

THE WHOLE MESH DEFORMATION MODEL FOR 2D AND 3D IMAGE SEGMENTATION

Przemyslaw Lenkiewicz¹, Manuela Pereira¹, Mário Freire¹ and José Fernandes²

¹ IT-Networks & Multimedia Group, Department of Computer Science, University of Beira Interior
Covilhã, Portugal

² Microsoft Portugal, Porto Salvo, Portugal

ABSTRACT

In this paper we present a novel approach for image segmentation using Active Nets and Active Volumes. Those solutions are based on the Deformable Models, with slight difference in the method for describing the shapes of interests – instead of using a contour or a surface they represented the segmented objects with a mesh structure, which allows to describe not only the surface of the objects but also to model their interiors. This is obtained by dividing the nodes of the mesh in two categories, namely internal and external ones, which will be responsible for two different tasks. In our new approach we propose to negate this separation and use only one type of nodes. Using that assumption we manage to significantly shorten the time of segmentation while maintaining its quality.

Index Terms— image segmentation, deformable models, energy function

1. INTRODUCTION

The Deformable Models are a family of general-purpose image segmentation methods that have been receiving a significant share of research interest in the last two decades. Introduced by Kass [1] in 1988, it has been serving as a foundation for numerous image segmentation methods. One of them, called the Topological Active Volumes (TAVs) has been presented by Barreira and Penedo in [2] and further described in [3]. The TAVs have inherited many features of the Deformable Models but also have introduced some interesting changes to the original formulation. The most significant one was related to the way in which the objects of interests were perceived and presented. In case of the Deformable Models the segmentation was performed by placing a contour or a surface in the input image and deforming it until it segments the objects of interest. The TAVs have replaced this said contour with a 3-d mesh, which resulted in broadening the capabilities of the segmentation algorithm – not only the boundary of the object could be detected, but also its interior structure could be modeled. In this paper we present our model, developed with similar assumptions as in the case of TAVs, but with some major differences in the segmentation process.

The remainder of this paper is organized as follows: in Section 2 we describe the Topological Active Volumes and outpoint some of its features that we find disadvantageous. In Section 3 we introduce our model and in Section 4 we present and compare experimental results of the two approaches. In Section 5 we present our conclusions.

2. THE WHOLE MESH DEFORMATION MODEL

In the original idea by Barreira and Penedo the mesh was initialized over the whole volume with all the nodes divided into two groups: internal and external ones. The external nodes, which were in fact the ones that would perform the vital part of the segmentation process, were distributed only over the superficial layer of the volume. What this means is that they were initialized in position rather far from the objects of interests and they were in number significantly lower than the internal nodes. The segmentation process would require them to advance for large distances until they reach their optimal positions and finish their task.

We propose a different approach to the way in which the mesh deformation is handled. The segmentation process would be based on the following assumptions: firstly the mesh would be distributed over the entire volume. In the following steps the positions of the nodes would be optimized using the energy function. In contrast to the TAV method all nodes would be treated equally and the said energy calculation would be carried out using a single formula. This would in turn mean that the formula would have to be constructed in a way to impose the correct behavior on all the nodes of the mesh, namely to attract a part of them to the borders of the objects of interests and to distribute another part inside of the objects in a way that they would model their interior structure.

This would result in a significant change in the segmentation process, offering us advantages, like the possibility to exploit the boundary information from the entire volume, not just from the single layer surrounding it. Also, the distance that nodes would need to travel during the segmentation process would become much smaller and more predictable, as it would depend highly on the initial size of the grid and not on the contents of the image. Finally, the precision of the segmentation could increase because the complex shapes present in the input images would be

surrounded from all sides by nodes participating in the segmentation process and not only approached by them from a single side.

We have proposed two versions of our image segmentation model. The first version was simplified and destined mainly for the purposes of results' comparison with the original TAV algorithm. According to that version the mesh was deformed using only a single formula for the energy function, but the outer nodes of the mesh still needed to reach the boundaries of objects of interests in order to finish the segmentation process. This algorithm was presented in our previous work [4] and it has proven to give results of the same quality as the TAVs. It was however not capable of performing topological changes and thus it could only be tested in rather simple scenarios. In this paper we present the second version of the algorithm, in which we have eliminated the need of some nodes to travel for large distances to reach the object of interests and introduced the ability to perform topology changes.

2.1. Energy formulation

The energy function of our model consists of two types of forces: internal and external ones, which are respectively responsible for preserving the objects' structure and for applying the features of input images. The Active Volume is defined parametrically as follows: $v(r, s, t) = (x(r, s, t), y(r, s, t), z(r, s, t))$, where $(r, s, t) \in ([0, 1] \times [0, 1] \times [0, 1])$. The energy function is hereby defined as follows:

$$E(v) = \int_0^1 \int_0^1 \int_0^1 E_{int}(v(r, s, t)) + E_{ext}(v(r, s, t)) dr ds dt \quad (1)$$

where E_{int} and E_{ext} are the internal and the external energy of the mesh. In order to calculate the energy, the parameter domain $[0, 1] \times [0, 1] \times [0, 1]$ is discretized as a regular grid defined by the internode spacing [2]. Next, each of the contributing forces described below is calculated using the appropriate algorithm. The internal energy is composed of two forces and it is defined as follows:

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

The two sum expressions represent the mesh continuity and curvature forces. The parameter \bar{m} represents the average distance between all the nodes and m_n is the average distance between the node n and its neighbors. The parameter μ_n is the average angle between the node n and its neighbors. As we can see from (2) 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:

$$E_{ext}(v) = \gamma \sum_{n=1}^k (1 - I(v_n)) + \delta \sum_{n=1}^k (1 - G(v_n)) + \varepsilon \sum_{n=1}^k E(v_n) \quad (3)$$

with $I(v)$ representing the intensity values of the input images, $G(v)$ being the Gradient Vector Flow [5] values and $E(v)$ values obtained by using the Canny Edge Detector algorithm over the input images [6].

The symbols next to the sum expressions (α , β , γ , δ and ε) represent their weights and serve to balance their impact on the whole energy. To achieve the desired behavior of all the nodes in the mesh we have considered the following assumptions when choosing their values: the continuity and curvature terms need to be weighted with values big enough to maintain a high regularity of the mesh. The nodes that are initialized in the proximity of objects of interests need to be attracted to their borders and they should be the only ones attracted. That's why we need to give the GVF force a weight slightly higher than the continuity and curvature ones, but keep the range of GVF on a low level. Finally, the edge detector energy parameter needs to be set to the highest value, as it will serve as a 'pin' and fix the position of the nodes that reach the edges of objects of interests.

2.2. Segmentation procedure

Starting from a regular mesh initialized over the whole volume, we do the following for each node: it is placed in 9 positions corresponding to its current location and points of its nearest neighborhood. For each of those locations the node's energy is calculated using the energy function. The position corresponding to the lowest energy value is chosen as the new position of the given node. As we can see the optimization task is achieved using the greedy algorithm [7] approach. Next, the links between the given node and all its neighbors are checked with the flexibility term. This is a function that controls if the links do not stretch over certain limit, given by a parameter. This serves to detect links that should be broken in order to allow topology changes. Whenever two nodes spread apart for a distance larger than allowed by the flexibility parameter, the link between them is deleted and will no longer be considered when calculating the continuity term in the next iteration.

After repeating the above described process for all the nodes of the mesh, the decision whether to finish the optimization step is taken using the number of nodes that have changed their location in the concluded iteration – a large number of moved nodes means that the mesh is not yet close to its stable position. If the segmentation should not be terminated at given point, the average distance between all the nodes is calculated again and the algorithm returns to the step of optimization of the nodes' positions.

At the end of the optimization step we will still have some unwanted information present in the structure. These would sub-meshed constructed out of nodes that have been initialized outside the boundaries of objects of interests. At the end of the optimization step it would be important to locate and discard them from our results. We do it by calculating the following values for each sub-mesh: the standard deviation of the image intensity values and the

number of nodes finalized on points marked as edges of objects of interests. Assuming that the optimization step had been performed correctly, the unwanted sub-meshes will show a very low standard deviation of the image intensities and the number of nodes finalized on edges will be zero.

3. EXPERIMENTS AND RESULTS

We have performed our experiments using two models introduced in the previous section. One of them assumed that all nodes of the mesh should progress towards the objects of interests and the segmentation would finish only after reaching them, the other assumed that nodes initialized near the objects of interest should progress towards their boundaries and the ones located outside of the objects should be detached from the mesh and discarded at the end of the segmentation process. Both models included the ability to perform topological changes. As the input data we have used a set of two artificial 3-d volumes, one representing a vase and the other a teapot. The image slices constructing the volume were sized 300 by 300 pixels. The vase was built from 13 slices whereas the pot from 9 image slices. The algorithm was implemented in C#. The execution environment included a machine equipped with 2.5 GHz processor and 4GB of RAM.

Figure 1 presents the progression of segmentation of the pot shape. For presentation purposes we have chosen only one slice, as the whole volume presented on a small image would be unreadable. As it can be seen, the nodes of the mesh are attracted to the boundaries of object and the unwanted parts of the mesh gradually disconnect from the remainder. Figure 2 shows the number of iterations necessary to perform the full segmentation of volumes using the TAVs approach and the Whole Mesh Deformation (WMD) model. As we can see, our method shows a noteworthy improvement in performance. The execution times (in milliseconds) of those two scenarios are depicted on Figure 3. They are proportional to the number of iterations, however the vase required much more time to

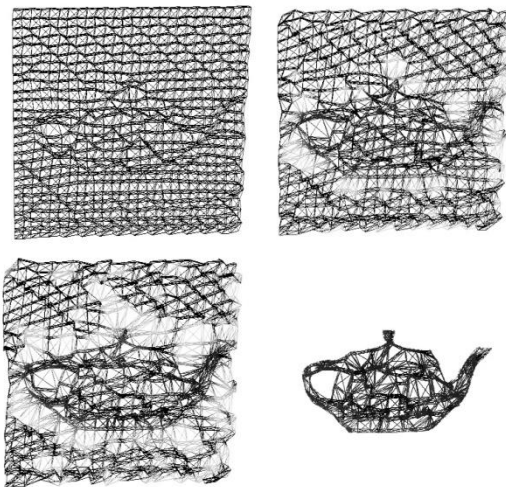


Figure 1. Consecutive steps of the pot shape segmentation.

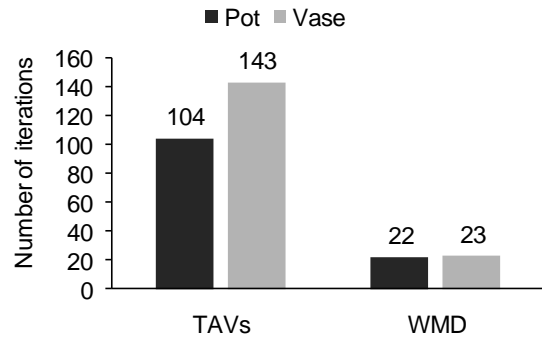


Figure 2. Number of iterations required for segmentation of the pot and vase shapes, using two described methods.

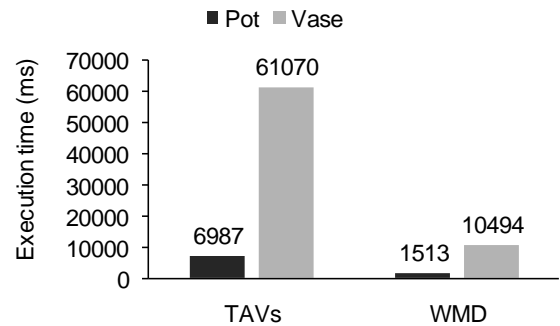


Figure 3. Execution times of the segmentation experiments.

complete the segmentation as it consisted of more slices.

The results that we have obtained follow our assumptions described in previous sections. The segmentation with Whole Mesh Deformation required always less time and less iterations than in the case of TAVs, giving the results of the same precision. We have also analyzed the number of iterations required to perform the segmentation of particular slices of the vase shape. The values of those are presented on Figure 4. It is visible that those values are not only much smaller for the WMD model, but also distributed much more evenly. This characteristic is also important due to the following: in every case of volume segmentation the required time would be determined by the portions of the volume that are the most time consuming. In other words,

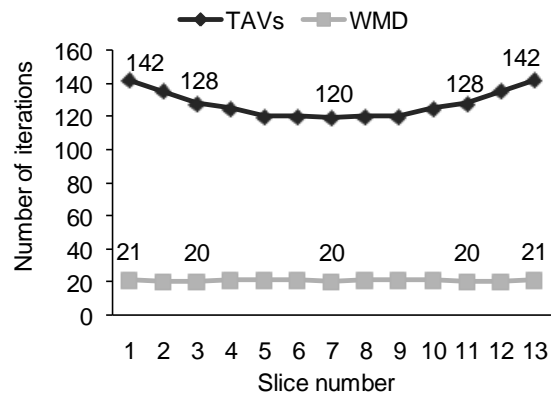


Figure 4. Number of iterations (y axis) required for segmentation of consecutive slices (x axis).

the segmentation would not be finished until all the nodes of the mesh approach their optimal locations. So in this particular case, even though around 120 iterations the most nodes are placed at their final positions, the segmentation still needs to progress until 143 iterations. This scenario does not take place if all the input slices require very similar time to be segmented, as in the case of Whole Mesh Deformation. To explain this characteristic of WMD model let us consider the following: the size of the mesh was 11 by 11 pixels, which results in 11 pixels of maximum possible distance needed for each node to travel, assuming that the segmentation is performed correctly and every edge would be approached only by the nearest nodes. The size of the two input images was 300 by 300 pixels, which results in mesh of the size 27 by 27 nodes, giving total number of 676 cells of the mesh for each slice. As we can see, the node's maximum travel distance can be estimated with high confidence without any information about the contents of the input images. Furthermore, the number of mesh cells is relatively high, which lets us assume with high probability that at least in one of the cells a node would have to travel the maximum possible distance. And since the whole segmentation time is defined by the node that has to move for the largest distance, we can assume that this pessimistic scenario is the factual scenario for majority of algorithm executions. Tracing the segmentation process iteration by iteration we have confirmed that the above assumption is correct. Through the initial 11 iterations the nodes are progressing towards the object borders and after that the unwanted connections between nodes are broken in order to perform topology changes and separate unwanted parts from the mesh. This part shows to be less predictable and is mainly responsible for the small differences in required iterations for different scenarios.

Finally, we have compared how the two approaches perform in scenarios of different sizes of the mesh. We have performed the segmentation using the pot shape and three different values of inter-node spacing, namely 16 by 16 pixels, 12 by 12 and 9 by 9 (in x and y axis). The results showed that the TAVs method requires exactly the same number of iterations to finish the segmentation regardless the density of the mesh. In turn, the number of iterations required by the Whole Mesh Deformation method is decreasing along with the growth of mesh density. Furthermore, the time required to carry out segmentation using the TAVs method is growing severely with the growth of mesh complexity, whereas in case of Whole Mesh Deformation we can notice only a very slight growth of the execution time. This is a much desired feature, as it allows performing precise segmentations of complex shapes, having almost no drawback of increased execution times. The explanation of this behavior lies in the fact that we have already mentioned, namely that with very small initial distances between mesh nodes the number of pixels that each node needs to travel is also decreasing, resulting in faster progression towards objects boundaries.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have presented our model for 3D image segmentation, which we called the Whole Mesh Deformation model. This method showed to be highly efficient, giving good segmentation results in short times. The segmentation was faster thanks to allocation of more nodes in boundary detection process. Also the whole procedure was more predictable and stable. Segmentation time was virtually independent from the size of the mesh.

In our future work we plan to further test the influence of different parameters of the energy function on the ability of the model to perform precise segmentation. We also plan to eliminate as many parameters from the energy formulation as possible and replace some of them with functions that would be able to detect automatically their appropriate values. Finally, we want to adapt the model to real-world segmentation problems, like medical image segmentation, and test its capabilities on very big volumes, which can be produced by the modern 3D acquisition devices.

5. ACKNOWLEDGMENT

This work has been partially funded by Microsoft Portugal and by the University of Beira Interior through the HIPERCAMBIO Project, and by the Institute of Telecommunications, Portugal.

5. REFERENCES

- [1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, 1988.
- [2] N. Barreira, M. G. Penedo, C. Mariño, and F. M. Ansia, "Topological Active Volumes," in *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, vol. 2756, Springer, Berlin Heidelberg, 2003, pp. 337-344.
- [3] N. Barreira, M. G. Penedo, C. Alonso, and J. Rouco, "Handling Topological Changes in the Topological Active Volumes Model," in *Image Analysis and Recognition*, Lecture Notes in Computer Science, vol. 5112, Springer, Berlin Heidelberg, 2008, pp. 122-131.
- [4] P. Lenkiewicz, M. Pereira, M. M. Freire, and J. Fernandes, "Accelerating 3D Medical Image Segmentation With High Performance Computing," in *Proc. International Workshops on Image Processing Theory, Tools & Applications*, Sousse, Tunisia, November 24-26, 2008, IEEE Press, New York, pp. 1-8.
- [5] X. Chenyang and J. L. Prince, "Gradient Vector Flow: a New External Force for Snakes," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 17-19 June, 1997, IEEE Computer Society, Los Alamitos CA, 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*, Academic Press, vol. 55, no. 1, pp. 14-26, 1992.