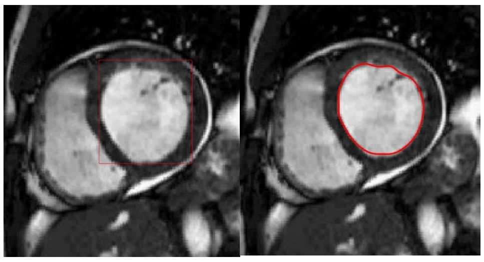
Using the parameters μ=0.04, λ=5, α=-1, σ=5 and 200 iterations, the segmentation of the cardiac ultrasound image demonstrates the effectiveness of the DRLSE algorithm in accurately detecting the edges of the cardiac wall, even in the presence of noise (Figure 11(a)). The moderate regularization ensured by μ=0.04 preserves the details of the contours, while λ=5 strikes a balance between contour attraction and stability. The parameter α=−1 enhances the algorithm’s ability to handle noisy regions, and σ=5 reduces noise while maintaining critical transitions in the image.

The set of parameters μ, λ, α, and σ allows for accurate and robust segmentation, proving the adaptability of the algorithm to the challenges posed by ultrasound images.

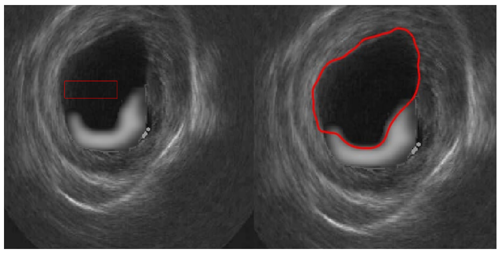
For the MRI heart image, the segmentation with the parameters μ=0.2, λ=1, α=1, σ=6, and 200 iterations demonstrates the DRLSE algorithm's ability to effectively highlight contrasts between different tissues, allowing for precise localization of the targeted contour (Figure 11(b)).



(a) The cardiac ultrasound image



(b) The cardiac MRI image



(c) The ultrasound image

Figure 11 . Segmentation results of MRI and echographic images

For the image shown in Figure 11(c), segmentation with the parameters μ =0.2, λ =5, α =-1, σ =6, and 200 iterations produces a satisfactory result. The segmented contours are both sufficiently sharp and well defined, while maintaining a certain smoothness that prevents the influence of noise. These parameter values provide an optimal compromise between fidelity to the image intensity data and contour smoothness, resulting in accurate segmentation while minimizing imperfections due to artifacts or noise.

However, the approach based on local information (gradients) has limitations when dealing with complex structures or artifacts, highlighting the need for global information integration to achieve accurate segmentation.

5.6 Results of the DRLSE model implementation on FPGA

In FPGA design, a block-based design uses predefined modules. Each block represents a specific functionality, such as processors or memory controllers, and the blocks are interconnected via standardized interfaces to enable communication between them. This modular approach facilitates the efficient use of FPGA resources, simplifying the overall FPGA design process [19].

Figure 12 represents the final block design of our architecture for the implementation of the DRLSE algorithm.

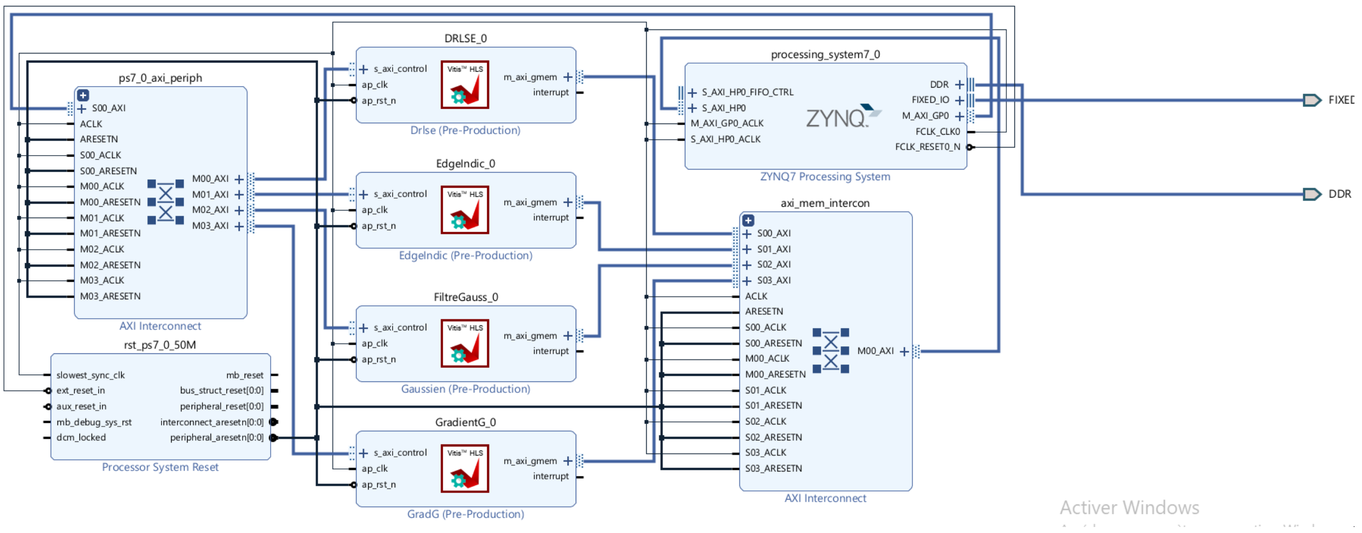


Figure 12. Block design for FPGA implementation of the DRLSE segmentation algorithm

By using the same parameter configuration and initial contour as in the MATLAB simulation, we implemented the DRLSE method on the FPGA. The visual results, shown in Figure 13, demonstrate the algorithm's effectiveness in segmenting medical images.

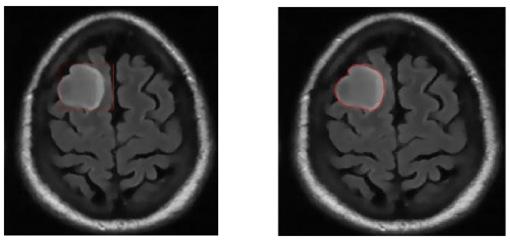


Figure 13. Result of implementing the DRLSE method

The implementation of the DRLSE algorithm on FPGA demonstrated satisfactory segmentation accuracy, with a rigorous visual comparison to the results obtained through MATLAB simulation [20]. No significant differences or errors were observed in the contours generated from the ϕ function.

We present the utilization and power reports for the DRLSE (Distance Regularized Level Set Evolution) algorithm implementation on FPGA. This implementation involves examining several crucial factors to assess the performance and efficiency of the algorithm in hardware. Specifically, the hardware resource usage is analyzed, including the consumption of Look-Up Tables (LUTs), Block RAMs (BRAMs), and Digital Signal Processing (DSP) slices, which are summarized in Table 1 these metrics offer insight into how effectively the FPGA's resources are utilized for the algorithm's operations. Additionally, an estimation of the power consumption associated with the DRLSE algorithm's FPGA implementation is provided in Table 2 This power report highlights the energy efficiency of the design and can serve as an essential guideline for optimizing power consumption in hardware accelerators.

Table 1. Utilization report summary

Consumption	Utilization Report (%)
LUT	59
LUTRAM	6
FF	37
BRAM	95
DSP	74

Table 2. On-chip power distribution

Component	Power (W)	Percentage
Dynamic	1.940	92%
Clocks	0.057	3%
Signals	0.159	8%
Logic	0.107	5%
BRAM	0.042	2%
DSP	0.047	2%
PS7	1.527	79%
Device Static	0.163	8%

To achieve these results, we optimized our computations by consolidating them into a single execution phase. This approach required the use of buffers for matrix updates, which significantly impacted the utilization of computational resources. Specifically, the need for these buffers increased the consumption of both memory and processing units within the FPGA, leading to a rise in resource usage. As a result, the overall execution time of the algorithm was extended, with a total processing time of 105.8523 seconds.

A distinctive feature of our approach was the frequent subdivision of computational blocks into smaller sub-blocks. This strategy was particularly aimed at optimizing memory usage by favoring DDR (Double Data Rate) memory over the FPGA's Block Ram (Bram). While DDR memory provides a larger available space for storing data, it generally offers slower access speeds compared to BRAM, which is situated closer to the FPGA's logic and thus facilitates faster data retrieval. Despite this inherent drawback, our decision to rely on DDR memory was driven by the goal of maximizing the overall memory efficiency, particularly when handling large datasets that would exceed the capacity of BRAM alone. This trade-off between memory efficiency and data access speed was a key consideration in balancing resource utilization and performance.

6. CONCLUSION

Our project has successfully addressed several key technical challenges, marking a significant advancement in the field of medical image segmentation. The DRLSE algorithm, based on distance regularization, has demonstrated its ability to deliver stable and precise segmentation. However, the DRLSE algorithm faces limitations in accurately segmenting sharp anatomical structures. Additionally, it utilizes DDR memory on the FPGA, which has lower performance compared to BRAM, potentially leading to slower data access and processing.

To overcome these limitations, several solutions are proposed. First, the integration of more robust neural networks, coupled with data augmentation techniques, could enhance the algorithm's ability to process complex structures more effectively. Second, a hybrid memory architecture combining DDR and BRAM is suggested to optimize data throughput, by storing critical data in BRAM for faster access, while utilizing DDR for less time-sensitive data. Finally, incorporating traditional segmentation methods, such as active contours, could complement the DRLSE algorithm to improve its handling of fine anatomical structures.

These proposed improvements would enhance the algorithm's efficiency, making it better suited for real-world applications. We have successfully implemented the algorithm on an FPGA platform after conducting a comprehensive analysis of resource utilization, execution time, and segmentation accuracy, comparing the results to those obtained using MATLAB. This FPGA implementation represents a crucial step toward achieving our goal of providing highly efficient image segmentation solutions tailored for embedded environments. Furthermore, it lays the groundwork for potential precision improvements through optimized use of advanced hardware resources, highlighting the strategic importance of our approach in meeting the stringent requirements of real-time, embedded applications, and reinforcing our commitment to innovation in the medical field.

ACKNOWLEDGMENT

We would like to express our deep gratitude to Prof. BENYAHIA from the Medical Imaging Department at Mustapha Bacha Hospital in Algiers for his invaluable support in providing the essential medical images for our project.

REFERENCES

[1] Kass, M., Witkin, A., Terzopoulos, D. (1988). Snakes: Active contour models. International Journal of Computer Vision, 1(4): 321-331. https://doi.org/10.1007/BF00133570

[2] Xu, C., Yezzi, A., Prince, J.L. (2000). On the relationship between parametric and geometric active contours. In Conference Record of the Thirty-Fourth Asilomar conference on signals, systems and computers (Cat. No. 00CH37154), Pacific Grove, CA, USA, pp. 483-489. https://doi.org/10.1109/ACSSC.2000.911003

[3] Malladi, R., Sethian, J.A., Vemuri, B.C. (1995). Shape modeling with front propagation: A level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(2): 158-175. https://doi.org/10.1109/34.368173

[4] Sethian, J.A. (1999). Advances in fast marching and level set methods for propagating interfaces. Hyperbolic Problems: Theory, Numerics, Applications. International Series of Numerical Mathematics, 130: 855-864. https://doi.org/10.1007/978-3-0348-8724-3_36

[5] Khelifi, R. (2008). Segmentation des images coronarographiques par les Contours Actifs. Doctoral dissertation, Alger, Ecole Nationale Polytechnique. http://repository.enp.edu.dz/xmlui/handle/123456789/4050.

[6] Chen, Y., Chen, G., Wang, Y., Dey, N., Sherratt, R.S., Shi, F. (2019). A distance regularized level-set evolution model based MRI dataset segmentation of brain's caudate nucleus. IEEE Access, 7: 124128-124140. https://doi.org/10.1109/ACCESS.2019.2937964

[7] Li, C., Xu, C., Gui, C., Fox, M.D. (2010). Distance regularized level set evolution and its application to image segmentation. IEEE Transactions on Image Processing, 19(12): 3243-3254. https://doi.org/10.1109/TIP.2010.2069690

[8] Viswanath, K., & Gunasundari, R. (2015). Analysis and implementation of kidney stone detection by reaction diffusion level set segmentation using xilinx system generator on FPGA. VLSI Design, 2015(1): 581961.https://doi.org/10.1155/2015/581961

[9] Wang, D. (2019). Extremely optimized DRLSE method and its application to image segmentation. IEEE Access, 7: 119603-119619. https://doi.org/10.1109/ACCESS.2019.2937512

[10] Li, C., Xu, C., Gui, C., Fox, M.D. (2005). Level set evolution without re-initialization: A new variational formulation. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, pp. 430-436. https://doi.org/10.1109/CVPR.2005.213

[11] Caselles, V., Kimmel, R., Sapiro, G. (1997). Geodesic active contours. International Journal of Computer Vision, 22: 61-79. https://doi.org/10.1023/A:1007979827043

[12] Lakra, M., Kumar, S. (2021). Disparity computation through PDE and data-driven CeNN technique. Traitement du Signal, 38(4): 1051-1059. https://doi.org/10.18280/ts.380415

[13] PYNQ: Python Productivity for Zynq. (2024). Xilinx. http://www.pynq.io.

[14] Jarrah, A., Amri, S. (2022). Optimized FPGA-based implementation of brain tumor detection by combining K-means and grey wolf optimization algorithms. Traitement du Signal, 39(6): 1879-1891. https://doi.org/10.18280/ts.390601

[15] Vitis High-Level Synthesis User Guide. (2021). https://docs.amd.com/r/2021.2-English/ug1399-vitis-hls.

[16] Vivado Design Suite. (2021). https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0002-vivado-design-hub.html.

[17] Mándi, Á., Máté, J., Rózsa, D., Oniga, S. (2021). Hardware accelerated image processing on FPGA based PYNQ-Z2 board. Carpathian Journal of Electronic and Computer Engineering, 14(1): 20-23. https://doi.org/10.2478/cjece-2021-0004

[18] PYNQ. (2021). Getting Started with PYNQ. https://pynq.readthedocs.io/en/latest/getting_started.html.

[19] Fradi, M., Lazhar, K., Zahzah, E.H., Machhout, M. (2024). FPGA implementation of a CNN application for ECG class detection. Traitement du Signal, 41(1): 179-188. https://doi.org/10.18280/ts.410114

[20] MATLAB for Artificial Intelligence. https://www.mathworks.com/, accessed on Jun. 10, 2022.