Momentum Based Optimization Methods for Level Set Segmentation

Gunnar Läthén[1,3], Thord Andersson[2,3], Reiner Lenz[1,3], and Magnus Borga[2,3]

[1] Department of Science and Technology, Linköping University
[2] Department of Biomedical Engineering, Linköping University
[3] Center for Medical Image Science and Visualization, Linköping University

Abstract. Segmentation of images is often posed as a variational problem. As such, it is solved by formulating an energy functional depending on a contour and other image derived terms. The solution of the segmentation problem is the contour which extremizes this functional. The standard way of solving this optimization problem is by gradient descent search in the solution space, which typically suffers from many unwanted local optima and poor convergence. Classically, these problems have been circumvented by modifying the energy functional. In contrast, the focus of this paper is on alternative methods for optimization. Inspired by ideas from the machine learning community, we propose segmentation based on gradient descent with momentum. Our results show that typical models hampered by local optima solutions can be further improved by this approach. We illustrate the performance improvements using the level set framework.

1 Introduction

A very popular and powerful approach 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 [1]. However, this can only be applied to a small class of energy functionals. For more general problems, the standard method has been to deform an initial contour in the steepest (gradient) descent of the energy. Equations of motion for the contour is derived using the Euler-Lagrange equation and the condition that the first variation of the energy functional should vanish at a (local) optimum. Then, the contour is evolved to steady-state given the resulting equations. A standard implementation of this strategy is usually hampered by two common problems. The first problem is sensitivity to local optima, which are manifested due to noisy data. To avoid this, the usual approach has been to modify the energy functional by adding regularizing terms. The second common problem is poor convergence due

X.-C. Tai et al. (Eds.): SSVM 2009, LNCS 5567, pp. 124–136, 2009.

to difficulties in choosing good initial conditions. To improve convergence, very effective solvers based on multi-grid [2, 3] and AOS schemes [4, 5, 6] have been developed. However, these methods all search for a solution in the gradient descent direction and little focus has been given to the underlying optimization problem. This has been identified in recent work [7, 8], 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.

Along the same direction, this paper presents an alternative search strategy for the optimization solver. Our idea stems from the machine learning community, where an optimization problem is solved to update the system to adapt to a given stimulus. A simple, but effective, modification to gradient descent was proposed in [9], which basically adds a momentum to the motion in solution space. This simulates the physical properties of inertia and momentum and effectively allows the search to avoid local optima and accelerate in favorable directions. In this paper, we show how this idea can be used for 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 gradient descent with momentum in a general setting and give examples highlighting the benefits. Then, Section 3 presents how this idea can be used to solve segmentation problems in a level set framework. This is exemplified in Section 4 and Section 5 where we give implementation details and compute segmentations given a common energy functional. Finally, Section 6 concludes the paper and presents ideas for future work.

2 Gradient Descent with Momentum

Considering general optimization problems, gradient descent is a very simple approach which can handle many types of cost functions. It is intuitive, since it always moves in the direction of steepest descent, which locally gives the largest amount of decrease in the cost function. In addition, it only requires first order derivatives of the function, providing simple and fast computations. On the other hand, it is well known that gradient descent suffers from poor convergence and high sensitivity to local optima for many practical problems. Therefore, other descent directions (Newton, Quasi-Newton, etc.) have been studied and proved superior, see e.g. [10] for a rigorous reference.

A simple alternative to these, more theoretically sophisticated methods, is often applied in the machine learning community. A typical problem here is the construction of adaptive systems that can classify unknown inputs. This can be formulated as an optimization problem and one of the goals of machine learning is to construct fast learning or adaptation rules that can be implemented in very simple hardware or software devices. To improve the convergence and robustness of a simple gradient descent solution, while avoiding the complexity of more sophisticated optimization methods, gradient descent *with momentum* was proposed [9]. The starting point of our derivation of the proposed method is the following description of a standard line search optimization method:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k \tag{1}$$

$$\boldsymbol{s}_k = \alpha_k \boldsymbol{p}_k \tag{2}$$

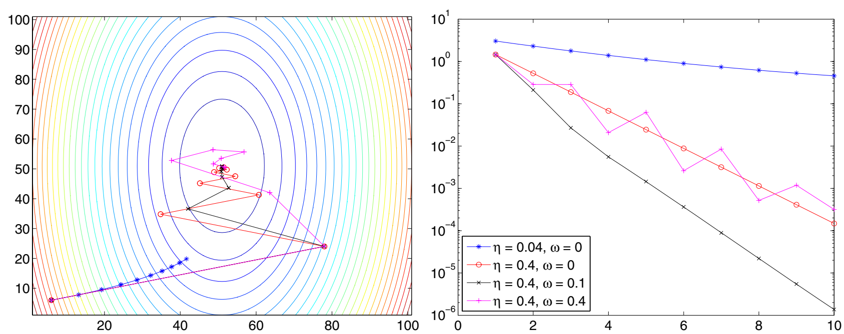
where $\boldsymbol{x}_k$ is the current iterate, $\boldsymbol{s}_k$ is the next step consisting of length α_k in direction $\boldsymbol{p}_k$. To guarantee convergence, it is often required that $\boldsymbol{p}_k$ be a descent direction while α_k gives a *sufficient decrease* in the cost function. A simple realization of this is gradient descent which moves in the steepest descent direction according to $\boldsymbol{p}_k = -\boldsymbol{\nabla} f_k$, where f is the cost function, while α_k satisfies the *Wolfe conditions* [10].

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

$$\boldsymbol{s}_k = -\eta(1-\omega)\boldsymbol{\nabla} f_k + \omega \boldsymbol{s}_{k-1} \tag{3}$$

where η is the *learning rate* and $\omega \in [0, 1]$ is the *momentum*. Note that $\omega = 0$ gives standard gradient descent $\boldsymbol{s}_k = -\eta\boldsymbol{\nabla} f_k$, while $\omega = 1$ gives "infinite inertia" $\boldsymbol{s}_k = \boldsymbol{s}_{k-1}$. The intuition behind this strategy is that the current iterate has an *inertia*, which prohibits sudden changes in the velocity. This will effectively filter out high frequency changes in the cost function and allow for greater steps in favourable directions. Selecting appropriate parameters, our hope is that the rate of convergence is increased while eventual local optima will be overstepped.

The effect of the momentum term is illustrated in Figure 1. The iterates with momentum $\omega = 0$ show the behaviour of standard gradient descent when varying the learning rate (step length) η. In comparison, for an appropriate choice of momentum $\omega = 0.1$, the solution approaches the optimum more rapidly. It can be seen however, that too high momentum of $\omega = 0.4$ leeds to oscillations.



(a) Iterates and cost function (b) Convergence rate

Fig. 1. Gradient descent search with/without momentum on a quadratic cost function

3 Minimizing Level Set Flows

As was previously outlined, segmentation problems in image analysis are often described as optimization problems with solutions derived using the calculus of variations. The standard procedure is to formulate an energy functional by means of a contour and various image derived terms. Extremals of this functional are then identified by an Euler-Lagrange equation, which is used to derive equations of motion for the contour [11]. This typical procedure yields a gradient descent search in a high dimensional solution space, in which each possible contour is represented by a point. For example [11] presents, among others, the derivation of *weighted region* described by the following functional:

$$E(C) = \iint_{\Omega_C} f(x,y)dxdy \tag{4}$$

where C is a 1D curve embedded in a 2D domain, Ω_C is the inside region of C, and $f(x,y)$ is a scalar function. This functional is used to maximize some quantity given by $f(x,y)$ inside C. A simple example is $f(x,y)=1$ which measures, and maximizes, the area. Calculating the first variation of Eq. (4) yields the evolution equation:

$$\frac{\partial C}{\partial t} = -f(x,y)\boldsymbol{n} \tag{5}$$

where $\boldsymbol{n}$ is the curve normal. Again, setting $f(x,y)=1$ gives a constant flow in the normal direction, typically referred to as the "balloon force".

The representation, or parameterization, of the contour C can in general be chosen arbitrarily. However, it is often convenient to use the implicit level set method by Osher and Sethian [12], since this allows for arbitrary topological changes. To summarize the basic ideas, a contour is represented *implicitly* as a zero level set of a time dependent scalar function (referred to as the level set function). Formally, a contour C is described by $C = \{\boldsymbol{x} : \phi(\boldsymbol{x},t) = 0\}$. To deform C, the level set function is evolved in time according to a set of partial differential equations (PDEs). The transition from the equations of motion for a parametrized curve (Eq. (5)) to a level set PDE is accomplished by a simple procedure. In general, the motion $\frac{\partial C}{\partial t} = \gamma\boldsymbol{n}$ translates to the level set equation $\frac{\partial \phi}{\partial t} = \gamma\,|\boldsymbol{\nabla}\phi|$ [11]. Thus, Eq. (5) gives the familiar level set equation:

$$\frac{\partial \phi}{\partial t} = -f(x,y)\,|\boldsymbol{\nabla}\phi| \tag{6}$$

The remainder of this section will describe how we modify the typical level set method update scheme to incorporate a momentum term as presented in Section 2.

3.1 Momentum for Minimizing Level Set Flows

We have noted that the contour evolving according to the Euler-Lagrange equation yields a gradient descent search. Recall that each contour can be represented as a point in the solution space (the structure of the space will depend on parameterization). 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:

$$\nabla f(t_n) \approx \frac{\phi(t_n) - \phi(t_{n-1})}{\Delta t} \tag{7}$$

where f is a cost function in compliance with the terminology used in Section 2. Note that this is indeed an approximation, depending on the time difference $\Delta t = t_n - t_{n-1}$. Following the ideas from Section 2, we update the level set function to incorporate a momentum term:

$$s(t_n) = -\eta(1-\omega)\frac{\widetilde{\phi}(t_n) - \phi(t_{n-1})}{\Delta t} + \omega s(t_{n-1}) \tag{8}$$

$$\phi(t_n) = \phi(t_{n-1}) + \Delta t s(t_n) \tag{9}$$

The complete procedure works as follows:

Procedure `UpdateLevelset`

1. Given the level set function $\phi(t_{n-1})$, compute the next (intermediate) time step $\widetilde{\phi}(t_n)$. This is performed by evolving ϕ according to a PDE (such as Eq. (6)) using standard techniques (e.g. Euler integration).
2. Compute the approximate gradient by Eq. (7).
3. Compute a step $s(t_n)$ according to Eq. (8). This step effectively modifies the gradient direction by incorporating the momentum term as a fraction of the previous step $s(t_{n-1})$.
4. Compute the next time step $\phi(t_n)$ by Eq. (9). Note that this replaces the intermediate level set function computed in Step 1.

The procedure is very simple and is directly compatible with any type of level set implementation.

4 Experiments

We now describe some details of the implementation and illustrate properties of the suggested method using two examples. Here we study 1D curves embedded in a 2D domain, but the approach readily generalizes to 2D surfaces in 3D given the level set framework.

4.1 Implementation Details

We have implemented the proposed ideas in Matlab using standard level set techniques based on [13, 14]. Reference code can be found online at the site `http://dmforge.itn.liu.se/ssvm09/`. Some details of our implementation are the following:

- The level set function is reinitialized (reset to a signed distance function) after Step 1 and Step 4. This is typically performed using the fast marching [15] or fast sweeping algorithms [16]. There are two reasons for this: Firstly it is required for stable evolution in time due to the use of explicit Euler integration. Secondly we want a momentum induced by the zero level set of ϕ (the contour), rather than *all* level sets of ϕ. Reinitialization could be omitted, with the effect of introducing a momentum on all individual level sets. Interpreting each sample of ϕ as a parameter of the contour, this is equivalent to applying momentum on each parameter. While feasible, we have not experimented with momentum without incorporating reinitialization.
- We avoid instabilities by dampening $s(t_n)$ in Step 3 using a sigmoidal function:

$$\hat{s}(s(t_n), s_{max}) = \frac{2s_{max}}{1 + e^{-2s(t_n)/s_{max}}} - s_{max} \tag{10}$$

  where s_{max} is the maximum step length allowed.
- 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 Δt time units.

4.2 Weighted Region Based Flow

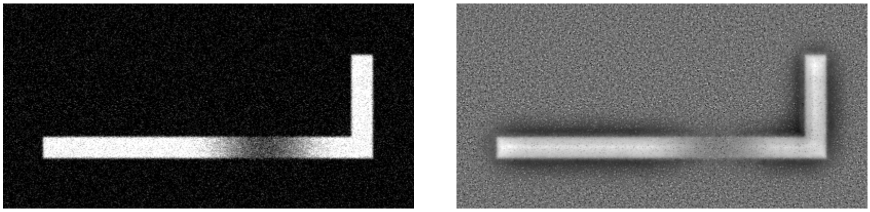
To verify our idea, we have used a simple energy functional based on a weighted region term (Eq. (4)) combined with a penalty on curve length for regularization. The goal is to maximize:

$$E(C) = \iint_{\Omega_C} f(x, y) dxdy - \alpha \oint_C ds \tag{11}$$

where α is a regularization weight parameter. The target function $f(x, y)$ is image based, computed using the approach in [17]. This method uses quadrature filters [18] across multiple scales to detect line structures. Taking the real part of the complex filter response, $f(x, y)$ gives positive values on the inside of linear structures, negative on the outside, and zero on the edges. Translating Eq. (11) to a level set PDE following [11] gives:

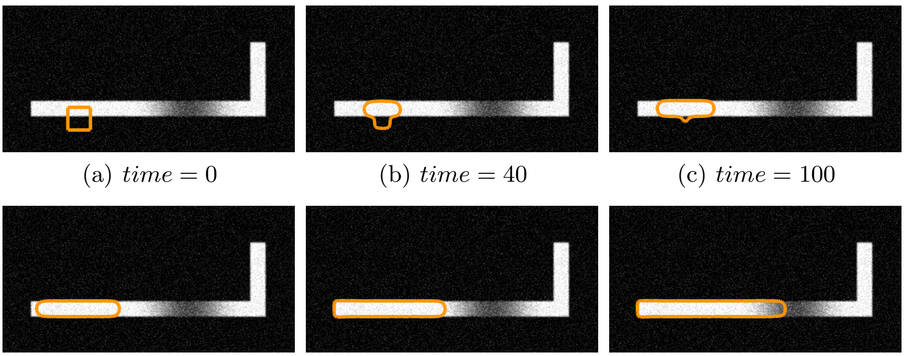
$$\frac{\partial \phi}{\partial t} = -f(x, y) \left|\nabla \phi\right| + \alpha\kappa \left|\nabla \phi\right| \tag{12}$$

where κ is the curvature of the contour. First we illustrate some properties of the method with a synthetic test image depicted in Figure 2(a), which mimics the common problem of intensity variation in medical imaging. The intensity of the object ranges from 0.3 to 1, while the noise level is 0.1. This image yields the target function $f(x, y)$ in Figure 2(b) where bright and dark colors indicate positive and negative values respectively. As exemplified in our first experiment (Figure 3) the dip in contrast results in a local optimum in the solution space.



(a) Input image (b) Target function $f(x, y)$

Fig. 2. Synthetic test image illustrating the presence of a local optima in the solution space



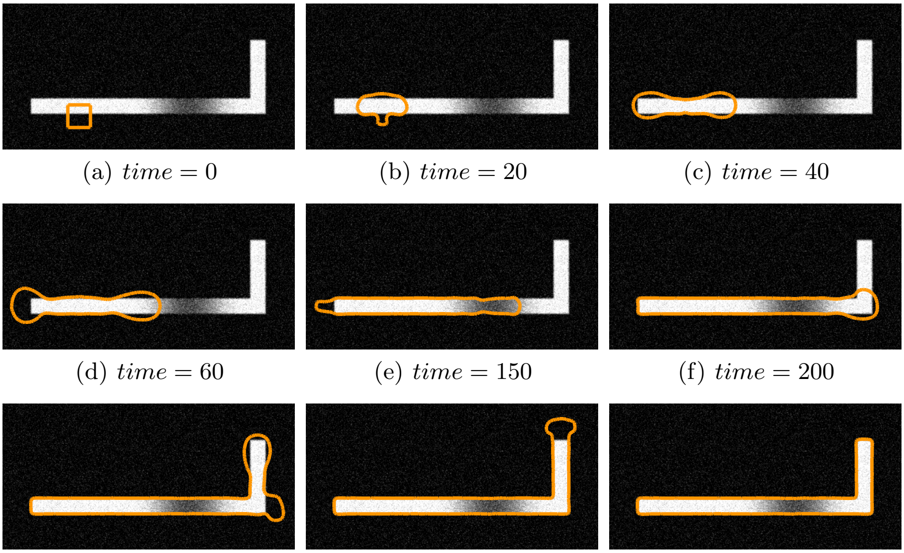
(a) $time = 0$ (b) $time = 40$ (c) $time = 100$

(d) $time = 170$ (e) $time = 300$ (f) $time = 870$

Fig. 3. Iterations without momentum (conventional gradient descent)

Figure 3 shows the results after evolving the level set function by Eq. (12) until convergence *without* momentum, using conventional methods. We define convergence as $|\nabla f|_\infty < 0.03$ (using the infinity/maximum norm), with ∇f given in Eq. (7). For this experiment we used parameters $\alpha = 0.7$ and we reinitialized the level set function every fifth time unit. For comparison, Figure 4 shows the results after running our method using parameters $\alpha = 0.7, \omega = 0.8, \eta = 10, s_{max} = 100, \Delta t = 5$. Plots of the energy functional for both experiments are shown in Figure 5. Here, we plot the weighted area term and the length penalty term separately, to illustrate the balance between the two. Note that the functional without momentum in Figure 5(a) is monotonically increasing, due to the nature of gradient descent, while the functional with momentum visits a number of local maxima during the search.

To further exemplify the behaviour of our method, we created a slightly modified version of Figure 2(a), shown in Figure 6(a). In contrast to Figure 2(a), the shape in Figure 6(a) is disconnected, so the global optimum is expected to contain two separated regions. Not surprisingly, conventional gradient descent captures only a local minimum as displayed in Figure 7, while gradient descent with momentum succeeds in capturing the global solution as two separated

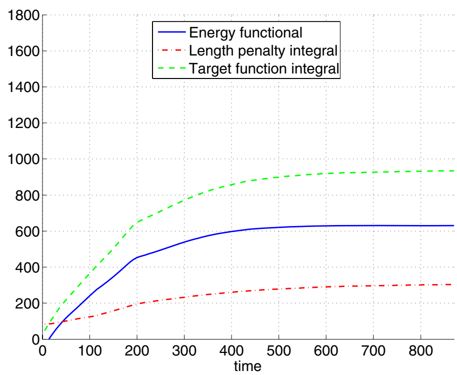


(a) $time = 0$ (b) $time = 20$ (c) $time = 40$

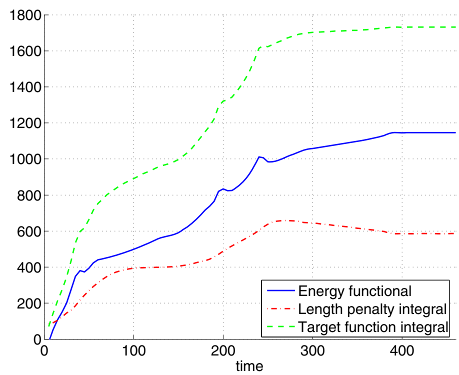
(d) $time = 60$ (e) $time = 150$ (f) $time = 200$

(g) $time = 245$ (h) $time = 320$ (i) $time = 460$

Fig. 4. Iterations using momentum



(a) Without momentum

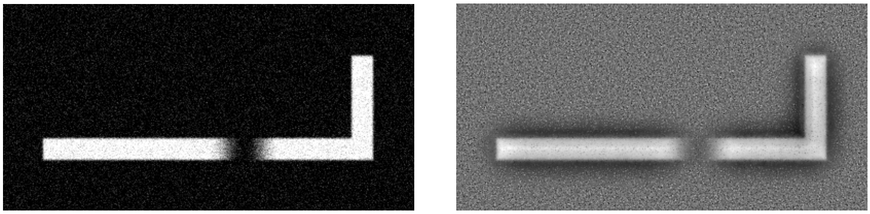


(b) With momentum

Fig. 5. Plots of energy functionals for synthetic test image in Figure 2(a)

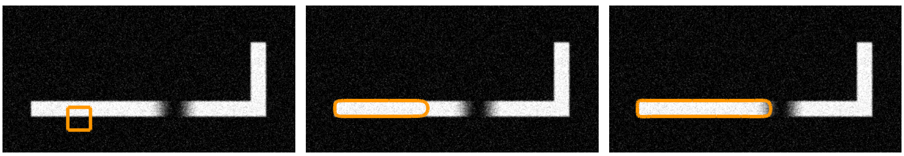
regions (Figure 8). For this experiment, we used the same parameters as in Figure 3 and Figure 4.

As a third test image we used a 458 × 265 retinal image from the DRIVE database [19], shown in Figure 9(a). The target function $f(x, y)$ is illustrated in Figure 9(b). As in the previous experiment, bright and dark colors indicate positive and negative values for $f(x, y)$. The convergent result *without* momentum using parameters $\alpha = 0.07$ and reinitialization every tenth time unit is shown in Figure 10, given the initial condition in Figure 10(a). Applying



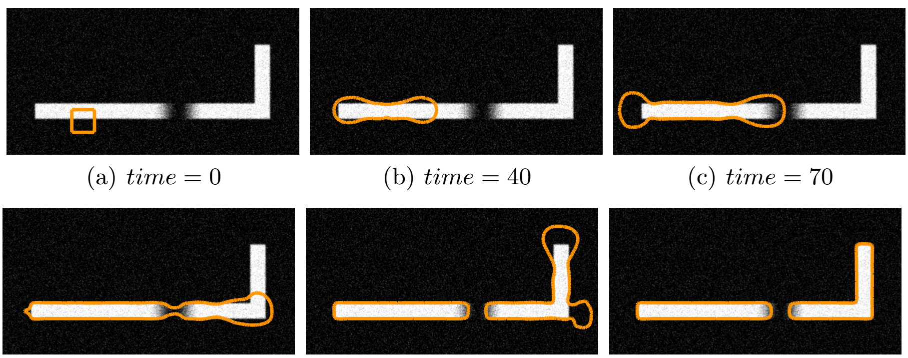
(a) Input image (b) Target function $f(x, y)$

Fig. 6. Synthetic test image illustrating the presence of a local optima in the solution space



(a) $time = 0$ (b) $time = 200$ (c) $time = 515$

Fig. 7. Iterations without momentum (conventional gradient descent)



(a) $time = 0$ (b) $time = 40$ (c) $time = 70$

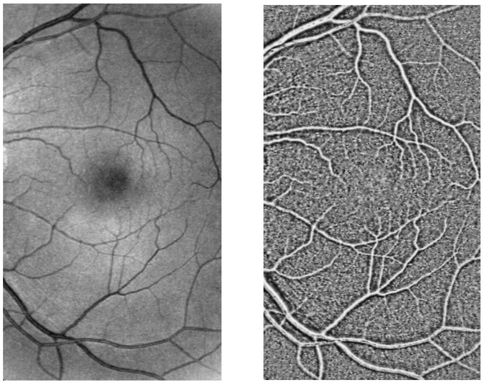
(d) $time = 180$ (e) $time = 240$ (f) $time = 485$

Fig. 8. Iterations using momentum

the idea of momentum yields the result in Figure 11, using the parameters $\alpha = 0.07, \omega = 0.5, \eta = 1.3, s_{max} = 40, \Delta t = 10$. The energy functionals are plotted in Figure 12 to display the convergence of both methods.

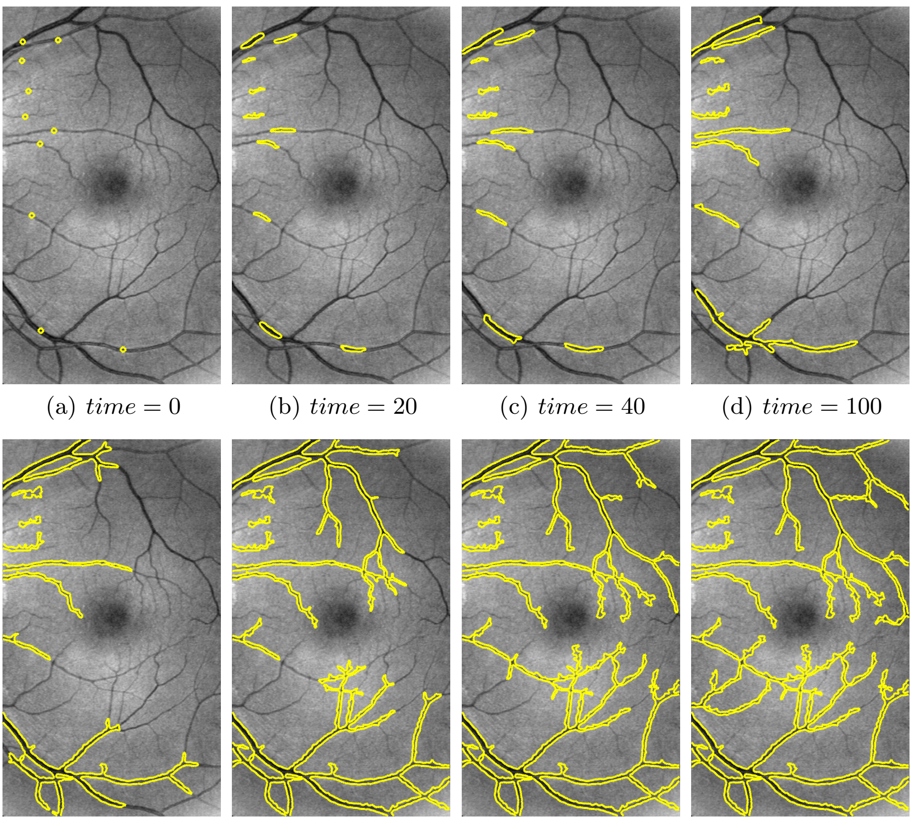
5 Results

The synthetic test image in Figure 2(a) illustrates a local optimum in the solution space when applying the parameters in our first experiment. As expected,



(a) Input image (b) Target $f(x, y)$

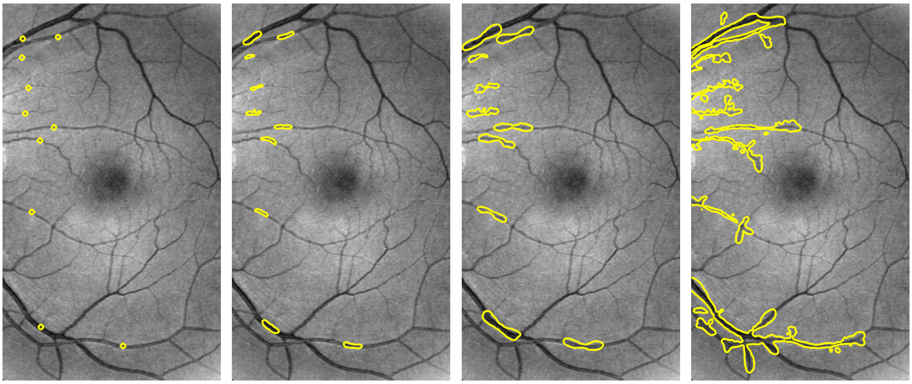
Fig. 9. Retinal image



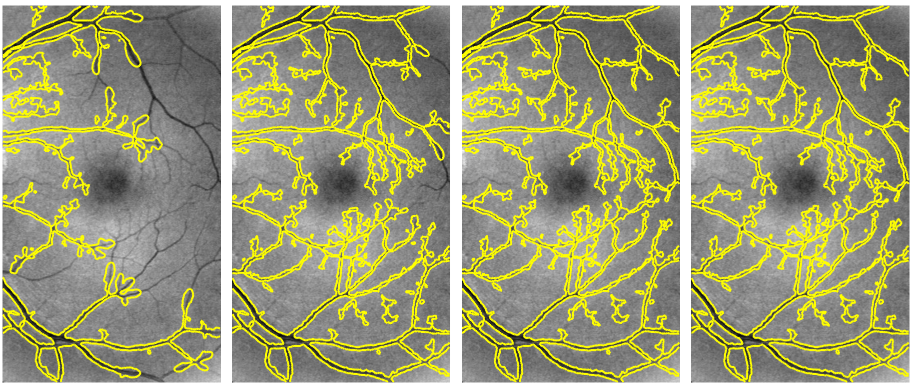
(a) $time = 0$ (b) $time = 20$ (c) $time = 40$ (d) $time = 100$

(e) $time = 200$ (f) $time = 400$ (g) $time = 600$ (h) $time = 1210$

Fig. 10. Iterations without momentum (conventional gradient descent)

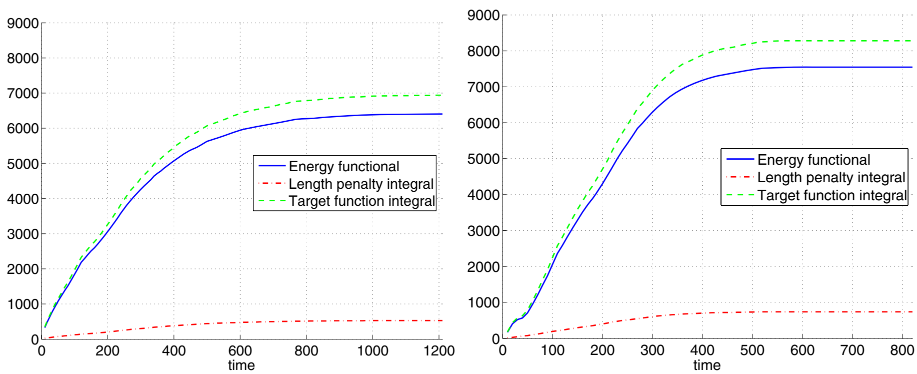


(a) $time = 0$ (b) $time = 20$ (c) $time = 40$ (d) $time = 100$



(e) $time = 200$ (f) $time = 400$ (g) $time = 600$ (h) $time = 820$

Fig. 11. Iterations using momentum



(a) Without momentum (b) With momentum

Fig. 12. Plots of energy functionals for the retinal image in Figure 9(a)

the conventional gradient descent approach converges to this local optimum as depicted in Figure 3. In contrast, our proposed method gains enough momentum in order to overstep the optimum, while at the same time the global solution is reached more rapidly. The process (illustrated in Figure 4) intuitively expands the curve beyond a local optimum, followed by a retraction if the search does not provide any increase in that direction. Using a slightly modified input image, our second example shows that our method is capable of capturing global optima, even when the solution consists of separated regions (Figure 8).

Our third example illustrates our method on real data using a retinal image. In Figure 10 we see that conventional gradient descent fails to capture many weak signal blood vessels. This is a typical case of local optima solutions introduced by noise and poor image contrast. Under the same conditions, gradient descent with momentum captures practically all visible vessels as shown in Figure 11. Note that this example does not include any verification of the accuracy of the segmented vessels. The primary purpose is to illustrate that our method reaches a stronger optimum value for the energy functional, as shown in Figure 12.

6 Conclusions and Future Work

In this paper we have presented the idea of gradient descent with momentum in the context of segmentation using the level set method. We have illustrated the drawbacks of conventional gradient descent and showed examples on how the solution is improved by adding momentum. In contrast to much of the previous work, we have improved the solution by changing the method of solving the optimization problem rather than changing the parameters of the energy functional.

In the future, we will further study the general optimization problem of image segmentation to propose more efficient solutions. Regarding the particular idea of momentum, we will apply this on real applications and verify the quality of the results.

References

1. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. ICCV 2003, October 2003, vol. 1, pp. 26–33 (2003)
2. Papandreou, G., Maragos, P.: Multigrid geometric active contour models. IEEE Transactions on Image Processing 16(1), 229–240 (2007)
3. Kenigsberg, A., Kimmel, R., Yavneh, I.: A multigrid approach for fast geodesic active contours. Technical Report CIS-2004-06, Technion–Israel Inst. Technol., Haifa (2004)
4. Paragios, N., Mellina-Gottardo, O., Ramesh, V.: Gradient vector flow fast geometric active contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(3), 402–407 (2004)
5. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. IEEE Transactions on Image Processing 10(10), 1467–1475 (2001)

6. Weickert, J., Kühne, G.: Fast methods for implicit active contour models. In: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, Heidelberg (2003)
7. Charpiat, G., Keriven, R., Pons, J.P., Faugeras, O.: Designing spatially coherent minimizing flows for variational problems based on active contours. In: Proc. ICCV 2005, October 2005, vol. 2, pp. 1403–1408 (2005)
8. Sundaramoorthi, G., Yezzi, A., Mennucci, A.: Sobolev active contours. International Journal of Computer Vision 73(3), 345–366 (2007)
9. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation, pp. 318–362. MIT Press, Cambridge (1986)
10. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Heidelberg (2006)
11. Kimmel, R.: Fast edge integration. In: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, Heidelberg (2003)
12. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)
13. Osher, S., Fedkiw, R.: Level Set and Dynamic Implicit Surfaces. Springer, New York (2003)
14. Peng, D., Merriman, B., Osher, S., Zhao, H.K., Kang, M.: A pde-based fast local level set method. Journal of Computational Physics 155(2), 410–438 (1999)
15. Sethian, J.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Science 93, 1591–1595 (1996)
16. Zhao, H.K.: A fast sweeping method for eikonal equations. Mathematics of Computation (74), 603–627 (2005)
17. Läthén, G., Jonasson, J., Borga, M.: Phase based level set segmentation of blood vessels. In: Proc. ICPR 2008, Tampa, FL, USA, IAPR (December 2008)
18. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Netherlands (1995)
19. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 23(4), 501–509 (2004)