

Modification of the GPF method for efficient segmentation of high dimensional medical scans

Igor Sazonov*, Xianghua Xie* and Perumal Nithiarasu*

*Swansea University, Singleton Park, Swansea, U.K., SA2 8PP, i.sazonov@swansea.ac.uk

SUMMARY

Compared with standard segmentation methods, the GPF (geometric potential force) method proposed in [1] has advantages in processing low quality, noisy scans. But its main drawback is: it requires too much memory to allocate all necessary intermediate arrays. In this work, we propose a memory economic modification of the GPF method. Numerical studies show that the proposed method looks very promising for biomedical applications based on 3D and 4D scans.

Key Words: *image segmentation, active surface methods, level set methods.*

1 INTRODUCTION

Gradient based deformable modelling is a popular approach in medical image segmentation as it allows segmenting objects of complicated geometrical shape [2,3]. Convenient techniques suffer from weak edge, image noise and also convergence issues. Therefore, attempts to improve of the performance of these methods are described in numerous works on image segmentation.

In [1], Yeo *et al.* have proposed a deformable model that is based on a hypothetical non-physical interaction between the image gradient vector and active level set surface. It is shown that this method, called the geometrical potential force (GPF) method, is robust towards noise interference, weak edges, and exhibits invariant some convergence capabilities.

Let $I(\mathbf{x})$ be a given n -dimensional greyscale image defined in domain Ω with $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \Omega$ be the vector of coordinates of the image grid-points (voxel centres). The segmentation process of image I by the GPF method comprises two stages. At the first stage, the geometrical potential $G(\mathbf{x})$ is computed as a convolution of the image gradient $\nabla I(\mathbf{x})$ and the special kernel $\mathbf{K}(\mathbf{x})$:

$$G(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega} \nabla I(\mathbf{x}') \cdot \mathbf{K}(\mathbf{x} - \mathbf{x}'), \quad \mathbf{K}(\mathbf{x}) = \begin{cases} \mathbf{x} / \|\mathbf{x}\|^{n+1}, & \mathbf{x} \neq \mathbf{0} \\ \mathbf{0}, & \mathbf{x} = \mathbf{0} \end{cases} \quad (1)$$

where dot stands for the dot product. The image gradient ∇I is computed by central differences.

At the second stage, the once computed geometrical potential (GP) is used in the conventional PDE (e.g. [4]) with respect to the levels set function $\Phi(t, \mathbf{x})$

$$\partial \Phi / \partial t = \alpha g \kappa \|\nabla \Phi\| - (1 - \alpha) \mathbf{F} \nabla \Phi. \quad (2)$$

Here $g(\mathbf{x}) = 1 / (1 + \|\nabla I\|^2)$ is the stopping function; $\kappa(t, \mathbf{x}) = \nabla \hat{\mathbf{n}}$ is the curvature of isosurfaces of Φ ; $\hat{\mathbf{n}}(t, \mathbf{x})$ is the unit vector normal to isosurfaces of Φ ; $\mathbf{F}(t, \mathbf{x}) = G \hat{\mathbf{n}}$ is the GPF that acts as the external force; $\alpha \in [0, 1]$ is a weighting parameter. Deformable contour/surface/hypersurface is defined as $S(t) = \{\mathbf{x}, \Phi(t, \mathbf{x}) = 0\}$.

Direct calculation of the geometrical potential G in 3D and 4D is computationally expensive. A natural approach to calculate convolution (1) is to apply the fast Fourier transform (FFT). This approach described in [1] has a significant drawback: it requires too much of computer memory. We have to compute and store n components of the image gradient ∇I and twice more for the real and imaginary part of their Fourier image, also n components of the kernel \mathbf{K} and twice more for the Fourier image. For example, for a 3D image, it requires about 20 fold memory size of the initial image. Therefore becomes inefficient for a typical 3D scan size of 512^3 on a standard computers with, say, 4G memory. That's why a memory economic and computationally efficient method to evaluate the GP is so desirable.

In this work, we propose several modifications of computing algorithm for GP allowing us to decrease essentially the memory requirement. The proposed methods are valuated on both numerical examples and real world 3D data.

2 MATHEMATICAL DERIVATIONS

To start mathematical manipulating with Eq. (1) in order to obtain a more appropriate computational equations we write down the continuous analogue of (1)

$$G(\mathbf{x}) = \int \nabla I(\mathbf{x}') \cdot \mathbf{K}(\mathbf{x} - \mathbf{x}') d^n \mathbf{x}', \quad \mathbf{K}(\mathbf{x}) = P.V. \frac{\mathbf{x}}{\|\mathbf{x}\|^{n+1}}. \quad (3)$$

Here P.V. means that function $\mathbf{K}(\mathbf{x})$ should be regarded as a distribution (generalized function), and an integral with it should be considered in terms of the *principal value* (see, e.g. [5]).

One of the possible approaches to reduce memory usage is to derive an analytical formula for the kernel spatial spectrum and use it in computations. This is less time and memory consuming than computing the kernel in the x -space via (1) and after that evaluating its spectrum via the FFT. Performing the Fourier transform we can obtain

$$\tilde{\mathbf{K}}(\mathbf{k}) = \int \mathbf{K}(\mathbf{x}) e^{i\mathbf{k}\mathbf{x}} d^n \mathbf{x} = iB_n \frac{\mathbf{k}}{\|\mathbf{k}\|} \quad (4)$$

where $i = \sqrt{-1}$, factor B_n depends on the image dimension: $B_2 = \pi$, $B_3 = \pi^2$, $B_4 = \frac{4}{3}\pi^2$, etc.

Alternatively, we rearrange the integrand in (3) as a product of a scalar function and a scalar kernel instead of a dot product between vectors. Applying sort of integration by part we convert (3) into

$$G(\mathbf{x}) = \int I(\mathbf{x}') \nabla \mathbf{K}(\mathbf{x} - \mathbf{x}') d^3 \mathbf{x}' = \int I(\mathbf{x}') \mathcal{K}(\mathbf{x} - \mathbf{x}') d^3 \mathbf{x}'. \quad (5)$$

Thus, we only have to deal with the scalar kernel $\mathcal{K} = \nabla \mathbf{K}$. It is also a distribution but it has a higher order singularity than the initial kernel \mathbf{K} .

The both modifications can be combined together as the Fourier image of the scalar kernel can be easily calculated:

$$\tilde{\mathcal{K}} = i\mathbf{k}\tilde{\mathbf{K}} = -B_n \|\mathbf{k}\|. \quad (6)$$

To distinguish all these approaches we refer to the direct computation of the convolution in (1) as to method 0. Then we refer to the approach described in [1] and based on the use of the FFT for computation the convolution (1) as to method 1. The approach based on the use of formula (4): the analytical computation of the vector kernel, will be called method 2. The approach based on the use of scalar kernel (5) will be called method 3. The combined approach based on use of (5) and (6) will be called method 4.

To compute the GP with the use of method 4, we need to store only the real and imaginary parts of the image’s spatial spectrum. Then we can multiply them element-by element by the scalar kernel spectrum computed for every element of the arrays directly. So we have to allocate only 4 arrays of the same size as the initial image: the initial image, its spectrum (two arrays) and the output array—GP. The number of arithmetic operations is reduced as well.

The CPU time and memory requirement for computation GP of a 3D image size of 256^3 for all the methods is presented in the Table below.

	method 0	method 1	method 2	method 3	method 4
CPU time	~ 7days	91s	55s	42s	30s
Memory required	0.6G	1.8G	1.0G	0.6G	0.4G

The computation is made in Linux, Intel(R) Xeon 3.00GHz, RAM 4G. The higher the number of the method, the more effectively it can economize the memory. The 512^3 image (typical 3D medical scan) can be processed only by method 4: it requires 7 min of the CPU time and 3G of memory. Thus the numerical results confirm effectiveness of the algorithms.

3 REDUCING THE NOISE SENSITIVITY

GP computed on the base of methods 2,3,4 works well for segmentation of artificial smooth images without noise. But being applied to a noisy image or a real medical scan, they give GP with more oscillating behaviour as seen in Fig. 1(left) that decreases the quality of the segmentation. This is because the GP computed by methods 2,3,4 occurs to be more sensitive to the noise, especially to the delta-correlated noise. The matter is: deriving Eqs. (4)–(6) we neglect the finiteness of the domain and its discreteness to simplify the mathematical manipulation.

The analytical spectrum (4) is not decaying when $\|\mathbf{k}\|$ grows whereas the spectrum computed numerically does decay towards maximal $\|\mathbf{k}\|$ that makes method 1 to be less sensitive to the noise. To improve the performance of the method 2 we can multiply $\tilde{\mathbf{K}}$ by a decaying function $f(\mathbf{k})$ which plays role of a low-pass filter. An example of such function is

$$f(\mathbf{k}) = 1 - \|\mathbf{k}'\| + \frac{(\xi \|\mathbf{k}'\| - 1)^2}{(\xi + \xi \|\mathbf{k}'\| - 2) \xi}, \quad \mathbf{k}' = \left[\frac{k_1}{k_{1,\max}}, \frac{k_2}{k_{2,\max}}, \frac{k_3}{k_{3,\max}} \right]^T, \quad \xi = \max_{i=1,2,3} |k'_i|. \quad (7)$$

To improve performance of method 3 we can compute the vector kernel $\mathbf{K}(\mathbf{x})$ by Eq. (1) first and then to calculate its divergence $\mathcal{K} = \nabla \mathbf{K}$ numerically by the central differences.

In method 4, when computing ∇I , the central differences are substituted by derivatives computed through the FFT, i.e. through the multiplication by $i\mathbf{k}$ which makes the result more sensitive to the delta-correlated noise. Note that the central differences operator in the \mathbf{k} -space is equivalent to the multiplication by function \mathbf{g} :

$$\mathbf{g}(\mathbf{k}, \mathbf{h}) = \left[\frac{\sin k_1 h_1}{h_1}, \frac{\sin k_2 h_2}{h_2}, \dots, \frac{\sin k_n h_n}{h_n} \right]^T \quad (8)$$

where $\mathbf{h} = [h_1, h_2, \dots, h_n]^T$ is the vector of voxel sizes. Therefore the best way to compute $\tilde{\mathcal{K}}$ in method 4 is

$$\tilde{\mathcal{K}}(\mathbf{k}) = -B_n \frac{\mathbf{k} \cdot \mathbf{g}}{\|\mathbf{k}\|} f(\mathbf{k}). \quad (9)$$

Right plot in Fig. 1 indicates that the GP computed by methods 2,3,4 with the noise reduction correction is very close to that computed via methods 0,1. Example of blood vessel segmentation by the GPF approach when the GP is calculated by method 4 is shown in Fig. 2

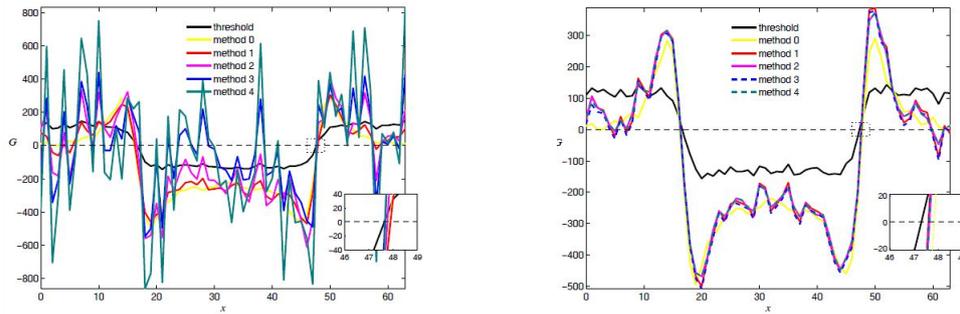


Figure 1: A 1D cut of a 3D GP: left—without the noise reduction correction, right—with it.

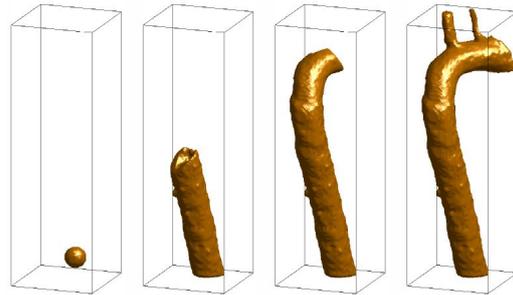


Figure 2: Stages of the automatic segmentation of thoracic aorta.

4 CONCLUSIONS

We proposed several modifications for algorithm of computation the geometrical potential in the GPF model and tested them. The approach which combines analytical kernel spectrum and scalar kernel conversion is the most computationally efficient and memory economic one. The methods were evaluated on 3D and 4D synthetic datasets, as well as 3D real world data. This preliminary work provided promising results which suggest that the proposed method has a great potential in efficient deformable modelling in high dimensional space without decomposing the space into a sequential order.

REFERENCES

- [1] S. Yeo, X. Xie, I. Sazonov and P. Nithiarasu, Geometrically Induced Force Interaction for Three-Dimensional Deformable Models, *IEEE Transactions on Image Processing*, 20(5), 1373–1387, 2011.
- [2] R. Malladi, J. A. Sethian, B. C. Vemuri, Shape modelling with front propagation: A level set approach, *IEEE Transactions on Image Processing*, 17(2), 158-175, 1995.
- [3] R. Whitaker, Modeling deformable surfaces with level sets. *IEEE Computer Graphics and App.*, 24(5), 6-9, 2004.
- [4] V. Caselles, R. Kimmel, G. Sapiro, Geodesic active contour, *International Journal of Computer Vision*, 22(1), 6179, 1997.
- [5] V. S. Vladimirov, *Methods of the Theory of Generalized Functions*. Taylor & Francis, 2002.