

# A NEW 3D IMAGE SEGMENTATION METHOD FOR PARALLEL ARCHITECTURES

*Przemyslaw Lenkiewicz<sup>1,2</sup>, Manuela Pereira<sup>1</sup>, Mário M. Freire<sup>1</sup>, and José Fernandes<sup>2</sup>*

<sup>1</sup>IT-Networks & Multimedia Group, Dep. Computer Science, University of Beira Interior, Portugal  
<sup>2</sup>Microsoft Portugal

## ABSTRACT

This paper presents a novel model for 3D image segmentation and reconstruction. It has been designed with the aim to be implemented over a computer cluster or a multi-core platform. The required features include a nearly absolute independence between the processes participating in the segmentation task and providing amount of work as equal as possible for all the participants. As a result, it is avoid many drawbacks often encountered when performing a parallelization of an algorithm that was constructed to operate in a sequential manner. Furthermore, the proposed algorithm based on the new segmentation model is efficient and shows a very good, nearly linear performance growth along with the growing number of processing units.

**Index Terms**— Image segmentation, Deformable Models, parallel processing

## 1. INTRODUCTION

Parallel architectures are becoming a de facto standard for handling complex operations that require significant computational power. In the last 20 years this platform has evolved from a low-availability, highly expensive solution reserved for a small group of specialists, into an accessible standard that is present in many home desktop computers. Taking into consideration the current progress in the field of computer performance, it is highly probable that this trend will be maintained. However, the possibilities of parallel architectures require a specific approach in order to be utilized effectively. Algorithms that have been designed to operate in a sequential manner are often very hard to modify into a parallel algorithm.

In a previous work [1], we have presented an early version of our image segmentation model, constructed using some assumptions of the Deformable Models (DM)

framework [2] and more specifically the Topological Active Volumes (TAV) model [3]. We have used several ideas of the original formulation of TAV as our foundation and introduced a number of improvements, including a major change in the organization of the model and in the way in which the segmentation is performed. In [4], we have introduced a new version of our model, called the Whole Mesh Deformation (WMD), working with a new schema for the shape deformation and with the ability to perform shape topology changes. Thanks to these features it is able to process very complex scenes with a dramatical increase in terms of performance.

In this paper, we present an extension of the WMD model for execution in parallel architectures. We perform experiments on a computer cluster equipped with 10 processing units and test the capabilities of the algorithm with a large 3D computed tomography scene.

This paper is organized as follows. Section 2 explains the structure of the Whole Mesh Deformation model for image segmentation. Section 3 describes the parallel implementation of the WMD model. Section 4 presents the experimental results and section 5 presents main conclusions.

## 2. THE WHOLE MESH DEFORMATION MODEL

Our model, similarly to the DM, is based on energy function optimization. The segmentation is initialized by instancing a 3-dimensional mesh, which would incorporate the images used as the input data. The said mesh would be constructed out of nodes evenly distributed through the input images, interconnected with each other using links. During segmentation it would be subjected to deformations by moving its nodes one by one until the whole structure would assume the shape of the objects present in the input images, thus segmenting the scene of interests. These deformations would be guided by the abovementioned energy function, which would guarantee the correct behavior of the mesh.

### 2.1. The Whole Mesh Deformation approach

The innovative schema for the mesh deformation that we have mentioned earlier lays in the following feature: in the

---

This work has been partially funded by Microsoft Portugal and by University of Beira Interior through the HIPERCAMBIO Project, by Fundação para a Ciência e a Tecnologia, Portugal, through the TRAMANET Project contract PTDC/EIA/73072/2006, and by Institute of Telecommunications, Portugal.

original TAV formulation the nodes of the mesh are divided in two categories. First of them, called the external nodes, includes only the group of nodes initialized on the edges of the volume. Those nodes would be responsible for detecting the objects of interest by progressing towards their borders and acting similarly as the Snakes method [2]. The remaining nodes are categorized as internal nodes and their function will be to model the internal structure of the objects. As it can be seen, the shape detection is performed by a group of nodes that is instanced very far from the objects of interest and in a relatively low number. In our opinion this leads to not optimal utilization of the available resources and limits the capabilities of the TAV method. That is why in our solution we change that approach and allow for all the nodes of the mesh to handle the boundary detection process. To achieve this we have formulated our own energy function with proper selection of its parameters, which is able to force the correct behavior on all the nodes of the mesh without dividing them into two categories. As a result the shapes of interests are segmented by the nodes that have been initialized very near to them, not the ones initialized on the borders of the scene.

## 2.2. Model description

The energy function is composed out of two parts, called the internal and external energy of the mesh. The former is responsible for maintaining regularity of the structure, the latter for applying the features of the input images. The internal energy is composed of two forces and it is defined as follows:

$$E_{int}(v) = \alpha \sum_{n=1}^k |\bar{m} - m_n| + \beta \sum_{n=1}^k |\arctan \mu_n - \frac{\pi}{2}| \quad (1)$$

The two sum expressions represent the mesh continuity and curvature forces. The parameter  $\bar{m}$  represents the average distance between all the nodes of the mesh and  $m_n$  is the average distance between the node  $n$  and its neighbors. The parameter  $\mu_n$  is the average angle between the node  $n$  and its neighbors. As we can see from (1) the continuity force attracts the nodes of the mesh to maintain equal distances between each other, whereas the curvature term attracts the nodes to keep a 90 degrees angle. The external energy is composed of three forces and is defined as follows:

$$E_{ext}(v) = \gamma \sum_{n=1}^k 1 - I(v_n) + \delta \sum_{n=1}^k 1 - G(v_n) + \varepsilon \sum_{n=1}^k E(v_n) \quad (2)$$

with  $I(v)$  representing the intensity values of the image,  $G(v)$  being the Gradient Vector Flow [5] values and  $E(v)$  values obtained by the Canny Edge Detector algorithm [6]. The symbols next to the sum expressions ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  and  $\varepsilon$ ) represent their weights and serve to balance their impact on the whole energy.

During the segmentation procedure the following is performed for each node  $N$  of the mesh: if the coordinates  $(x_n, y_n, z_n)$  describe the position of the node  $N$  at the time  $t$ , then for time  $t+l$  the coordinates of  $N$  would be described with  $(x_{n+l}, y_{n+l}, z_{n+l})$  where  $k$  and  $l \in \{-1, 0, 1\}$  and correspond to the lowest possible value of:

$$E(N_{x+k,y+l,z}) = E_{int}(N) + E_{ext}(N) \quad (3)$$

which is the energy of the node  $N$  calculated with (1) and (2) at these given coordinates. As it can be seen, the node is moved in its nearest neighborhood in  $X$  and  $Y$  planes and the position with the lowest energy is chosen as the new position of the node. After finding a new location for the node, the links between it and all its neighbors are checked with the following term:

$$|link(N_n, N_c)| \leq \frac{G_x + G_y}{2} * flex \quad (4)$$

where  $G_x$  and  $G_y$  represent the average distance between mesh nodes in  $x$  and  $y$  directions and  $flex$  is the flexibility parameter, defined upon segmentation execution. Whenever a certain link fails to satisfy (4) it is marked as broken, which means that it no longer will be considered when calculating the continuity term. As a result the nodes that were interconnected with it are allowed to progress away from each other without any restriction. This feature is the base for the topology changes in the mesh structure, namely to segment several objects in the scene or to create discontinuities in the mesh in order to successfully detect holes in the objects of interests.

The steps described above are repeated for each node of the mesh until the following rule is satisfied:

$$sum_n \leq \mu \quad (5)$$

where  $\mu$  is a small value near zero and the  $sum_n$  expression is defined as follows:

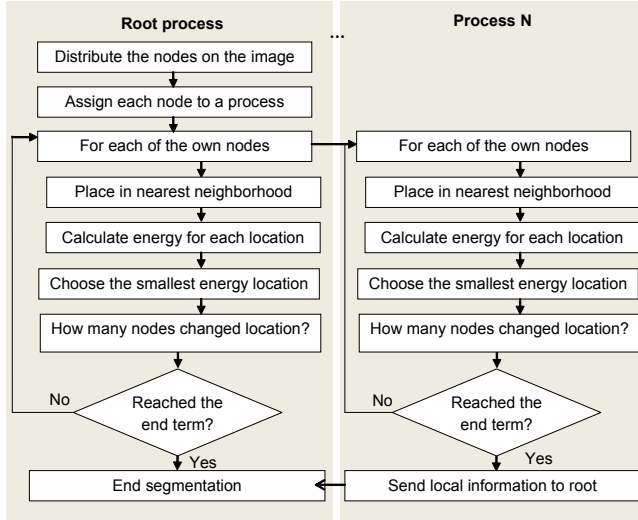
$$sum_n = \sum N: N(x, y, z, t) \neq N(x, y, z, t + 1) \quad (6)$$

The given term verifies the number of nodes that have changed their position in the last algorithm iteration. Whenever this number is decreased to zero (or very near to zero) the mesh is assumed to be in its stable position and the segmentation is finished.

As it can be seen, the segmentation procedure is handled by solving an optimization task using the energy function and the Greedy Algorithm [7] optimization method.

## 3. PARALLEL IMPLEMENTATION OF THE SEGMENTATION ALGORITHM

We have implemented a parallel image segmentation algorithm using our model as the foundation. It is able to take advantage of several processing units by dividing the segmentation task into equal part, which would be carried out by the said units simultaneously. In such a scenario the



**Figure 1.** Flowchart representing the parallelization of the algorithm.

workload distribution is performed in the following way: all the nodes of the mesh are divided in as many groups, as there are processing units available. Each of the units will have full access to the input data in order to allow full mesh deformation and avoid a situation when the given node is not able to move outside a specific region, because its processing unit does not hold any information about the necessary image data. Naturally this approach imposes the drawback of time required to propagate the input data between all the processing units, but our tests with computer cluster have shown that this time is not very significant in terms of the whole segmentation time. In a scenario with multi-core machine being the execution environment, this time would be even less considerable.

### 3.1. Model description

The general idea of algorithm parallelization is presented on Figure 1. Each of the processing units will create a temporary set of variables to use while the segmentation progresses. Those values will be representing the current state of the model on given processing unit and will not be shared nor synchronized with other participants. Naturally this is done to eliminate any dependencies between the processing units and avoid need for synchronization, which could severely increase the time required for the segmentation. The possibility of performing the optimization in an independent manner is guaranteed by the fact that most of the calculations carried out when minimizing the energy function are performed using only the local data. The only values that are defined as global for the entire model are the average distances and angles between the nodes of the mesh. However, experiments have shown that their values are modified very slightly during the entire segmentation process and thus there is no need to track their changes very

precisely – it is safe to assume that the locally calculated value describes well the state of the entire mesh.

The processing units perform the mesh optimization simultaneously using the steps described in Section 2.2 and test the equation (5) in order to decide when to conclude the segmentation.

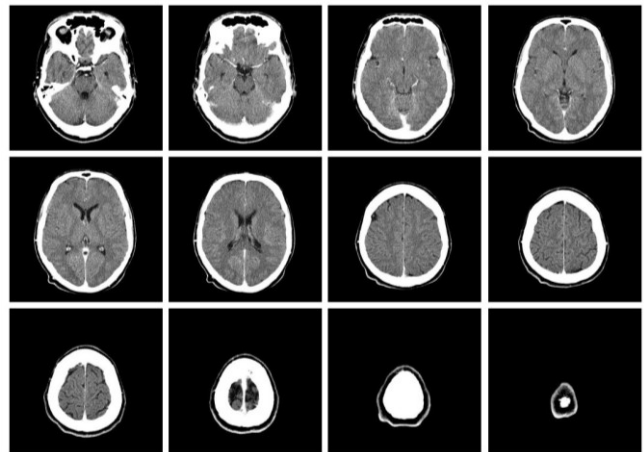
### 3.2. Suitability of the WMD model for parallelization

It is important to note that in the case of TAV the equal distribution of the workload would not be as straightforward as dividing the nodes in equally large groups. This possibility of our model is a result of the approach that we have taken, namely the lack of division between internal and external nodes. Thanks to this feature the full input image (or volume) is covered with nodes that are able to detect the boundaries of objects and thus the time required to perform segmentation with our algorithm depends very little on the contents of the input image. As a result, this time is very uniform for different parts of the mesh. On the other hand, in the case of TAV, the distance that the external nodes need to travel during the segmentation task depends highly on the contents of the image, which results in a big variation of the segmentation times for different parts of the mesh. This is the reason why our method is much more suitable for parallelization than TAV.

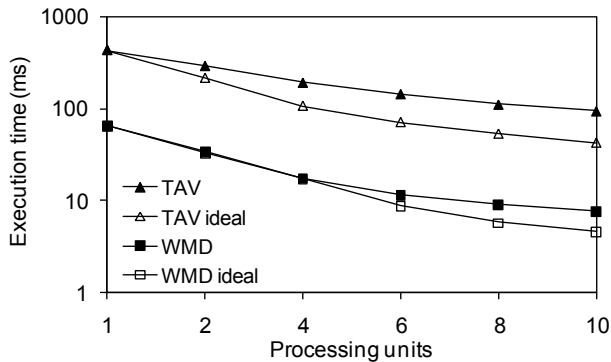
## 4. EXPERIMENTS AND RESULTS

The implementation of our algorithm has been performed using C# programming language along with the MS-MPI [8] message passing interface. The experiments have been executed on a computer cluster constructed from 10 processing units of 3GHz each, running the Microsoft HPC Server 2008 operating system. As the input image we have used a 12-slice CT brain scan presented on Figure 2, with each slice being a 512x512 image.

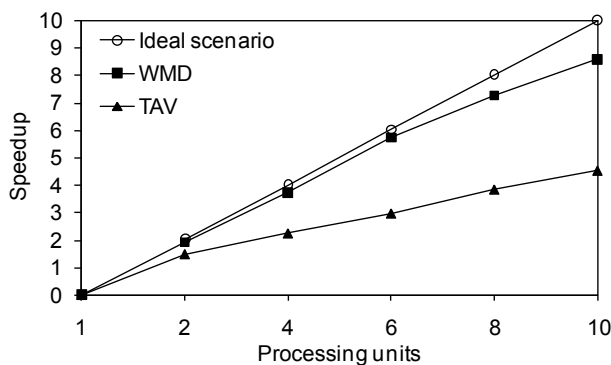
For comparison purposes we have implemented also a



**Figure 2.** Computed tomography images of a human brain used for the purposes of this paper.



**Figure 3.** Execution times of the two segmentation algorithms in milliseconds compared to ideal scenarios of their parallelization.



**Figure 4.** Speedup of the two segmentation algorithms when executed on different numbers of processing units, compared to the ideal scenario of parallelization.

segmentation algorithm that uses approach similar to the one of TAV, namely where the shape of the objects in the scene is detected using only one layer of external nodes. The times required to carry out the segmentation with the two approaches are presented on Figure 3. Furthermore, Figure 4 presents the performance gain obtained with the parallel implementation.

As it can be seen, our approach performs in significantly more efficient way when compared to the TAV approach. The execution times are smaller nearly by a level of magnitude in all the scenarios. This shows that our assumptions about the imperfections of the TAV have been correct and that we have managed to improve those features significantly. Figures 3 and 4 show that the parallelization of the algorithm is also performed efficiently and the resulting performance growth is very close to ideal in the case of the Whole Mesh Deformation model. By this we mean half the execution time for 2 processing units, quarter for 4 processing units and so on. However, in the case of TAV we can see a different behavior – the performance gain is more moderate. In the last two scenarios our model outperforms the TAV solution by more than a level of magnitude. This is a result of the high dependence of the TAV model on the contents of the image. As it can be seen on Figure 2, the last

slice of the volume contains very little data located far from the image borders. In TAV segmentation method the external nodes of the mesh have to proceed through nearly the entire image to reach the object of interests. As a result, the segmentation of the given slice is very time consuming and the entire segmentation process is prolonged – even if the remaining processing units carry out their tasks, they still need to ‘wait’ for this part to be finished.

## 6. CONCLUSIONS

In this paper, we have presented the Whole Mesh Deformation model, which is a 2D and 3D image segmentation model. It has been created with the aim of parallel image processing, which makes it highly suitable for implementing over architectures like computer clusters and multi-core platforms. This is achieved thanks to the approach that we have introduced, namely the lack of division between internal and external nodes of the mesh and treating all the nodes equally. This resulted in highly increased performance of the model and in far better suitability for workload distribution. Segmentation carried out with the algorithm based on our model significantly outperformed the TAV in all the tested scenarios and showed a much better utilization of resources when executed on a computer cluster.

## 7. REFERENCES

- [1] P. Lenkiewicz, M. Pereira, M. Freire, and J. Fernandes, "Accelerating 3D Medical Image Segmentation with High Performance Computing," in *Proc. 1st Workshops on Image Processing Theory, Tools and Applications, 2008 (IPTA 2008)*, pp. 1-8.
- [2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988.
- [3] N. Barreira, M. G. Penedo, C. Mariño, and F. M. Ansia, "Topological Active Volumes," in *Computer Analysis of Images and Patterns*. LNCS 2756, Springer, Berlin Heidelberg, 2003, pp. 337-344.
- [4] P. Lenkiewicz, M. Pereira, M. Freire, and J. Fernandes, "A New Approach to Medical Image Segmentation Based on Topological Active Volumes: The Whole Mesh Deformation Model," unpublished.
- [5] X. Chenyang and J. L. Prince, "Gradient Vector Flow: a New External Force for Snakes," in *Proc. Computer Vision and Pattern Recognition*, 1997, pp. 66-71.
- [6] J. F. Canny, "A Computational Approach to Edge Detection," in *Readings in computer vision: issues, problems, principles, and paradigms*: Morgan Kaufmann Publishers Inc., 1987, pp. 184-203.
- [7] D. Williams and M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," *CVGIP: Image Understanding*, vol. 55, pp. 14-26, 1992.
- [8] University of Indiana, "MPI.NET: High-Performance C# Library for Message Passing," 2009.