# Breast Lesion Segmentation Using a 3D Active Contours Without Edges Approach with a GPGPU Implementation

Olmo Zavala-Romero, Anke Meyer-Baese, and Marc Lobbes

## Abstract

Breast cancer is the second most frequently diagnosed type of cancer in women, just after skin cancer, and is the second leading cause of cancer deaths. Thanks to improvements in early detection and treatments, the death rates of breast cancer have been reduced considerably. This work describes a 3D breast lesion segmentation algorithm that can be used for preoperative staging, evaluation after chemotherapy, and screening of women at high risk of developing breast cancer. The segmentation algorithm is a 3D implementation of the Active Contours Without Edges (ACWE) method, introduced by Chan and Vese [2001]. The proposed algorithm uses the General-Purpose Computing on Graphics Processing Units (GPGPU's) to accelerate the original model by parallelizing two main steps of the segmentation process, the computation of the Signed Distance Function (SDF) and the evolution of the segmented curve. The segmentation process is visualized in real time with a graphical user interface (GUI) developed with Qt and OpenGL.

## Active Contours Without Edges

The ACWE segmentation algorithm is based on the level set method. The objective of this method is to follow a function $\gamma(t)$ that is propagating with curvature speed **F** by a higher dimension function $\Phi(\vec{x},t)$ whose zero-level set corresponds to the position of the function of interest, as shown in figure 1.

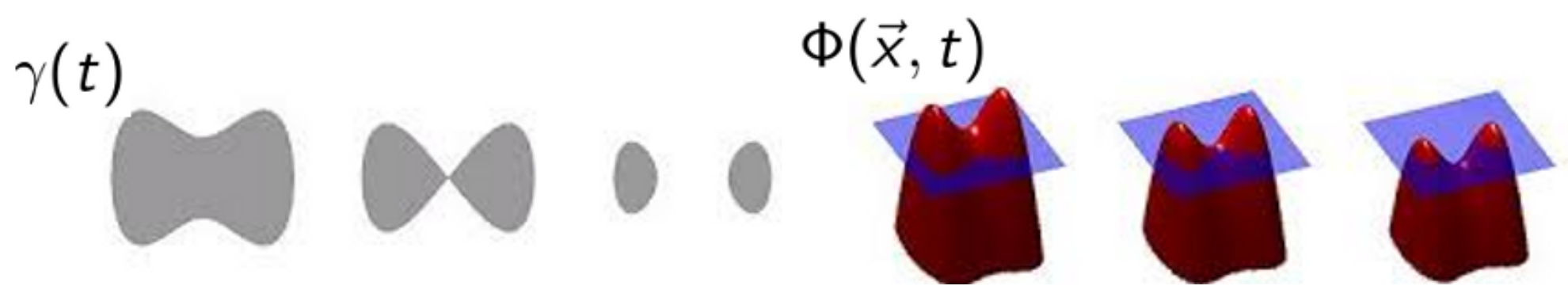


**Figure 1:** 2D example of the level set method.

The proposed speed **F** for the ACWE algorithm is:

$$\begin{aligned} F(c_1,c_2,C) &= v\cdot \text{length}(C) + w\cdot \text{area}(C) \\ &+ \lambda_1 \int_{inside(C)} |u_0(x,y,z)-c_1|^2 dx\,dy \\ &+ \lambda_2 \int_{outside(C)} |u_0(x,y,z)-c_2|^2 dx\,dy \end{aligned} \qquad (1)$$

where $C$ is the segmented curve, $\boldsymbol{u_0}$ is the 3D image, and the functions **length** and **area** are regularizing terms of the functional. The constants $c_1$ and $c_2$ are the average values of the image inside and outside the segmented curve, and $\lambda_1, \lambda_2, v$ and $w$ are parameters used to weight each of the functional terms. For this work $\lambda_1, \lambda_2$, and $v = 1$ and $w = 0$. Equation 1 is solved by the level set method [Stanley and James, 1988] and the solution is given by:

$$\begin{aligned} \frac{\partial \Phi}{\partial t} &= \delta(\Phi)[v \,\text{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) \\ &- \lambda_1(u_0(x,y)-c_1)^2 + \lambda_2(u_0(x,y)-c_2)^2] \end{aligned} \qquad (2)$$

where $\Phi$ is the level set and $\text{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)$ is the image curvature. Finally, the segmented contour is evolved until it converges.

## References

H. Breu, J. Gil, D. Kirkpatric, and M. Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Learning*, 1995.

T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

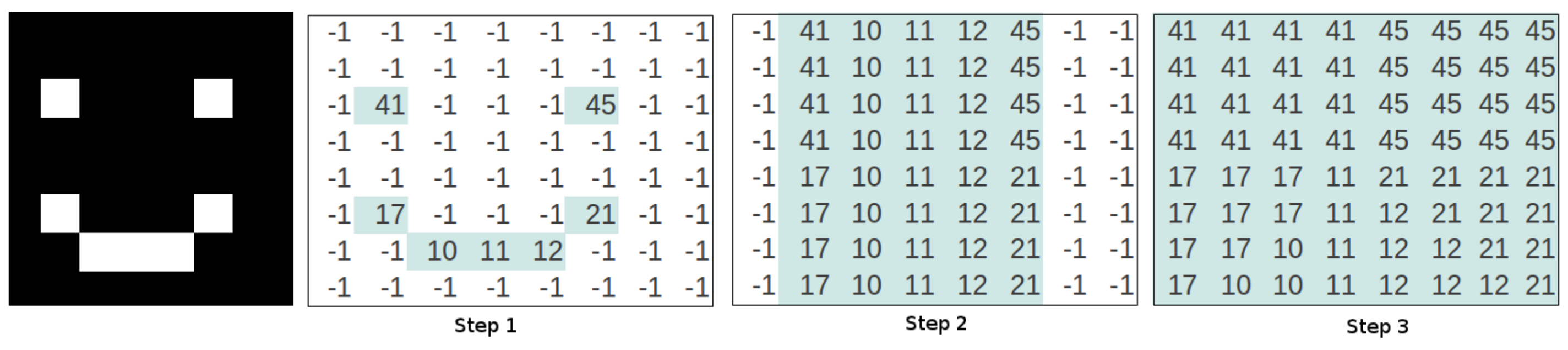
S. Lankton. Sparse field methods-technical report. *Georgia institute of tech*, 2009.

C. R. Maurer, Jr, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):265–270, 2003.

Osher Stanley and A. Sethian James. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Computational Physics*, pages 12 –49, 1988.

## Signed Distance Function

The ACWE algorithm requires an initialization contour that corresponds to the zero-level set of $\Phi$. To initialize the higher dimension function $\Phi(\vec{x},t)$ the SDF is commonly used. In this work, the method introduced by Maurer et al. [2003] is used to compute the SDF. This method obtains the exact Euclidean distance of binary images in any dimension. Maurer et al. proposed that, rather than computing the distance directly, they treat every feature pixel (pixel with value of 1) as a Voronoi site, and find the discrete Voronoi diagram of the image.



Step 1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | 41 | -1 | -1 | -1 | 45 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | 17 | -1 | -1 | -1 | 21 | -1 | -1 |
| -1 | -1 | 10 | 11 | 12 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

Step 2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -1 | 41 | 10 | 11 | 12 | 45 | -1 | -1 |
| -1 | 41 | 10 | 11 | 12 | 45 | -1 | -1 |
| -1 | 41 | 10 | 11 | 12 | 45 | -1 | -1 |
| -1 | 41 | 10 | 11 | 12 | 45 | -1 | -1 |
| -1 | 17 | 10 | 11 | 12 | 21 | -1 | -1 |
| -1 | 17 | 10 | 11 | 12 | 21 | -1 | -1 |
| -1 | 17 | 10 | 11 | 12 | 21 | -1 | -1 |
| -1 | 17 | 10 | 11 | 12 | 21 | -1 | -1 |

Step 3

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 41 | 41 | 41 | 41 | 45 | 45 | 45 | 45 |
| 41 | 41 | 41 | 41 | 45 | 45 | 45 | 45 |
| 41 | 41 | 41 | 41 | 45 | 45 | 45 | 45 |
| 41 | 41 | 41 | 41 | 45 | 45 | 45 | 45 |
| 17 | 17 | 17 | 11 | 21 | 21 | 21 | 21 |
| 17 | 17 | 17 | 11 | 12 | 21 | 21 | 21 |
| 17 | 17 | 10 | 11 | 12 | 12 | 21 | 21 |
| 17 | 10 | 10 | 11 | 12 | 12 | 12 | 21 |

**Figure 2:** Example of the outputs obtained at each step of Maurer et al. algorithm applied to a 2D image.

**Step 1.** Assign to each pixel that represents a Voronoi site its corresponding index on the image.
**Step 2.** Compute a 1D partial Voronoi diagram; each pixel is assigned the index of its closest Voronoi site for first dimension.
**Step 3.** Compute the 2D partial Voronoi diagram.

## Results

The SDF algorithm was compared with the Matlab function *bwdist* which uses an efficient sequential algorithm introduced by Breu et al. [1995]. The proposed implementation in this work runs 3.5 times faster than the Matlab function *bwdist* for image sizes of $160^3$.

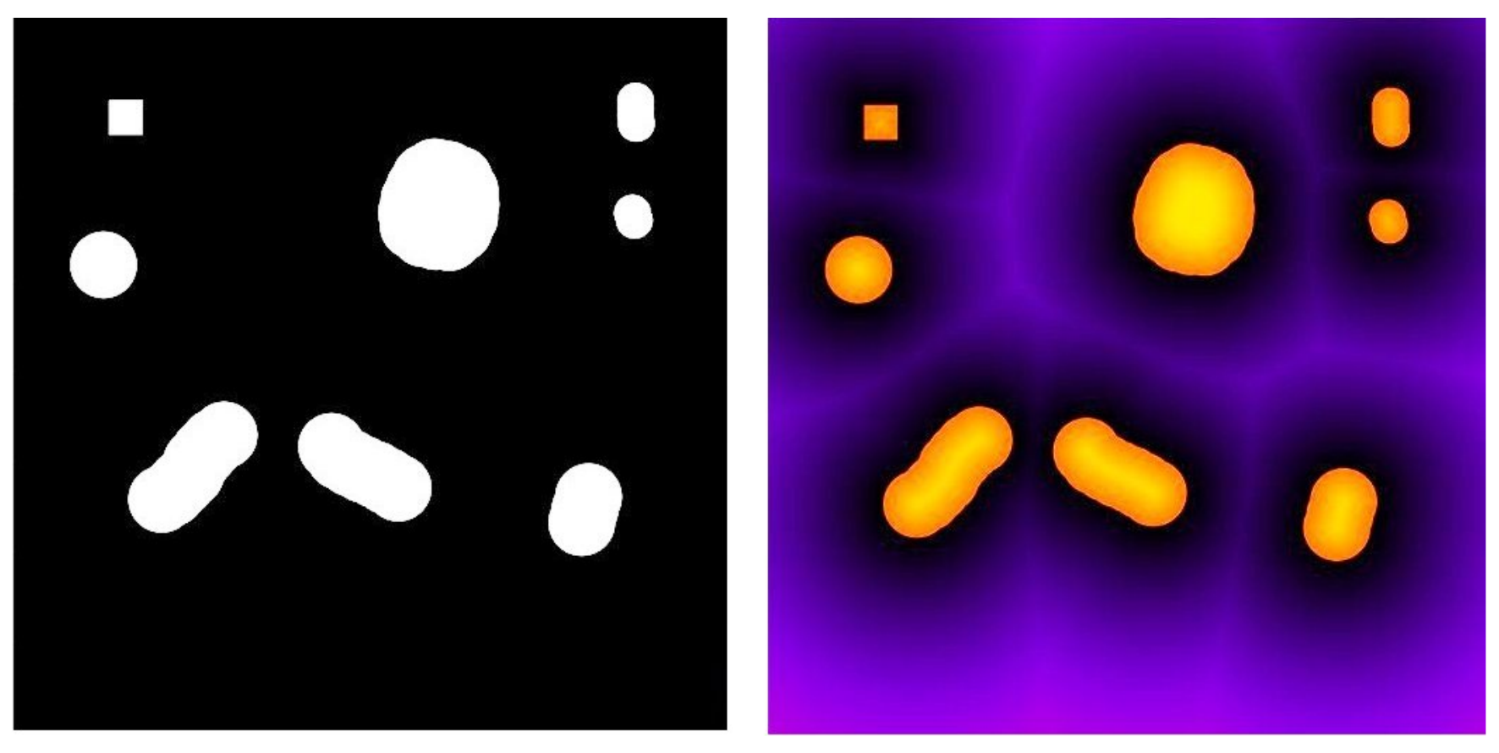


**Fig 3:** SDF 2D Example. Ligther colors represent higher distances in the output image on the right.

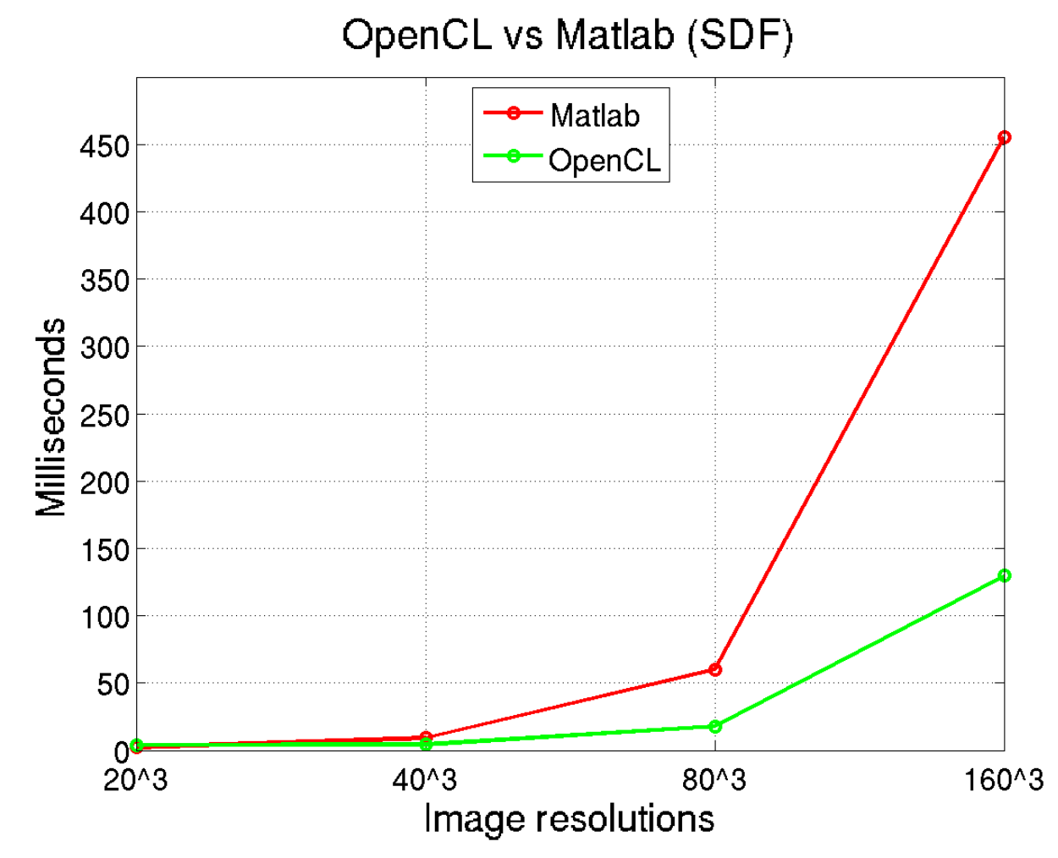


**Fig 4:** Time comparison between Matlab and OpenCL implementations (100 iterations).

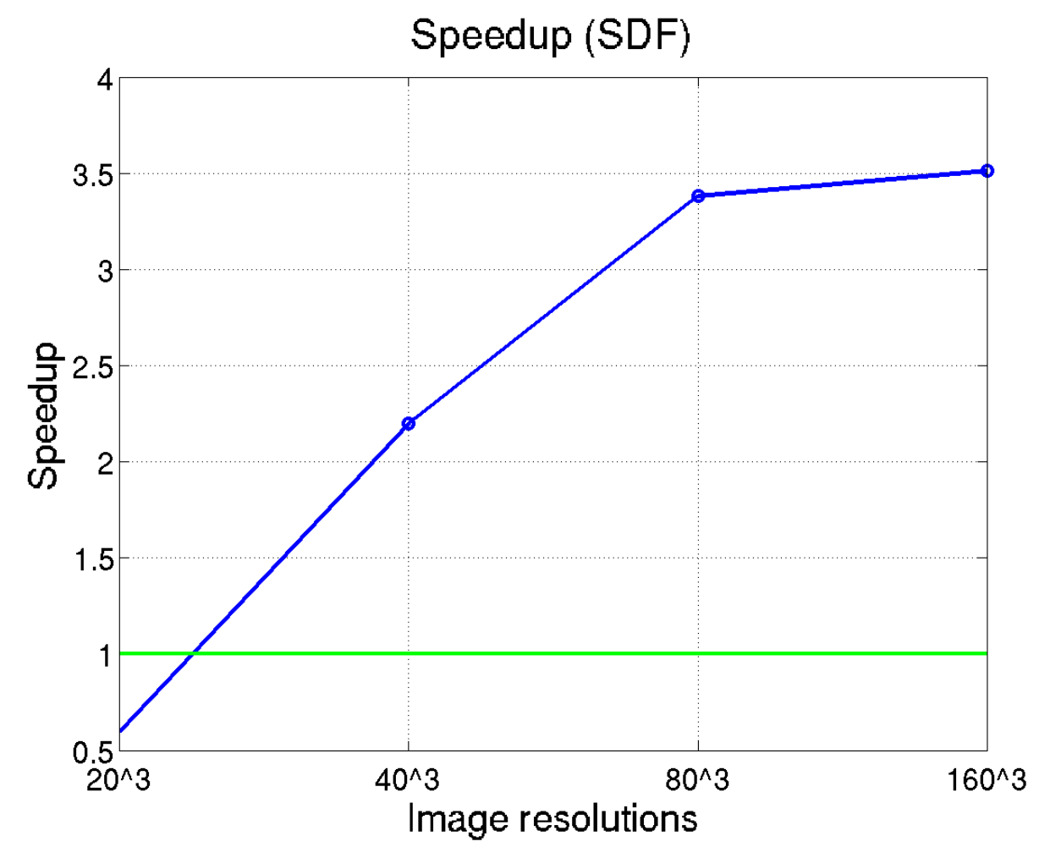


**Fig 5:** SDF speedup obtained for different image sizes.

Two visualization schemes were developed using OpenGL and Qt. The first one uses three orthogonal planes that display interpolated data from the 3D image. The second scheme uses a ray casting method to show a translucent view of the 3D data.

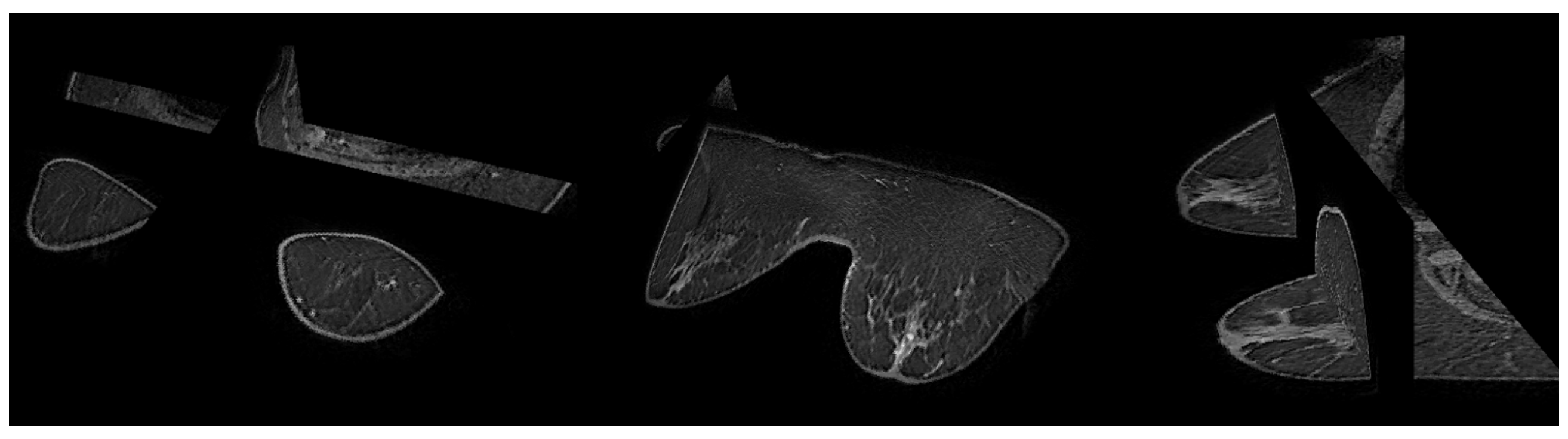


**Fig 6:** 3D Visualization using orthogonal planes.

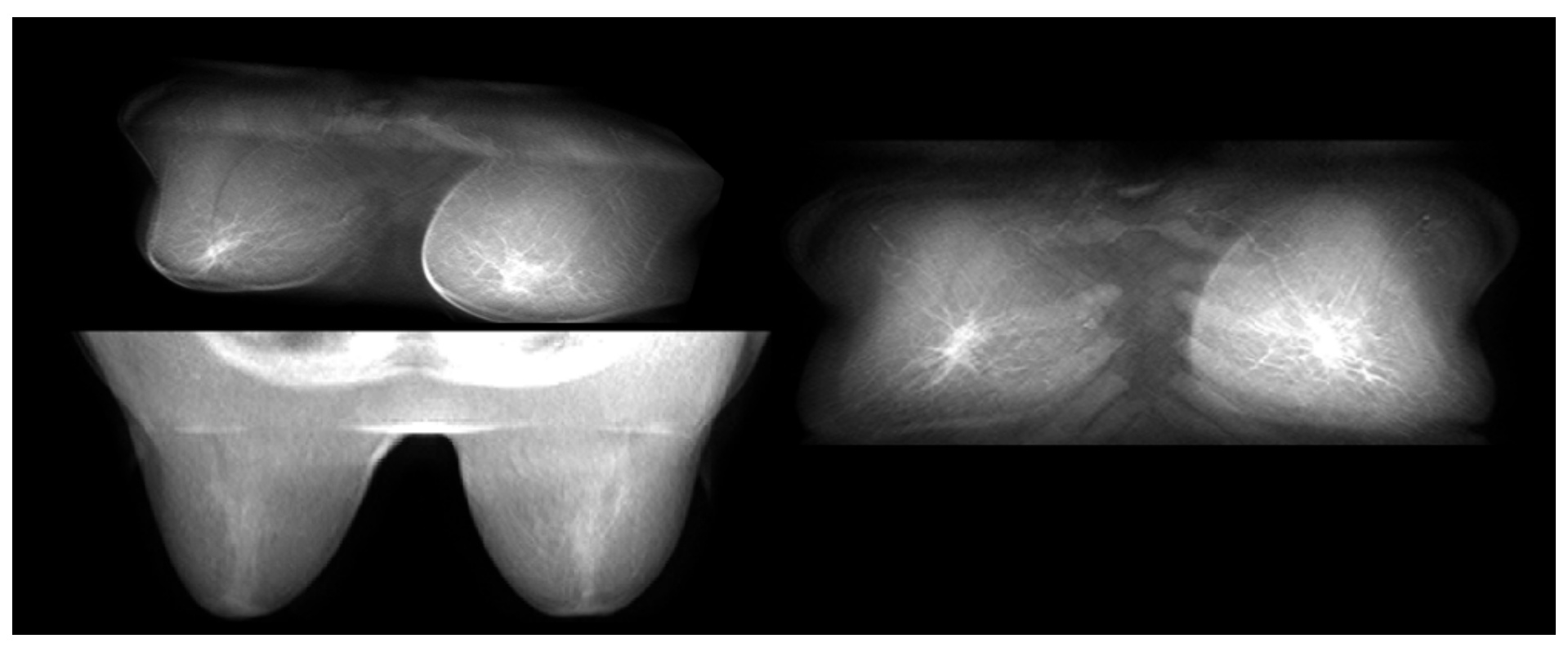


**Fig 7:** 3D Visualization using ray casting.

The OpenCL implementation of the ACWE was compared with an extended 3D Matlab implementation of Lankton [2009] 2D version. The Matlab implementation only updates the values of $\Phi$ in a **narrow band** close to the segmented curve to optimize the computational time in the CPU.

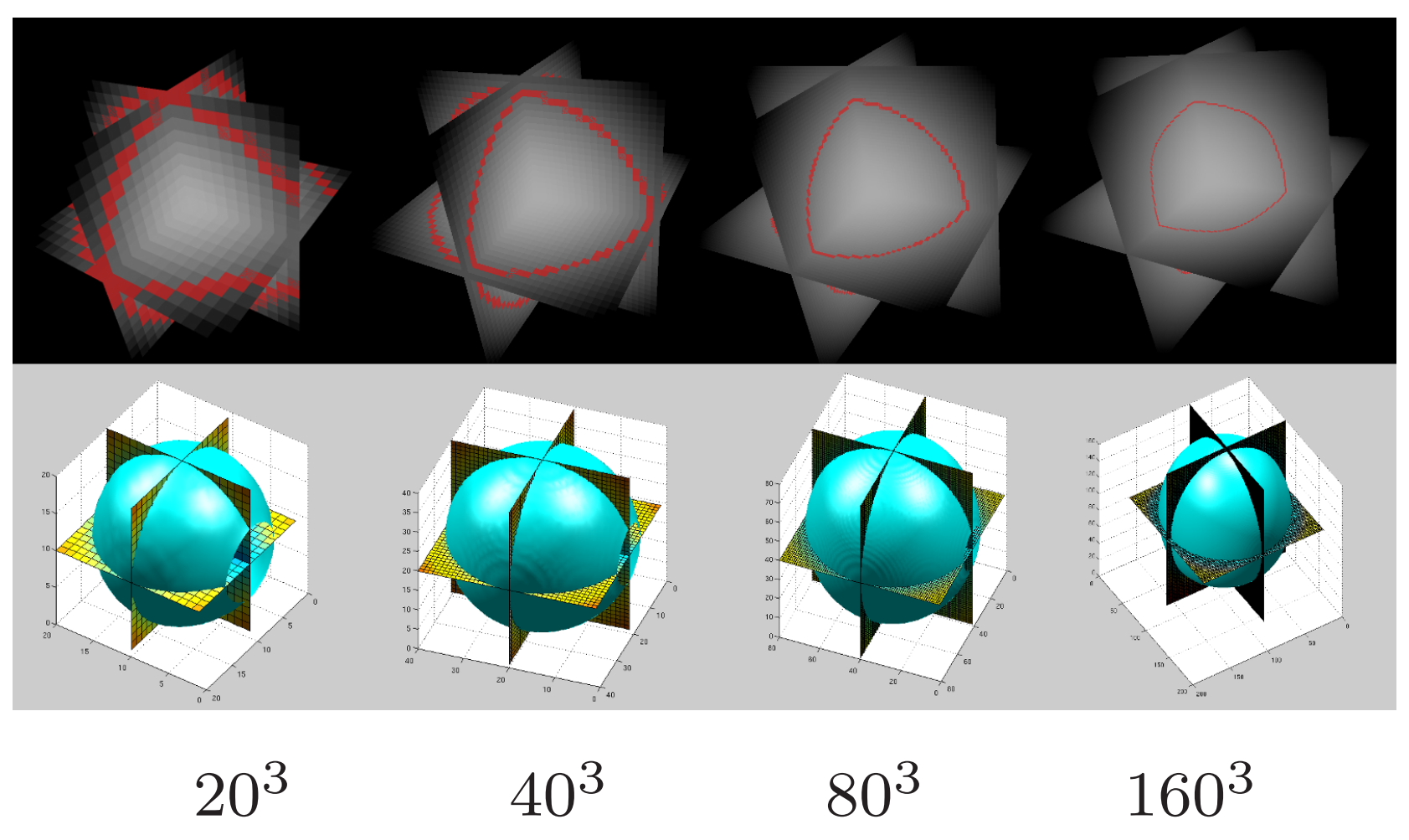


**Fig 9:** 3D Segmentation using OpenCL (top row) and Matlab (bottom row).

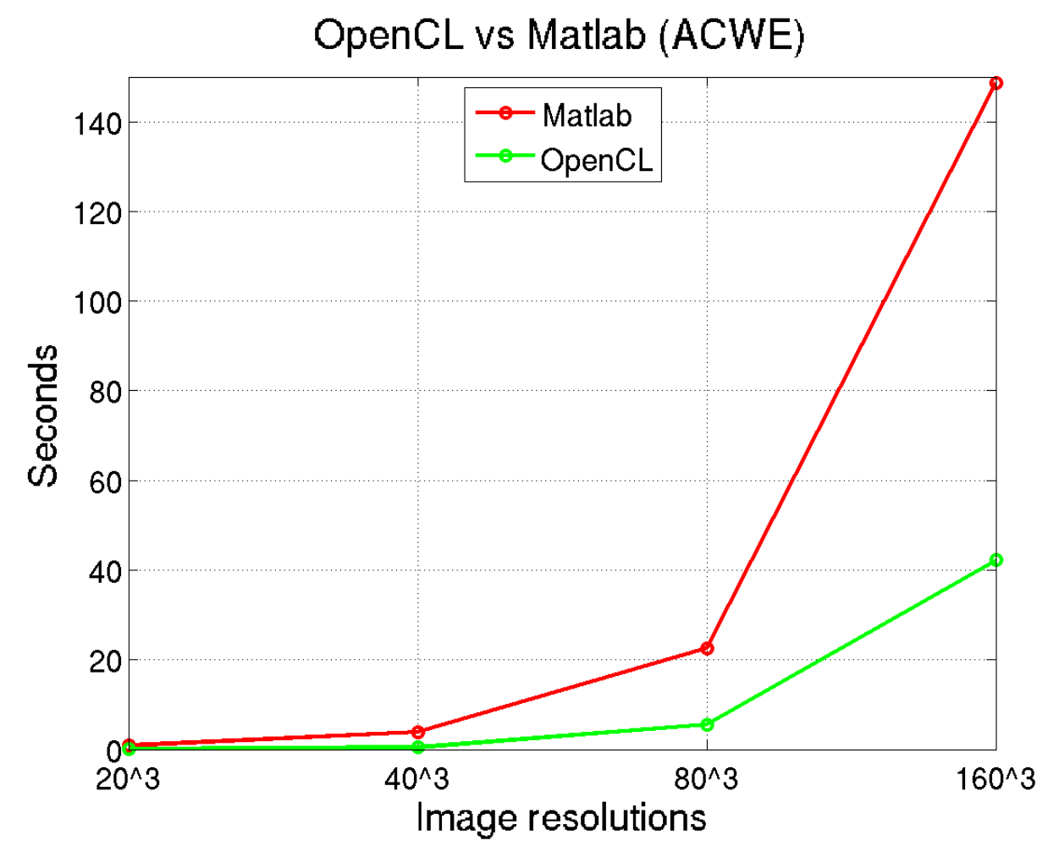


**Fig 10:** Time comparison between Matlab and OpenCL implementations (100 iterations).

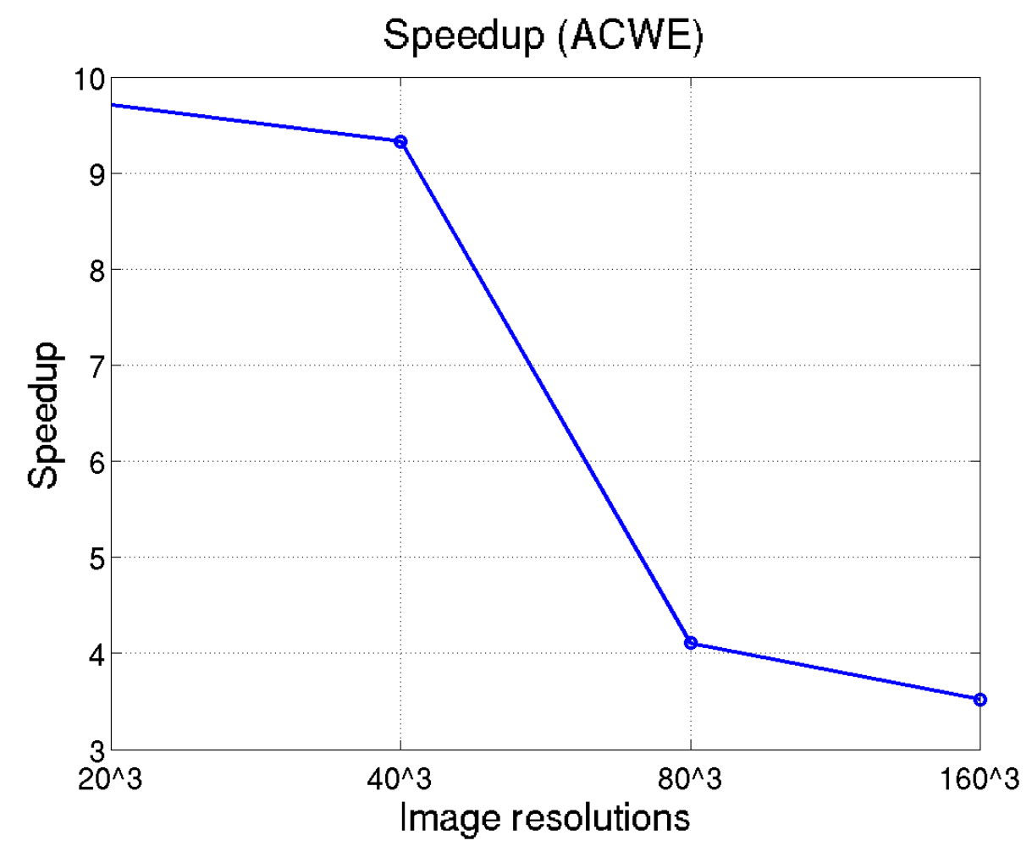


**Fig 11:** ACWE speedup obtained for different image sizes.

## Future Work

The speedup obtained by the proposed segmentation algorithm decreases for larger image sizes. This is not the expected behavior and requires further analysis of each OpenCL module implementation. Figure 12 shows that most OpenCL modules have a computational cost $O(n^3)$ and the expected computational cost is $O(n)$ (when enough resources are available in the GPU).

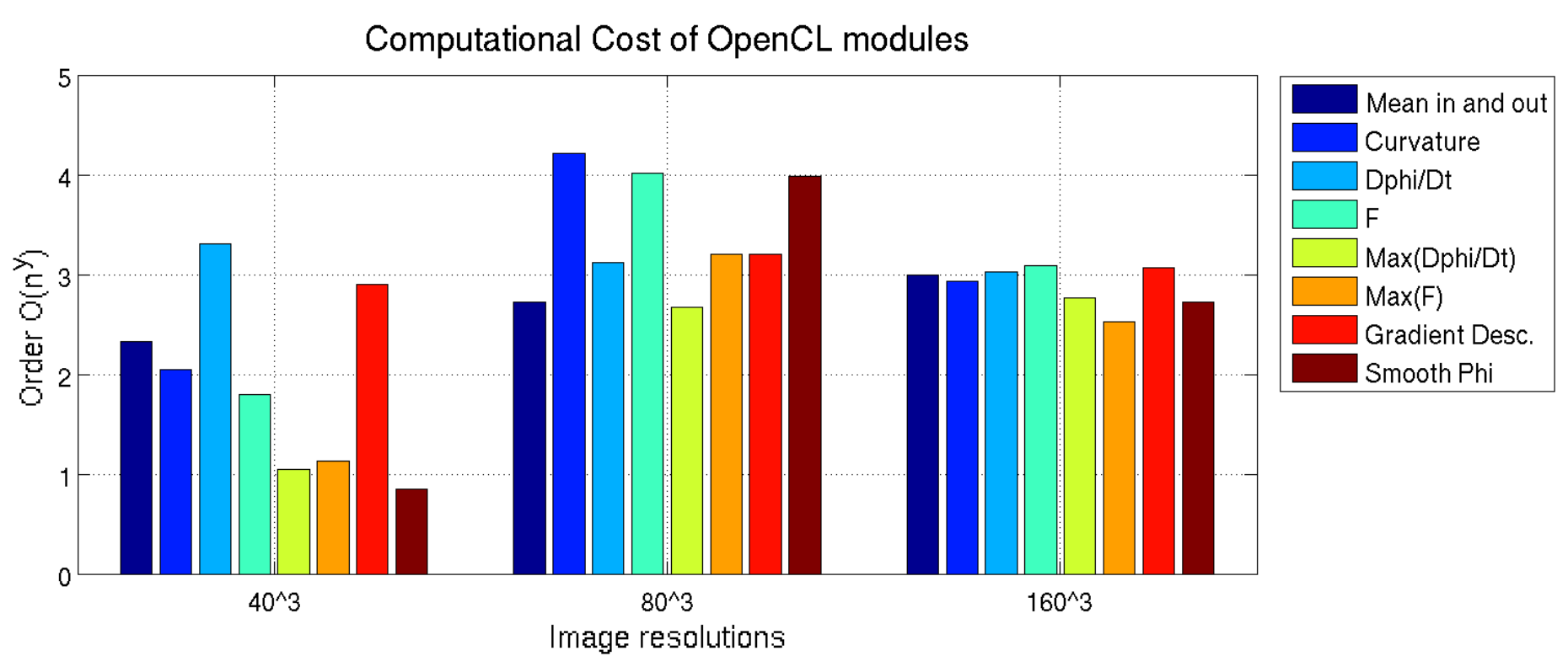


**Fig 12:** Computational cost for each OpenCL module.

## Conclusions

The described ACWE OpenCL implementation reduces the computational time of the algorithm and could be used to assist physicians in the delineation of breast lesions, and in other areas of medical imaging. Further improvements are needed to accelerate even more the segmentation algorithm in order to use it as a real time decision tool or with larger images. Finally, a study comparing the segmentation obtained with the ACWE algorithm and the segmentation obtained from experts in the field is required to evaluate the benefits of the automatic segmentation.

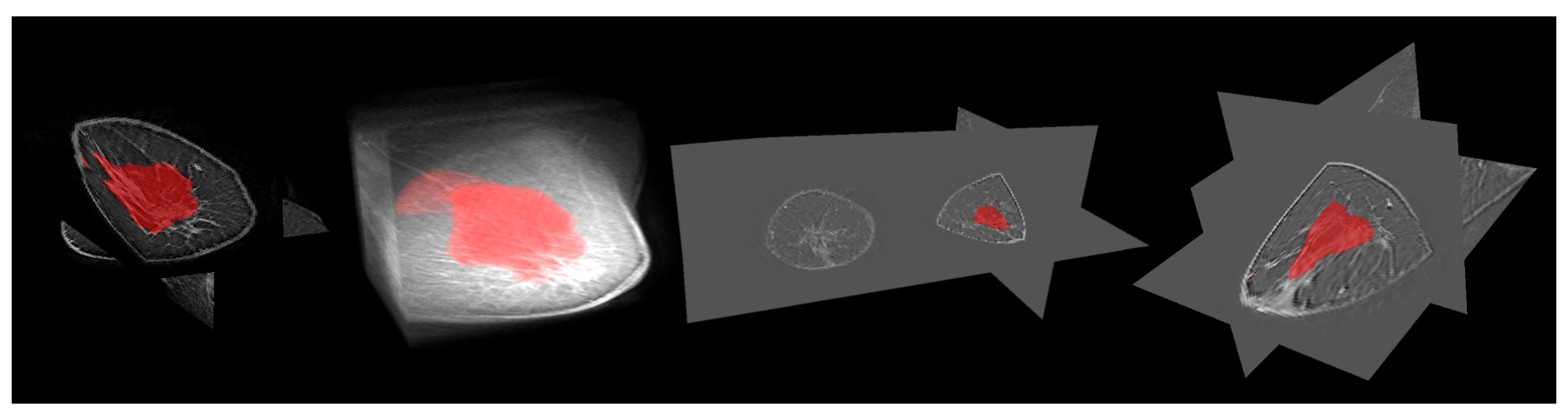


**Fig 13:** Examples of breast lesions segmentations.