Speeding Up Active Mesh Segmentation by Local Termination of Nodes

Carl J. Nelson[1]
carl.nelson@dur.ac.uk
Martin Dixon[2]
martin.dixon@durham.ac.uk
P. Philippe Laissue[3]
plaissue@essex.ac.uk
Boguslaw Obara*[1]
boguslaw.obara@dur.ac.uk

[1] School of Engineering & Computer Science
Durham University
[2] School of Biological and Biomedical Sciences
Durham University
[3] School of Biological Sciences
University of Essex

Abstract

This article outlines a procedure for speeding up segmentation of images using active mesh systems. Active meshes and other deformable models are very popular in image segmentation due to their ability to capture weak or missing boundary information; however, where strong edges exist, computations are still done after mesh nodes have settled on the boundary. This can lead to extra computational time whilst the system continues to deform completed regions of the mesh. We propose a local termination procedure, reducing these unnecessary computations and speeding up segmentation time with minimal loss of quality.

1 Introduction

Deformable models [6] are a group of algorithms used to segment images, particularly images with weak or missing boundaries. Such deformable models, also known as active models and snakes, are used heavily in biomedical imaging segmentation where low contrast, high noise or obscured edges make other segmentation techniques unsuitable. Deformable models come in several forms and have been used for multiple modalities and scenarios (see [5] for a comprehensive review). Here we introduce and briefly describe the field limited to active meshes, three-dimensional polygonal-faced meshes that can be deformed in a discrete manner to segment objects of interest [7].

1.1 Active Meshes: a Quick Background

Active meshes are a subset of deformable models that commonly use a triangular-faced mesh to segment an object of interest from an image with weak or missing boundaries (see [4] for an early example). The use of a mesh with distinct nodes, or vertices, and a known connectivity, through face or edge information, allows computationally simple and speedy calculations to be performed over the whole mesh. To find the segmentation shape the mesh is deformed in one of several ways. The use of physics-based forces allows a new position to be calculated at each iteration by Euler-Lagrange mechanics [8]; however, there have been concerns

that optimality of such systems is not guaranteed [1]. A more common system is to provide each vertex with an energy and, for each iteration, identify some trial points for the vertex to move to; one of these points is then accepted by some optimisation method, the simplest of which is a greedy algorithm [10]. Many of these systems require a balloon force [3] or another force (e.g. [11]) to attract the shape towards the edges from any initialisation. As with many techniques each variant has its advantages and drawbacks and the optimal choice for a particular case is dependent on the experiment and researcher.

1.2 Termination of Meshes

Deformable models tend to work on an iterative system with each iteration deforming the model to minimise energies or balance forces, eventually the mesh reaches an optimal state and the desired object is segmented. With many optimisation techniques there may be no definite and/or optimal termination; as such, many systems are provided with an explicit stopping criterion in order to terminate the process. One key feature of termination criteria that are mentioned in the literature is that they focus on the overall mesh, i.e. they terminate in a global fashion, e.g. the percentage of 'inactive' vertices [8] or a maximum number of iterations [9]. We have been unable to find any articles in the literature that use a system of local vertex termination, that is to say, each vertex is deformed as part of the mesh until it reaches some independent termination criteria. In this article we propose a heuristic system for local vertex termination during deformation. We show that this method of termination can speed up the overall process of segmentation by testing the algorithm on synthetic data and also demonstrating the system on real 3D images.

2 Methods

For the purpose of this article we have used a simple balloon-driven, energy minimisation mesh (based on [2]). The mesh is initiated as an icosahedron of user-defined size. For each iteration, every vertex is assigned a new 'more optimal' position selected, by a simple greedy algorithm, from a 3D neighbourhood. A steepest gradient descent method is employed to link the image gradient to the size of the steps available. After each iteration of deformation the mesh undergoes local resampling, again based on [2]. During each iteration, all vertices are checked to see if they should be terminated (described fully below). Deformation continues until a termination criterion is met; this criterion will be described for each experiment in Results. All codes were written in MATLAB 2012b (The MathWorks, Inc., US) on a Windows 7 64-bit PC running an Intel Core i5-2320 CPU (3.0 GHz) with 6GB RAM.

2.1 Local Termination of Vertices

In order to terminate vertices we identify, at the end of each deformation, a list of vertices to be considered 'terminated'. During the next iteration we only deform vertices that are not terminated, decreasing computation time per iteration. Our system checks whether or not a vertex should be considered terminated after every iteration; this prevents a vertex being turned off due to, for example, a large noise signal, and being prevented from re-initiation as the mesh around it deforms. We define whether or not a vertex should be terminated using the following cost function,

$$C_v = \alpha C_I + \beta C_G + \gamma C_T, \tag{1}$$

where α, β and γ are user-defined parameters such that $\alpha + \beta + \gamma = 1$. The first two terms, C_I and C_G, are the value of the normalised image and gradient magnitude image,

respectively, calculated through trilinear interpolation. Finally, C_T is one if the vertex was terminated during the last iteration and zero otherwise. We note that more advanced cost functions could look at the history or neighbourhood of the vertex in more detail but we have, so far, only looked at basic, faster cost functions. In order to make this cost-based method more robust we use a combination of the cost at the vertex and the average cost of its direct neighbourhood. So T_v, the choice to terminate a vertex or not, is given by,

$$T_v = \begin{cases} \texttt{TRUE} & \text{if } \mu C_v + (1-\mu)\frac{\Sigma_j C_j}{N_v} > c, \\ \texttt{FALSE} & \text{otherwise;} \end{cases} \tag{2}$$

here, N_v is the size of the neighbourhood around the vertex, v, C_j is the cost at a neighbour, j, and μ is a user-defined parameter. The user-defined cost-level, c, is a threshold above which a point is terminated.

2.2 Measure of Quality

To determine segmentation 'quality', Q, of synthetic images we have used a simple metric comparing the segmented mesh to the binary 'ground truth' image, G. The resultant mesh is voxelised, B, and the measure of quality is calculated as $Q = B \cap G / B \cup G$. If the mesh extends outside of the ground truth the denominator increases and if the mesh has not fully extended to the ground truth boundaries then the numerator decreases. An ideal segmentation will always have a quality of one. Any segmentation that over or under segments will have a quality less than one; over segmentation and under segmentation in different regions cannot balance out using this metric.

3 Results

In order to clearly show when and where local termination may be effective for active mesh segmentation we have run experiments on five noise-free synthetic shapes. We also compare the results of globally terminated and locally terminated segmentation on real data of dehisced mature pollen grains.

3.1 Synthetic Data

We have chosen five synthetic shapes to represent various scenarios found in biological and biomedical images: a sphere, a cube, a hollow, rectangular cuboid, a tube and a weeble (a cone placed on top a hemisphere). We have chosen to demonstrate on noise-free data to show the differing effect of local termination on different shape scenarios, where we feel the difference is most significant. User-chosen parameters for local termination were used for each shape. The origin for segmentation was user-chosen based on manual selection of an approximate centroid. For all the control segmentations, i.e. segmentations using global termination, we terminated the procedure based upon the Haussdorf distance before and after each iteration. For each step the Haussdorf distance between the set of vertices before deformation and resampling and after were compared. If the Haussdorf distance was below a user-defined value then the shape was considered to be terminated.

Table 1 shows the time taken to segment and final segmentation quality for these shapes under global and local termination procedures. As can be seen there is a dramatic decrease in segmentation time with only a moderate loss of quality. We note that shapes that are fairly round or with a surface fairly evenly distributed from the point of initiation, i.e. the weeble and sphere, reap the smallest benefit in terms of time. The very slight increase in

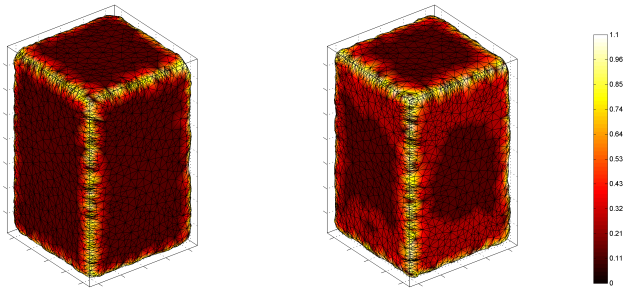
Shape (Vertices)	Global Termination		Local Termination		Reduction	
	Time(s)	Quality	Time(s)	Quality	Mean Time	Quality
Sphere (4,342)	186.1(0.1)	0.9564	70.0(0.4)	0.9595	62.4%	-0.3%
Cube (4,281)	336.0(0.7)	0.9241	69.8(0.2)	0.8730	79.2%	5.5%
Weeble (6,606)	251.4(0.6)	0.9589	118.2(0.5)	0.9647	53.0%	-0.6%
Tube (6,617)	265.5(0.7)	0.9580	40.2(0.07)	0.9287	84.9%	3.1%
Cuboid (3,694)	124.8(0.3)	0.9142	59.3(0.03)	0.8738	52.5%	4.4%

Table 1: Table showing reduction in time and quality of segmentation using local termination procedures. Times are taken from an arithmetic mean of ten full segmentations; the number in standard brackets is the standard error upon the mean to one significant figure. Number of vertices, taken from global termination results, are shown in brackets after shape names.

quality seen in some shapes is caused by the 'wiggle' of vertices across the surfaces due to the multidirectional nature of our method: as the Haussdorf distance measures the minimum distance between one vertex before deformation and its closest match after, it is conceivable that, as the mesh settles on the surface of the shape, vertices 'wiggle' in a Brownian motion-like manner across the surface, leading to resampling of particular regions of the surface, hence keeping the Haussdorf distance high. We should also note that we tracked the quality of the segmentation for each iteration (data not shown) and found that the maximum quality was not at the point of termination but a few iterations before termination due to the same reason. Figure 1 shows distance maps from the segmentation mesh of the hollow cuboid. The colouring at that area of the surface shows the average distance between that region of the mesh and the true boundary of the synthetic shape.

3.2 Noisy Data

In order to test our local termination data on images more similar to those found in biological and biomedical modalities we applied pseudo-random noise to our hollow cuboid image. Under these noisy conditions the local termination method lead to a decrease of 50 ($\pm$10)% in run time with only a 7.9 ($\pm$0.5)% decrease in quality compared to the global termination method. The average peak signal to noise ratio for the global termination trials (n=10) was 73.1685 ($\pm$0.0002) and for the local termination trials (n=10) was 73.1688 ($\pm$0.0003); peak signal to noise ratio was calculated as $\texttt{maximum}(I_s)/\texttt{variance}(N)$, where I_s is the noise-free image and N the noise applied to the image.



(a) (b)

Figure 1: Local Termination of Vertices Reduces Segmentation Time with a Small Reduction in Quality. (a) Distance difference map of segmentation using global termination. The segmented mesh is compared to a ground truth image and the colours represent the distance between the mesh and ground truth. (b) Distance difference map of local termination-based segmentation. The scale bar applies to both subfigures. Units are in voxels.

Local Termination...	Time(s)	Quality	Reduction from Global	
			Mean Time	Quality
with Random Benefit	54(2)	0.9737(0.008)	79.4%	-1.6%
with Double Take	49.54(0.08)	0.9733	81.3%	-1.6%

Table 2: Table showing the results of variations upon the basic cost-based local termination method. Note that the Quality of Local Termination with Random Benefit segmentation is stated as a mean (standard error) of ten repeats due to the use of a random value.

3.3 Improving Segmentation Quality

As shown above, local termination of active mesh segmentation can lead to a dramatic reduction in computing time; however, segmentation within biological and biomedical imaging requires as high a precision as possible. Here we show two slight modifications of our local termination scheme that help to improve the quality of segmentation. The results of these methods are shown in Table 2 for the synthetic tube; we have chosen this shape as there was the largest effect on reduction in time and a small decrease in quality. Tube-like structures are also heavily prevalent in biomedical images.

The first variation subtracts a pseudo-random value between 0 and 0.1 from the termination cost before comparison to c; this causes some vertices that should be terminated this iteration to be left on for at least one more iteration. As this is random it should lead to a slightly improved segmentation quality at the end result. With this method there is, however, a chance that no iteration will have all vertices terminated and, as such, a global termination criterion must be employed; for consistency we have chosen the Haussdorf distance as used above. In Table 2, this method is referred to as 'with Random Benefit'. The second method ignores the first termination of a vertex, i.e. a vertex must be classified as terminated for two iterations before it is actually terminated. The final termination criterion for this method is that when all vertices are terminated the segmentation is considered completed. In Table 2, this method is referred to as 'with Double Take'.

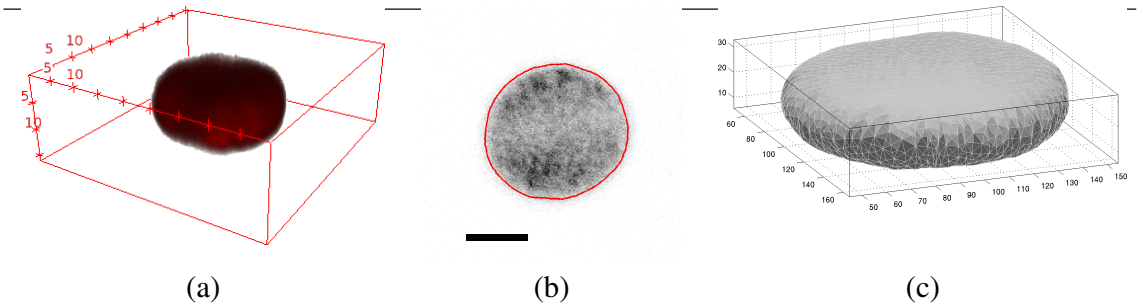
Table 2 shows that both variations keep the reduction in segmentation time low but improve the segmentation quality; in fact, both methods improve the quality beyond the original segmentation system. It is likely that this difference in quality is due to the particular stopping criterion for global termination used throughout this paper. As noted above the maximum quality for segmentation is often not the final quality when using the Haussdorf distance.

3.4 Biological Data

The proposed approach has been tested and validated on 3D CLSM images of DAPI-stained dehisced mature pollen grains obtained from wildtype *Arabidopsis thaliana* plants of the Columbia (Col-0) ecotype (Figure 2). On average, the proposed local termination approach has been able to speed up the overall process of pollen grains segmentation by 51%.

4 Conclusion

We have demonstrated the effect of local termination of vertices on the segmentation of shapes using active mesh systems. We have shown how the segmentation of long thin shapes is sped up dramatically by terminating those vertices that are 'complete' and no longer deforming the mesh in that region. We have chosen to show a basic cost function with some interesting improvements upon the basic idea; however, we are currently looking into more advanced ways of deciding whether or not a vertex is completed and, in general, the idea



(a) (b) (c)

Figure 2: Mature pollen grain: (a) 3D volume rendering. (b) 2D contour and (c) 3D mesh representation of the segmentation results. Scale bar corresponds to $10\mu m$.

of a cost function can be easily extrapolated for many imaging scenarios. Scenarios where this system will have the greatest effect are the segmentation of long, tubular structures, e.g. blood vessels, nerves or pathways such as the branchial network of the lungs, and by combining local termination with a topologically adaptive active mesh we could segment complicated tubular systems in a fraction of the time of the traditional methods. We are currently expanding application of the algorithm on biological and medical datasets.

References

[1] AA Amini, TE Weymouth, and RC Jain. Using dynamic programming for solving variational problems in vision. *Pattern Anal.*, 12(9):855–67, 1990.

[2] AJ Bulpitt and ND Efford. An efficient 3D deformable model with a self-optimising mesh. *Image Vis. Comput.*, 14:573–80, 1996.

[3] LD Cohen. On active contour models and balloons. *CVGIP*, 53(2):211–8, 1991.

[4] H Delingette. Adaptive and deformable models based on simplex meshes. *Proc. 1994 IEEE Work. Motion Non-rigid Articul. Objects*, pages 152–7, 1994.

[5] L He, Z Peng, B Everding, X Wang, CY Han, KL Weiss, and WG Wee. A comparative study of deformable contour methods on medical image segmentation. *Image Vis. Comput.*, 26(2):141–63, 2008.

[6] M Kass, A Witkin, and D Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vis.*, 331:321–31, 1988.

[7] D Molloy and PF Whelan. Active-meshes. *Pattern Recognit. Lett.*, 21:1071–80, 2000.

[8] J-Y Park, T McInerney, D Terzopoulos, and M-H Kim. A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. *Comput. Graph.*, 25(3):421–40, 2001.

[9] X Wang, L He, and WG Wee. Deformable Contour Method: A Constrained Optimization Approach. *Int. J. Comput. Vis.*, 59(1):87–108, 2004.

[10] DJ Williams and M Shah. A fast algorithm for active contours and curvature estimation. *CVGIP*, 1992.

[11] C Xu and JL Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, 7(3):359–69, 1998.