International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-3 Issue-1, October 2013

Level Set Based Image Segmentation Using Momentum and Resilient Propagation

Dhanraj Katta, G.Raghotham Reddy, R. Srikanth

Abstract*— In this paper image segmentation problems are solved by using the level set methods. Level Set Methods are involves to optimize the contour space and cost functional is minimized. Gradient descent methods are often used to solve this optimization problem since they are very easy to implement and applicable to general no convex functional. They are, however, sensitive to local minima and often display slow convergence. Traditionally, cost functional has been modified to avoid these problems. In this paper, I propose level set based image segmentation using momentum and resilient propagation. The proposed methods are very simple modifications of the basic method, and are directly compatible with any type of level set implementation. This approach consists of using the algorithmic core for processing images to detect parameter sensitivity is investigated.

***Index Terms*—Active contour, gradient methods, image segmentation, level set method, optimization.**

I. INTRODUCTION

One of the most important operations in Computer Vision is segmentation [1]. Segmentation is the task of recognizing objects in an image. The rest of the image is background. The aim of image segmentation is the domain independent partition of the image into a set of regions which are visually distinct and uniform with respect to low level information, such as gray level, texture or colour. The output of this is used as input in high level processing such as object recognition, scene analysis etc. Generally segmentation methods are based on two basic properties of the pixels in relation to their local neighborhood: discontinuity and similarity. Methods based on some discontinuity property of the pixels are called boundary based methods, whereas methods based on some similarity property are called region-based methods. Unfortunately, both techniques often fail to produce accurate segmentation [2]. Image segmentation is used in various applications. For all the applications, a single method cannot produce the produce desired results. It is all due to the image have different property and some ther factors also like noise, brightness etc. image segmentation simplifies and changes the representation of an image, i.e., the image is transferred into something that is more meaningful and easier to analyze. A very popular and powerful approach by level set method for solving image segmentation problems is through the calculus of variations. In this setting the solution is represented by a contour, which parameterizes an energy functional depending on various image based quantities such as intensities or gradients. In general, the set of possible contours constitutes the solution space, where the goal is to find the contour which extremizes the energy in this space. As an optimization problem, there are many possible strategies to find this solution. One approach is to use the method of graph cuts to find a global optimum. However, this can only be applied to a small class of energy functional. Along the same direction, this paper presents an alternative search strategy for the optimization solver .Where a simple, but effective, modification to gradient descent was proposed. The idea is to add momentum to the motion in solution space, allowing the search to avoid local optima and accelerate in favorable directions, and updating the step-sizes that gives acceptable convergence performance.

In this paper, we apply this to image segmentation in a variational framework using level set methods. The results show faster convergence and less sensitivity to local optima. The paper will proceed as follows. In section 2, we describe the idea of general level set method, section 3 presents Active contour. In Section 4, we describe the idea of gradient descent with momentum and resilient propagation in a general setting. Then, Section 5 presents how this idea can be used to solve segmentation problems in a level set framework. This is exemplified in Section 4 and Section 6 where we give implementation details and compute segmentations given a common energy functional. Finally, section 7 discussion Section 8 concludes the paper and presents ideas for future work.

I.LEVEL SET METHOD

The Level set method is a numerical technique for tracking interface & shapes, the advantages of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects(also called the Eulerian Approach) Level set methods make it very easy to follow shapes that change topology. This method is a great tool for modeling time varying objects, like inflations of a air bag or a group of oil floating in water.

$$\Gamma=\{(x,y)|\varphi(x,y)=0\}\text{---}(1)$$

The two dimensions that level set method amounts to representing a closed curve'Γ' using an auxiliary function φ called the level set function Γ is represented as the zero level set of φ by eq(1). If Γ value is +ve region delimited by the curve, if Γ value is -ve outside the curve.

$$\frac{\partial \varphi}{\partial t} = v|\nabla\varphi| \quad \text{--- (2)}$$

If the curve Γ moes in the normal direction with a speed 'v'then the level set function φ satisfies the level set eq(2). Segmentation should stop when the region of interest (ROI)in the application have been isolated.

Manuscript published on 30 October 2013.
***** Correspondence Author (s)
Dhanraj Katta*, M.Tech, MIETE Electronic & Communication Engineering, KITS, Warangal-15, AP, India.
G.Raghotham Reddy, M.Tech**,** MIEEE, MIETE, MISTE Assoc. Professor & Head Dept .of ECE, KITS Warangal-15, AP, India.
Rangu Srikanth, M.Tech, MIEEE, MIETE, Assistant Professor Dept of ECE KITS Warangal-15, AP, India.

Due this property of problem dependence autonomous segmentation is one of the most difficult tasks in image analysis. Noise and mixed pixels caused by the poor resolution of sensor image make the segmentation problem even more difficult.

II. ACTIVE CONTOUR

Early variational methods, such as *Snakes* and *Geodesic Active Contours* [1]–[4], often have boundary based terms such as an edge map. A parameterized curve, the active contour, is evolved according to the minimization of the cost functional until it converges to an equilibrium state representing the resulting segmentation. Later variational methods often include terms which are more region-based, allowing segmentation of objects without distinctive edges. These methods are often based on the Mumford-Shah segmentation model [5] where the image is partitioned into piecewise smooth regions with short boundaries. Chan-Vese used this model together with implicit contours represented by level sets [6]–[9]. This made the optimization problem easier and it naturally handles segmentation topology changes. In order to solve the optimization problem in level set segmentation methods, the gradient descent method is very often used. It deforms an initial contour in the steepest (gradient) descent of the energy. The equations of motion for the contour, and the corresponding energy gradients, are derived using the Euler-Lagrange equation [10] and the condition that the first variation of the energy functional should vanish at a (local) optimum. Then, the contour is evolved to convergence using these equations. The use of a gradient descent search commonly leads to problems with convergence to local optima and slow convergence in general. The problems are accentuated with noisy data or with a non-stationary imaging process, which may lead to varying contrasts for example. The problems may also be induced by bad initial conditions for certain applications. Traditionally, the energy functional have been modified to avoid these problems by, for example, adding regularizing terms to handle noise, rather than to analyze the performance of the applied optimization method. This is however discussed in [11], [12], where the metric defining the notion of steepest descent (gradient) has been studied. By changing the metric in the solution space, local optima due to noise are avoided in the search path.

III. GRADIENT DESCENT WITH MOMENTUM AND RPROP

Gradient descent is a very common optimization method which appeal lies in the combination of its generality and simplicity. It can be applied to all cost functions and the intuitive approach of the method makes it easy to implement. The method always moves in the negative direction of the gradient, locally minimizing the cost function. The steps of gradient descent are also easy to calculate since they only involve the first order derivatives of the cost function. Unfortunately, as discussed in Section I, gradient descent exhibits slow convergence and is sensitive to local optima. Other, more advanced, methods have been invented to deal with the weaknesses of gradient descent, e.g. the methods of conjugate gradient, Newton, Quasi-Newton etc, see for instance [31] for a review of continuous numerical optimization methods. To improve the convergence rate and the robustness against local optima for the gradient descent search, while avoiding the complexity of more sophisticated optimization algorithms, methods like gradient descent with *Momentum* [26] and *Rprop* [27] were proposed. The starting point of our derivation of the proposed methods is the following description of a standard line search optimization method:

$$\mathbf{x}_{k+1}=\mathbf{x}_k+\mathbf{s}_k \quad \text{----(3)}$$

$$\mathbf{s}_k=\alpha_k\,\hat{p}_k \quad \text{-----(4)}$$

where $\mathbf{x}_k$ is the current solution, $\mathbf{s}_k$is the next step consisting of length α_k and direction. To guarantee convergence, it is often required that $\hat{p}_k$ is a descent direction while $\alpha_k \geq 0$ gives a sufficient decrease in the cost function. A simple realization of this is gradient descent which moves in the steepest descent direction according to $\hat{p}_k = -\hat{\nabla} f_k$, where f is the cost function, while α_k satisfies the *Wolfe conditions* [31].

A. Momentum Method

Turning to gradient descent with Momentum, we will adopt some terminology from the machine learning community and choose a search vector according to:

$$\mathbf{s}_k=-\eta(1-\omega)\nabla f_k+\omega\mathbf{s}_{k-1} \quad \text{------(5)}$$

where η is the *learning rate* and $\omega \in [0, 1]$ is the *momentum*. Note that ω=0 gives standard gradient descent $\mathbf{s}_k=-\eta\nabla f_k$, while ω = 1 gives "infinite momentum" $\mathbf{s}_k= \mathbf{s}_{k-1}$. The intuition behind this strategy is that the current solution $\mathbf{x}_k$ has a momentum, which prohibits sudden changes in the velocity. This will effectively filter out high frequency changes in the cost function and allow for larger steps in favorable directions. Using appropriate parameters, the rate of convergence is increased while local optima may be overstepped.

B. Rprop Method

In standard implementations of steepest descent search, $\alpha_k=\alpha$ is a constant not adapting to the shape of the cost-surface. Therefore if we set it too small, the number of iterations needed to converge to a local optimum may be prohibitive .On the other hand, a too large value of α may lead to oscillations causing the search to fail. The optimal α does not only depend on the problem at hand, but varies along the cost-surface. In shallow regions of the surface a large α may be needed to obtain an acceptable convergence rate, but the same value may lead to disastrous oscillations in neighboring regions with larger gradients or in the presence of noise. In regions with very different behaviors along different dimensions it may be hard to find an α that gives acceptable convergence performance. The Resilient Propagation (Rprop) algorithm [27] was developed to overcome these inherent disadvantages of standard gradient descent using adaptive step-sizes k called *update-values*. There is one update-value per dimension in $\mathbf{x}$, i.e. dim(k)= dim($\mathbf{x}_k$). However, the defining feature of Rprop is that the size of the gradient is never used. Only the signs of the partial derivatives are considered in the update rule. Another advantage of Rprop, very important in practical use, is the robustness of its parameters; Rprop will work out-of-the-box in many applications using only the standard values of its parameters [32].We will now describe the Rprop algorithm briefly, but for implementation details of Rprop we refer to [28].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

For Rprop, we choose a search vector **sk** according to:

$$sk = -\textbf{sign}\,(\nabla fk) * \Delta k \quad \text{--------(6)}$$

where Δk is a vector containing the current update-values, a.k.a. *learning rates*, * denotes elementwise multiplication and **sign**$(\cdot)$ the elementwise sign function. The individual update value Δ_k^i for dimension i is calculated according to the rule:

$$\Delta_k^i = \begin{cases} \min\left(\left|\Delta_{k-1}^i\right|, \eta^+, \Delta_{\max}\right) & \nabla_{f_k}^i . \nabla_{f_{k-1}}^i > 0 \\ \max\left(\Delta_{k-1}^i, \eta^-, \Delta_{\min}\right) & : \nabla_{f_k}^i . \nabla_{f_{k-1}}^i < 0 \\ \Delta_{k-1}^i, \quad \cdots & : \nabla_{f_k}^i . \nabla_{f_{k-1}}^i = 0 \end{cases}$$

Where Δ_k^i denotes the partial derivative i in the gradient. Note that this is Rprop without backtracking as described in [28]. The update rule will accelerate the update-value with a factor $\eta+$ when consecutive partial derivatives have the same sign, decelerate with the factor η if not. This will allow for larger steps in favorable directions, causing the rate of convergence to be increased while decreasing the risk of convergence to local optima. The convergence rate behavior of Rprop is illustrated in Fig. 2 when run on the examples in Fig. 1 as before. In both examples the Rprop method succeeds in reaching the global minimum neighborhood within ten terations. It passes the area with small gradients quickly since it only uses the signs of the gradient components, not the magnitude. In areas with smooth gradients it will in addition accelerate the step lengths. This adaptively makes it less sensitive to the initial step length which can also be seen in the example.

II. ENERGY OPTIMIZATION FOR SEGMENTATION

The calculus of variations where an energy functional is defined representing the objective of the problem. The extreme to the functional are found using the Euler-Lagrange equation [10] which is used to derive equations of motion, and the corresponding energy gradients, for the contour [33]. Using these gradients, a gradient descent search in contour space is performed to find a solution to the segmentation problem. Consider, for instance, the derivation of the *weighted region* (see [33]) described by the following functional:

$$f(C) = \iint_{\Omega c} g(x,y)\,dx\,dy \text{----(8)}$$

where C is a 1D curve embedded in a 2D space, C is the region inside of C, and $g(x, y)$ is a scalar function. This functional is used to maximize some quantity given by $g(x, y)$ inside C. If $g(x , y) = 1$ for instance, the area will be maximized. Calculating the first variation of Eq.3 yields the evolution equation

$$\frac{\partial c}{\partial t} = -g(x,y)n \text{-------- (9)}$$

where $\boldsymbol{n}$ is the curve normal. Using $g(x, y) = 1$ gives the commonly known “balloon force,” which is a constant flow in the (negative) normal direction. The contour is often implicitly represented by the zero level of a time dependent signed distance function, known as the level set function.The *level set method* was introduced by Osher and Sethian [6] and includes the advantages of being parameter free, implicit and topologically adaptive. Formally, a contour C is described by

$$C = \{\mathbf{x} : \varphi(\mathbf{x}, t) = 0\}.$$

The contour C is evolved in time using a set of partial differential equations (PDEs). A motion equation for a parameterize curve eq(1)and eq(2). .

Using Momentum and Rprop for Minimizing Level Set Flows

We have noted that evolving the contour according to the Euler-Lagrange equation yields a gradient descent search. Recall that each contour can be represented as a point inthe solution space. Thus, we can approximate the direction of the gradient by computing the vector between two subsequent points. In the level set framework we achieve this by taking the difference between two subsequent time instances of the level set function, representing the entire level set function as one vector, $\varphi(t_n)$: $\nabla f(tn) \approx \dfrac{\varphi(t_n) - \varphi(t_n - 1)}{t}$ ------(10)

where $\varphi(t)$ is the level set function corresponding to the image, $t = t_n - t_n - 1$ and ∇f is the gradient of a cost function f with respect to φ. We can now present the update procedures for the Momentum and Rprop methods. Note that the required modifications to the standard gradient search algorithm are very simple and are directly compatible with any type of level set implementation. This makes it very easy to test and evaluate the proposals in your current implementation.

1) Level Set Updates: For the Momentum method, we follow the ideas from Section II-A and incorporate a momentum term in the update of the level set function:

$$s(t_n) = -\eta(1-\omega)\frac{\phi(t_n) - \phi(t_n - 1)}{\Delta t} + \omega s(t_{n-1}) \text{ -------- (11)}$$

$$\phi(t_n) = \phi(t_{n-1}) + s(t_n) \qquad \text{------ (12)}$$

For the Rprop method, we can just use Eq. 13 instead of Eq. 11. In Eq. 13 we use the update values estimated by Rprop as described in Section II-B

$$s(t_n) = -sign\left(\frac{\phi(t_n) - \phi(t_n - 1)}{\Delta t}\right) * \Delta(t_n) \qquad \text{----(13)}$$

where *, as before, denotes elementwise multiplication and **sign**$(\cdot)$ the elementwise sign function.

The complete procedure works as follows:

1) Given the level set function $\boldsymbol{\varphi}(tn-1)$, compute the next (intermediate) time step $\boldsymbol{\varphi}(tn)$. This is performed by evolving $\boldsymbol{\varphi}$ according to a PDE (such as Eq. 9) using standard techniques (e.g. Euler integration).
2) Compute the approximate gradient by Eq. 10.
3) Compute a step $\boldsymbol{s}(tn)$ according to Eq. 11 or Eq. 13 for the Momentum and Rprop method respectively.
4) Compute the next time step $\boldsymbol{\varphi}(tn)$ by Eq. 12. Note that this replaces the intermediate level set function computes step1 .

IV. METHOD

We will now evaluate the methods by solving four example segmentation tasks. We use both synthetic and real datasets in 2D and 3D. They will hereafter be referred to as *MRI scanned images of brain, hand, leg, and VascuSynth.*

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The data sets have ground truth segmentations which are used to evaluate the performance of the methods.

$s_k = -\eta\nabla f_k + \sigma e^{-\tau k}\xi_k$ (14)

where σ and τ are the noise level and time decay parameter respectively. ξ_k is a standard multivariate Gaussian noise term. Note that $\sigma = 0$ gives the standard gradient descent. The purpose of the time decaying noise in SGD is to avoid early local minima. Eq. 14 is also called Langevin updating and is effectively a simulation of an annealed diffusion process [35].

A. Weighted Region Based Flow

In order to test and evaluate our methods, we have used a simple energy functional to control the segmentation. It is based on a weighted region term (Eq. 7) combined with a penalty on curve length for regularization. The goal is to maximize:

$$f(c) = \iint_{\Omega c} g(x,y)\,dx\,dy - \alpha \oint_c ds \text{ ----- (15)}$$

where α is a regularization parameter controlling the penalty of the curve length. The target function *g(x, y)* can be based on intensity values alone, e.g. $g(x, y) = I(x, y) - T$ where $I(x, y)$ is the image intensity and T is a constant threshold intensity for the segmented objects. This choice will result in a regularized thresholding for $\alpha > 0$. However, objects in many real images are not robustly segmented by one threshold parameter, so our experiments use a function *g(x, y)* which is based on filtering methods. These filters (hereafter denoted as the *target filters*) detect and output positive values on the inside of line structures, negative values on the outside, and zero on the edges. We refer the reader to [36] for more information on the filter parameters and [37] for details on how to generate the filters.

For 3D datasets, the surface and line integrals in Eq. 15 are translated to volume and surface integrals over a 2D surface embedded in a 3D space. Irrespective of dimensionality, a level set PDE can be derived from Eq. 15 (see [33]):

$$\frac{\partial \phi}{\partial t} = -g(x)\left|\nabla \phi\right| + \alpha k \left|\nabla \phi\right| \text{ --------- (16)}$$

where κ is the curvature of the curve in 2D, and the mean curvature of the surface in 3D.

B. Implementation Details

We have implemented Rprop in Matlab as described in [28]. The level set algorithm has also been implemented in Matlab based on [9], [38]. Some notable implementation details are: 1) Any explicit or implicit time integration scheme can be used in Step 1. Due to its simplicity, we have used explicit Euler integration which might require several inner iterations in Step 1 to advance the level set function by _t time units. 2) In practice we want to apply the momentum and Rprop on the gradient of the *shape* of the contour, rather than on the level set gradient. The latter is in general not related to the shape, since the contour is merely represented as the zero level set. However, if the level set function is confined to a signed distance function, there is a mapping between the contour and the level set function, making the connection between the shape gradient and level set gradient more clear. To enforce this mapping, we reinitialize φ after Step 1 and Step 4, using standard methods such as fast marching [39] or fast sweeping [40]. 3) The parameters $\eta+$ and $\eta-$ of Rprop have been fixed to their default values ($\eta+ = 1.2$ and $\eta- = 0.5$ respectively) during all experiments.

C. Experiments

The parameters for all four experiments are given in Table I. The narrow band parameter *N* controls the size of the region close to the zero level set where calculations are performed [38]. This is a computational optimization which is valuable especially in 3D. In theory, this parameter should not affect the result of the propagation. In practice however, the narrow band might provide a "barrier" for the contour, especially for large learning rates, which effectively stops very large steps. This motivates including the narrow band parameter in the experiments. The "curvature weight" parameter, α in Eq. 16, controls the cost of having long contours. It is the balance between the target function *g()* gain and the curvature cost that creates the local optima, see Eq. 16. The last parameter specifies the convergence criterion, i.e. the segmentation has converged when the infinity norm of ∇f in Eq. 10 is less

IV. EXPERIMENTAL RESULTS

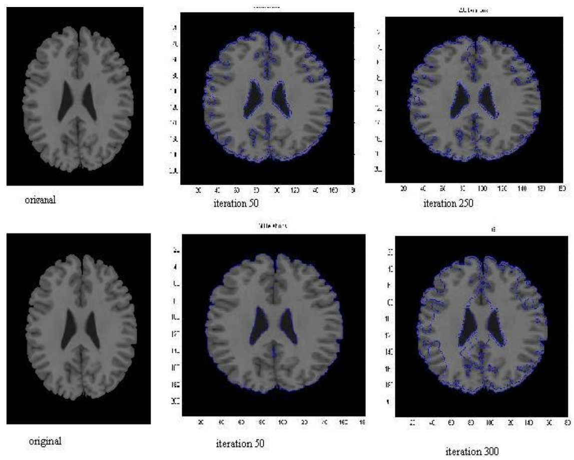

fig 1-with momentum method fig-2. with Resilient propragation

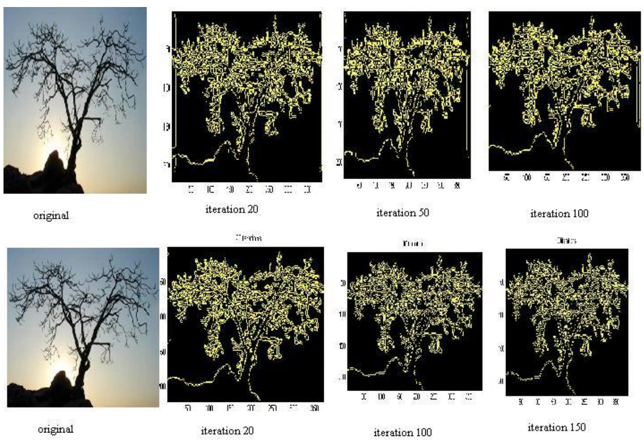

fig 3- with momentum method & fig 4 - with Resilient propagation

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

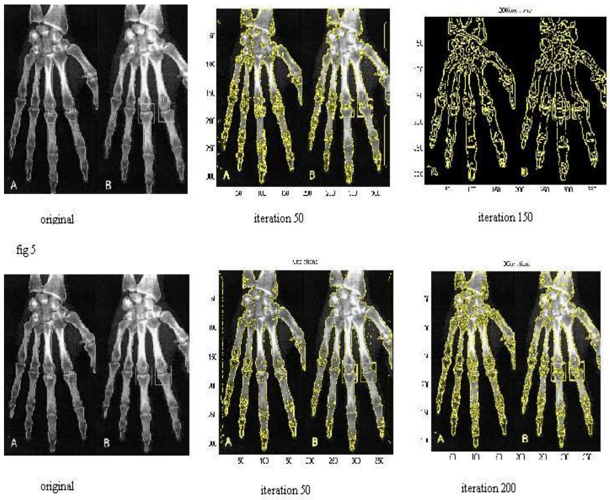

fig 5- with momnmtum method fig 6- with Resilient Propagation

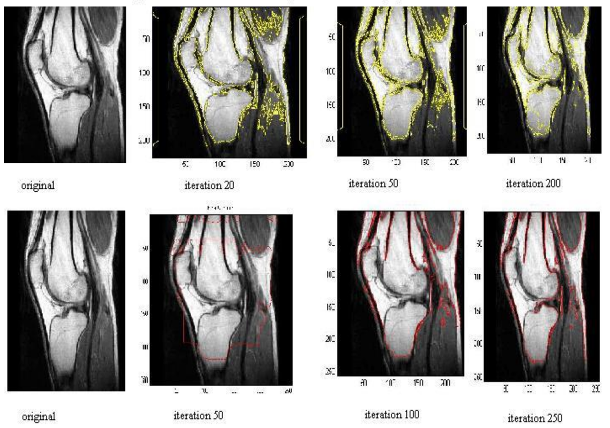

fig 7-with Momentum method fig 8- With Resilient propagation

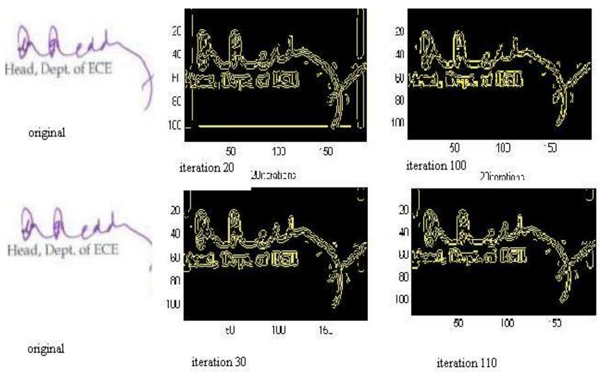

fig 9 With momentum method fig 10. with Resilient Propagation

The following above images are experimental

1) Original images
2) Momentum method
3) Resilient Propagation

By using the following parameters the shown in box

images parameters	Brain	Tree	Hand	Leg	Sign
Iteration	250/300	100/150	150/200	200/250	100/110
ω	0.2-0.8	0.2-0.9	0.1-0.5	0.2-0.7	0.3-0.8
Δt	10	5	10	5	5
η	1,2,4,8,16	1,2,3,4,5,6	0.5,1,1.5,2.3	0.5,1,1.5,2	0.5,1,1.5,2,2.5

But the enhanced method solved this problem better. On the similar computing proposal, under a 3.0GHz Pentium iv PC with 1 GB RAM on board, the average processing time of improved method was 9.6s, and that was 30.3s with Li's method. The evolution time was greatly reduced

V. DISCUSSION

This paper to compare different optimization methods given a certain inputs, the quality and accuracy of the resulting segmentation may not be comparable to the reference segmentations. The point of these experimental segmentation is to highlight the advantages of momentum and Rprop methods in contrast to the ordinary gradient.

VI. CONCLUSION

Image segmentation using the level set method involves optimization in contour space. In this context, gradient descent is the standard optimization method. We have discussed the weaknesses of this method and proposed using the Momentum and Rprop methods, very simple modifications of gradient descent, commonly used in the machine learning community. The modifications are directly compatible to any type of level set implementation, and downloadable reference code with examples is available online. In addition, we have shown in a series of experiments how the solutions are improved by these methods. Using Momentum and Rprop, the optimization gets less sensitive to local optima and the convergence rate is improved. Rprop in particular is also shown to be very insensitive to parameter settings and to different gradient behaviors. This is very important in practical use since Rprop will work out-of the-box in many applications using only the standard values of its parameters. In contrast to much of the previous work, we have improved the solutions by changing the method of solving the optimization problem rather than modifying the energy functional

REFERENCES

[1] R. C. Gonzalez, R. E. Woods,"Digital Image processing " 2ndPrentice-Hall Inc, 2002.

[2] L. G. Shapiro and G. C. Stockman, " Computer Vision, " Prentice-Hall Inc., New Jersey, 2001, pp. 279-325

[3] Thord Anderson,Gunnar Lathen,Reiner lenz,and Magnus Borga,member of IEEE"Modified Gradient Search for level set Based image segmentation "

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," Int. J. Comput. Vis., vol. 1, no. 4, pp. 321–331, 1988.

[5] L. D. Cohen, "On active contour models and balloons," CVGIP, Image Understand., vol. 53, no. 2, pp. 211–218, Mar. 1991.

[6] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," inProc. Int. Conf. Comp. Vis., Jun. 1995, pp. 810–815.

[7] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in Proc. IEEE Int. Conf. Comput. Vis., Jun. 1995, pp. 694–699.

[8] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," Commun. Pure Appl Math., vol. 42, no. 5, pp. 577–685, 1989.

[9] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," J. Comput. Phys., vol. 79, no. 1, pp. 12–49, Nov. 1988.

[10] T. Chan and L. Vese, "A level set algorithm for minimizing the Mumford–Shah functional in image processing," in Proc. IEEE Workshop Variat. Level Set Meth. Comput. Vis., Mar. 2001, pp. 161–168.

[11] T. F. Chan and L. A. Vese, "Active contours without edges," IEEE Trans Image Process., vol. 10, no. 2, pp. 266–277, Feb. 2001.

[12] S. Osher and R. Fedkiw,Level Set Dynamic Implicit Surfaces. New York: Springer-Verlag, 2003.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

[13] P. M. Morse and H. Feshbach, “The variational integral and the Euler equations,” in Proc. Meth. Theor. Phys., I, May 1953, pp. 276–280.
[14] G. Charpiat, R. Keriven, J.-P. Pons, and O. Faugeras, “Designing spatially coherent minimizing flows for variational problems based on active contours,” in Proc. IEEE Int. Conf. Comput. Vis., vol. 2. Oct. 2005, pp. 1403–1408.
[15] G. Sundaramoorthi, A. Yezzi, and A. Mennucci, “Sobolev active contours,” Int. J. Comput. Vis., vol. 73, no. 3, pp. 345–366, 2007.
[16] R. B. Kearfott, Rigorous Global Search: Continuous Problems (Non-convex Optimization and Its Applications), vol. 13. Dordrecht, TheNetherlands: Kluwer, 1996.
[17] S.Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671–680, 1983.
[18] [N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” J. Chem. Phys., vol. 21, no. 6, pp. 1087–1092, 1953.
[19] D. J. Wales and J. P. K. Doye, “Global optimization by basin-hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms,” J. Phys. Chem. A, vol. 101, no. 28, pp. 5111–5116, 1997.
[20] Y. N. Law, H. K. Lee, and A. Yip, “A multiresolution stochastic level set method for Mumford–Shah image segmentation,” IEEE Trans. Image Process., vol. 17, no. 12, pp. 2289–2300, Dec. 2008.
[21] A. Chambolle, “An algorithm for total variation minimization and applications,” J. Math. Imag. Vis., vol. 20, nos. 1–2, pp. 89–97, 2004.
[22] T. F. Chan, S. Esedoglu, and M. Nikolova, “Algorithms for finding global minimizers of image segmentation and denoising models,” SIAM J. Appl.Math., vol. 66, no. 5, pp. 1632–1648, 2006.
[23] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, “Fast global minimization of the active contour/snake model,” J. Math.Imag. Vis., vol. 28, no. 2, pp. 151–167, Jun. 2007.
[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning Internal Representations Error Propagation. Cambridge, MA: MIT Press, 1986,ch. 8, pp. 318–362.
[25] M. Riedmiller and H. Braun, “Rprop, description and implementation details,” Inst. für Logik, Univ. Karlsruhe, Karlsruhe, Germany, Tech.
[26] G. Läthén, T. Andersson, R. Lenz, and M.Borga,“Momentum based optimization methods for level set segmentation,” in Proc. Int. Conf.Scale Space Variat. Meth. Comput. Vis., LNCS 5567.Jun. 2009, pp.124–136.
[27] T. Andersson, G. Läthén, R. Lenz, and M. Borga, “A fast optimization method for level set segmentation,” in Proc. Scandinavian Conf. Image Anal., LNCS 5575. Jun. 2009, pp. as 400–409.
[28] T. Andersson, G. Läthén, R. Lenz, and M. Borga, “A fast optimization method for level set segmentation,” in Proc. Scandinavian Conf. mage Anal., LNCS 5575. Jun. 2009, pp. 400–409.
[29] J. Nocedal and S. J. Wright, Numerical Optimization, 2nd ed. New York:Springer-Verlag, 2006.
[30] W. Schiffmann, M. Joost, and R. Werner, “Comparison of optimized backpropagation algorithms,” in Proc. Eur. Symp. Artif. Neural etw., 1993, pp. 97–104.
[31] R. Kimmel, “Fast edge integration,” in Geometric Level Set Methods in Imaging, Vision and Graphics. New York: Springer-Verlag, 2003.
[32] W. A. Gardner, “Learning characteristics of stochastic gradient descent algorithms: A general study, analysis, and critique,” Signal Process., vol. 6, no. 2, pp. 113–133, Apr. 1984.
[33] T. Rögnvaldsson, “On Langevin updating in multilayer perceptrons,” Neural Comput., vol. 6, no. 5, pp. 916–926, 1993.
[34] G. Läthén, J. Jonasson, and M. Borga, “Blood vessel segmentation using multi-scale quadrature filtering,” Pattern Recognit. Lett., vol. 31, no. 8, pp. 762–767, Jun. 2010.
[35] G. H. Granlund and H. Knutsson, Signal Processing for Computer Vision. Norwood, MA: Kluwer, 1995.
[36] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, “A PDE based fast local level set method,” J. Comput. Phys.,vol.155, no. 2, pp.410–438, 1999.
[37] J. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, pp. 1591–1595, Feb. 1996.
[38] H.-K. Zhao, “A fast sweeping method for Eikonal equations,” Math.Comput., vol. 74, pp. 603–627, May 2005.
[39] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken “Ridge based vessel segmentation in color images of the retina,” IEEE Trans. Med. Imag., vol. 23, no. 4, pp. 501–509, Apr. 2004.
[40] P. Jassi and G. Hamarneh, “Vascusynth: Vascular tree synthesis software,” Insight J., pp. 1–12, Jan.–Jun. 2011.
[41] D. Green and J. Swets, Signal Detection Theory and Psychophysics. New York: Wiley, 1966.
[42] W. J. Youden, “Index for rating diagnostic tests,” Cancer, vol. 3, no. 1,pp. 32–35, 1950.
[43] S. Chen, G. Charpiat, and R. J. Radke, “Converting level set gradients to shape gradients,” in Proc. 11th Eur. Conf. Comp. Vis., 2010, pp. 715-728

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)