**Partial Differential Equation Acceleration for level set image segmentation**

Abstract

In this project we further applied a novel framework, which called PDE accelerations, to the problem of Chan-Vese active contours. The original Chan-Vese model convergences slowly and has a bad performance on noisy images. The feature of the level set surface is increased by employing a new model called Fuzzy region competition to replace the initial signed distance surface. The gradient descent method, which is utilized by general PDE problems to find the optimal solution including aforementioned models, has been replaced by our PDE acceleration scheme. A simple explicit finite difference scheme solution of the PDE acceleration is used in our project and in this case the acceleration is achieved by the enhancement in the CFL condition from Δt ~ Δx2 (for diffusion) to Δt ~ Δx (for wave equations). To exploit the potential of PDE acceleration, the hard constrain of surface in Fuzzy region competition is substantiated by soft penalization. We carefully explore our new algorithm on Chan-Vese level set segmentation with both total variation regularization and Beltrami regularization and experiment results are compared with another top-of –art algorithm called Chambolle-Pock primal dual.

Main text

Introduction

Binary segmentation has long been a hot topic in computer vision. The general goal is to detect the object we interested in and separate it from the background. To achieve this, a contour is evolved to overlap with the boundary of the object. There have been lots of methods proposed to solve this problem.

In the early stage classical segmentation models, such as Snake model[] and other active contours[], an edge detector is employed to help the curve find boundary based on the gradient of the image. However, they are not capable of handling complex images. For example, they are susceptible to images with complicated or ambiguous edges and they do not have a good performance on noisy images because in such cases the gradient of image is not a reliable indicator of the boundary. Therefore there is a great need to increase the generality of the active contour and to find more precise boundary.

Recently, region-based segmentation models, such as ChanVese[] or Mumfold-Shah[] are proposed to solve these problems and have been widely used in segmentation tasks. Region-based models segment the image based on the pixel values, assign the pixels with similar value to the same region and keep the regions as uniform as possible. The gradient flow of region based model is not related to the gradient of the image, therefore it could be applied to image segmentation with smooth boundary and discontinued boundary. It could also detect the boundary with meaningful gradient and meaningless gradient. But there still some limitation of these models. The convergence speed is too slow and there still space for improvement of performance on noisy image. Fuzzy region competition model[] is developed to improve the speed of convergence by using a new kind of level set, which constrains the level set between 0 and 1.

However, all the aforementioned models are classical PDE model and the gradient descent method is utilized to find the solution of the PDE. Most of the PDE image segmentation problems have the form

(1)

Where L is a convex functional of variable *u* and the gradient descent of this functional is:

(2)

It is not efficient to use gradient descent to solve the problem because it is under the constrain of stability condition(CFL condition) . More efficient methods are proposed, such as Chambolle-Pock primal-dual, to avoid this convergence speed limitation.

Accelerated gradient search algorithms have been widely used in machine learning. The gradient descent convergences slowly because on a large scale, the local gradient direction is not precise, resulting in large move in poor direction and large fixation in contrary direction. Accelerated method converges considerably faster and is more robust in searching local direction since pass information is incorporated into the current descent. In Polyak’s heavy ball method[], which is one of the oldest acceleration models, the update of the variable is:

(3)

where is the momentum term, which corrects the current gradient direction by incorporating previous gradient direction.

Nesterov[] momentum is a recently acceleration method that is very prevailing among machine learning community.

(4)

This was furthered developed by Sundaramoorthi and Yezzi to adapt it to Partial Differential Equation settings, which have been applied to many PDE problems such as active contour, diffusion, optical flow. In this project, we applied this framework to the level set image segmentation problem of with both TV regularization and Beltrami regularization. The hard clip in Fuzzy region competition is replaced by ramp penalization in TV regularization and quadratic penalization in Beltrami regularization to fully explore the potential of momentum method .The comparison is made between our model and other top-of-art model such as Chambolle-Pock.

Chan-Vese level set method

1. The Chan-Vese Model

1.1 Definition of energy function

Chan-Vese model[] require the definition of an energy functional. The purpose of energy functional is to create a model for the energy of the image that the contour C will attempt to minimize. At each time step, the goal of the active contour is to progress to a lower energy. The Chan-Vese model segment an image into two regions based on pixel values, therefore the goal of energy functional is to minimize the energy in both side of the contour.

The energy functional could be defined as follows:

(5)

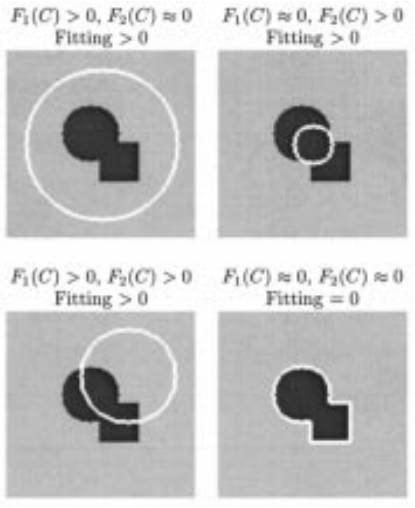


Figure : Region based segmentation[]

In the equation (5), is the image, and are constant values which represent the average pixel intensity inside and outside the contour respectively. ,, , and are weight of each energy term. The first two terms pertain to the intensity found in the two regions the contour seeks to create. Whatever is inside the contour should be an object of interest and whatever is outside of the contour should be the rest of the image. The first two terms are called fidelity, which seek to minimize the energy on both sides of the contour and keep the regions as uniform as possible. The third and fourth term are called regularity, which seek to penalize the length of the contour, which will lead the contour to attempt to enclose with as little length as needed. The final term seeks to penalize the area of the contour, which will attempt to force it only enclose the smallest area possible. The parameters in front of each term vary the influence each energy term has on the overall energy.

* 1. Level Set formulation

Instead of directly evolving and parameterizing the contour, we can redefine the problem in the level set formalism. A surface *φ(x,y)* is defined in the space *Ω: ¶Rmxn* , which is the same scale of the image and the contour is represented by the zero level set of surface *φ*.

(6)

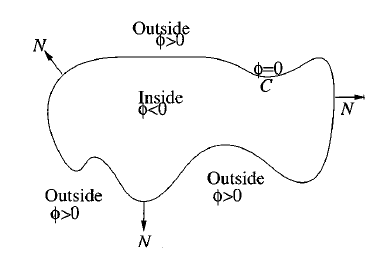
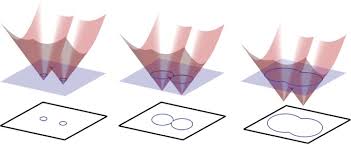
 

Figure : Surface and contour

The problem is converted to searching for the solution in terms of surface *φ*. So the energy function need to be rewritten in terms of surface *φ*. To find the minimum value of the functional (5), we have to use the Euler -Lagrange equation to obtain the gradient flow. The computation is much simpler if the integral boundary is fixed. Therefore some auxiliary functions are defined to convert the integral region of four integral terms into the image space *Ωmxn*.

The Heaviside function is defined as , which means the value of this function is one inside the contour and zero outside the contour. The delta function is defined as , which is the derivative of Heaviside function in terms of level set.

We express each energy terms by level set :

(7)

(8)

(9)

(10)

So the energy function can be rewritten as

(11)

* 1. Derive gradient descent

The gradient descent method is used to find the minimum of the energy function in terms of *φ*.

(12)

* 1. Numerical implementation

For numerical implementation, the Heaviside function could be written in the form as . The delta function is written as .

The above algorithm is expressed in a continuous form, both in time and space. The model has to be discretized with finite difference implicit scheme for implementation purpose. For notation: let Δ*x* be the space between pixels(assume Δ*x* = Δ*y*), Δ*t* represents the time step, *(i,j)* is the space location, n is the iteration time. The finite difference in space could be written as:

The forward difference in time is used in the update of level set :

(13)

* 1. Stability condition

In the original ChanVese method, the reinitialization process is utilized to help keep the surface as a signed distance function. Stability condition is not discussed and there is not strict limitation for time step because the reinitialization help prevent the surface from blowing up. However, in our PDE acceleration scheme reinitialization could not be used because it will interrupt the accumulation of the descent. Therefore the time step in our scheme has to be constrained by strict stability condition. Van Neumann analysis is used to determine the CFL condition for these terms.

(14)

Using a forward difference in time to approximate the time derivative on the left hand side we obtain:

Taking a Discrete Fourier Transform (DFT) on both sides of the homogeneous part to obtain:

(15)

where is called gradient amplifier, defined by .

To prevent blowing up , and the gradient amplifier is real and non-negative. In such cases the stability constraint takes the form of the following CFL condition

(16)

To obtain the maximum gradient amplifier, we seek for the maximum gradient amplifier for fidelity term and regularity term separately then we can get the maximum for the whole gradient amplifier.

1.5.1 Stability condition for fidelity term

The gradient flow of the fidelity term is

(17)

Expand the bracket to get

We have to rewrite it in the form of so that we could obtain the maximum gradient amplifier . Apply Taylor expansion for delta function at the point , where is a very small value so we can obtain , where Therefore .

Therefore

(18)

We separate the z(*x*,*t*) into two parts and , then we search for the maximum value of two parts independently and obtain the maximum value of their product.

For , we analysis the monotonicity of three variables then we obtain when reaches its maximum value

To obtain , we calculate the derivative of , which is , and set it as zero to get the extreme value point.

When , we can obtain

(19)

Multiply the maximum value of those two parts we obtain the maximum value of gradient amplifier for the fidelity term:

(20)

1.5.2 Stability condition for regularity term

The gradient flow of the regularity term is

(21)

It is not easy to derive the gradient amplifier if the gradient flow is in this form, therefore we rewrite it as

Where denotes the normal unite gradient vector of .

As we are calculating the gradient amplifier of the gradient flow, so it doesn’t matter what is the gradient direction. Thus we assume that its gradient direction align with the y-axis, .

Take the discrete Fourier Transform of it

Therefore we obtain the gradient amplifier

(22)

As . The maximum of its value is:

(23)

Then we sum the maximum value of the gradient amplifier for fidelity tern and regularity term to obtain the maximum value of the total gradient amplifier.

(24)

As there is always a point on the surface where , so . In this case, the CFL condition is *dt = 0*. As a result, the surface could not evolve. The problem is from the total variation regularization term.

If we want to use TV regularization for segmentation, we can use a minimum acceptable to plug in the gradient amplifier . A quantization level Q could be utilized to represent digital u and the approximation bounds for

(25)

Then we substitute the level set value between two pixels with the quantization interval Q. The quantization level will cause distortions between pixels, but it is not the same as the violation of CFL condition which will cause blowing up. Instead, the instability degree will be constrained by the quantization level. However, in that case, performance of the algorithm declines because the update of level does not strictly consistent with the gradient flow of the energy function.

We then plug

into the gradient amplifier to obtain:

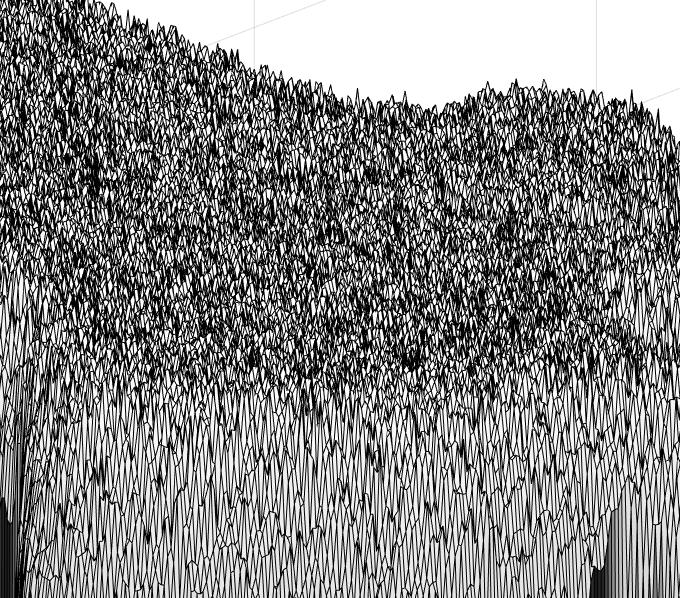
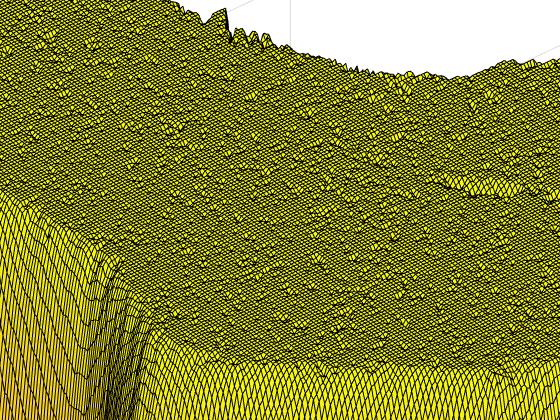
(26) 

Figure : Surface of TV regularization Figure : Surface of Beltram**i** regularization

* 1. Beltrami regularization

Beltrami regularization term is defined as . The new energy function is

(27)

The new gradient flow is:

(28)

To get the gradient amplifier of new gradient flow, we calculate the fidelity term and regularity term independently and then combine them as what we have done in the last part. As the fidelity term remains unchanged, we only need to calculate the regularity term.

(29)

As when , .

Plugging this into the gradient flow yield:

(30)

Take the discrete Fourier Transform of it

Remind that , therefore the upper bound for gradient amplifier

(31)

1. Fuzzy region competition

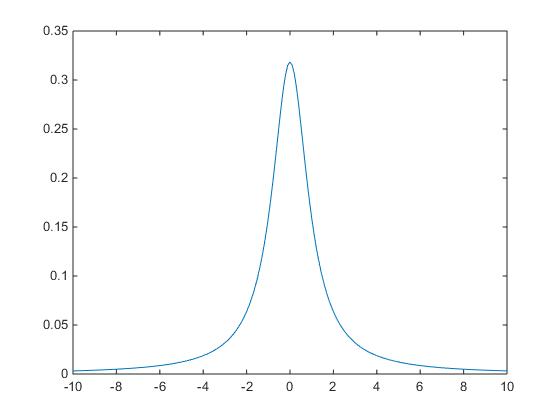
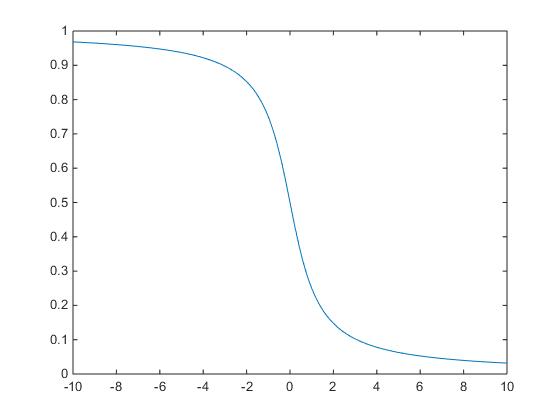
The convergence speed of traditional Chan-Vese model is considerable slow because of the delta function. In the gradient flow(28), all the terms are multiplied with delta function. Recall that Delta function is the derivative of the Heaviside function. 

Figure : Plot of Heaviside function Figure : Plot of Delta function

As shown in the plot, the delta function has a high peak value in the vicinity of 0 level set, which corresponds to the position of the curve. Therefore, for each iteration the evolvement of level set surface will only happen in the vicinity of the curve. The convergence speed is barricaded by the property of delta function.

In the fuzzy region competition model[], the surface is constrain within [0,1] and the contour is 0.5 level set. We use *u* to denote the new surface. Chan-Vese model, the initial surface is a signed distanced function, which means the slope of the surface is 1 and its shape is like a cone. There is no limitation of the surface value. Whereas in the fuzzy region competition model, the surface is strictly constrain with in [0,1]. If during an update the value of a point on the surface becomes larger than 1, then it will be hard clipped to 1. It is same for level 0. There is no need to utilize Heaviside function construct the link between surface value and category of that point(inside, outside or on the curve), so the delta function is on longer an impedance of the convergence speed.

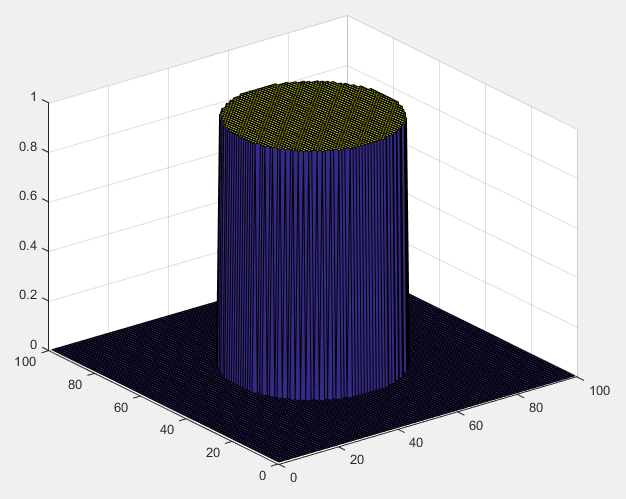
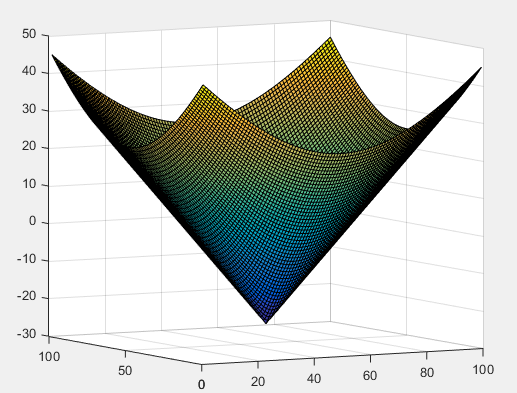


Figure : Chan-vese surface Figure : Fuzzy Region Competition Surface

The new energy function is:

Beltrami regularization:

(32)

TV regularization:

(33)

Obtain the gradient flow with Euler-Lagrange Equation

Beltrami regularization

(34)

TV regularization

(35)

In the gradient flow (34) and (35), the fidelity is homogenous term, therefore it has no contribution to the stability limitation. The gradient amplifier is only determined by regularity term.

Beltrami regularization:

(36)

(37)

TV regularization:

(38)

(39)

In this new level set we, the update of the surface is constrained within [0,1]

(40)

1. PDE acceleration scheme

3.1 General PDE acceleration framework

We define the new energy functional which integrates kinetic energy:

(41)

The new action integral comprises two parts. is the potential energy that we have already defined in the last part and represents the kinetic energy. denotes the weight of potential energy and is another time-dependent coefficient. denotes the distribution of mass. To obtain PDE accelerated motion equation for the action integral in the Lagrange sense, we take a variation on the action integral:

Obtain the derivative of action integral

Integral by part for we get

Remind that , therefore

(42)

We denote and we assume that and are constant as value one in space and time. We have

(43)

where *a* is called damping coefficient. Compared with gradient descent

* 1. Explicit Euler discretization for accelerated PDE

We now start to focus on the discretization for accelerated PDE. For the second order derivative term, we use central difference and for the first order derivative term, we use forward, backward and central difference respectively. Thus three different update schemes are yielded.

3.2.1 Forward difference

(44)

Rearrange the expression we have

(45)

Apply Discrete Time Fourier transform to (45) and substitute , where ξis the update amplifier. Then we obtain

(46)

We denote the coefficient for the second order term as A, coefficient for the first order term as B, constant as C. To maintain the stability, we have to keep .

According to the *Root Amplitude Lemma*, we have to satisfy

(47)

For the first condition, as *a* and are larger than zero, is always satisfied. Now we start to consider the second condition. It is . Then we analyze the absolute value. For , we get , which is always satisfied. For , we get . We plug in the maximum value of to obtain the extreme case. Remind that , so we obtain the CFL condition

(48)

For the update scheme we have

(49)

Rewrite the expression in the form of increment .

From here we clearly see a traditional momentum style structure and how previous descent information integrates into the current descent.

3.2.2 Backward difference

(50)

Rearrange the expression we have

(51)

Similarly, we apply Discrete Time Fourier transform to (51) and substitute . Then we obtain

(52)

According to the *Root Amplitude Lemma*, we have to satisfy

For the first condition, as *a* and are larger than zero, is always satisfied. Now we start to consider the second condition. It is . Then we analyze the absolute value. For , we get , which is always satisfied. For , we get . We plug in the maximum value of to obtain the extreme case. Remind that , so we obtain the CFL condition

(54)

For the update scheme we have

(55)

However，the CFL becomes increasingly strictly as the damping coefficient increase. Now we are exploring the PDE accelerated scheme with time-dependent , which means varies with time. In that case, if we apply backward difference scheme, the CFL condition is not independent from the damping coefficient . Therefore we decide to abandon the backward difference scheme.

3.2.3 Central difference

Central difference is applied for both second order and first order time difference, which leads to a second order difference scheme.

(56)

Then we obtain the update scheme:

(57)

Similarly, apply Discrete Time Fourier transform to () and substitute .

(58)

Then start to analysis the CFL condition based on *Root Amplitude Lemma.*

As damping coefficient and time step are always positive, is always satisfied. For the second condition , we have . Therefore . As is always positive, the CFL condition will be

(59)

* 1. Semi-implicit Scheme

The semi-implicit scheme is inspired by classic two-part Nesterov momentum method. The difference between Nesterov momentum and traditional momentum method is that Nesterov momentum adds a correction factor to the traditional momentum method. The idea of Nesterov momentum could be clearly illustrated by the following image, which is from Geoffrey Hinton’s slides.

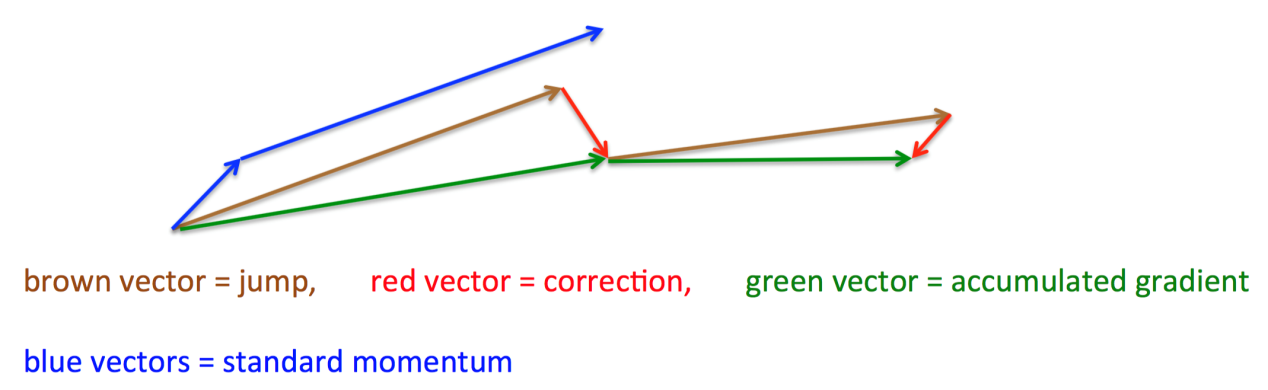


Figure 9: Nesterov momentum

First, take a step in the direction of previous accumulated gradient(brown line). Then calculate gradient of that point(red line). Finally correct the update with that gradient to obtain the real update direction(green line). Traditional momentum update is shown as blue line.

Similarly, Euler discretization is utilized to obtain a two-step style Nesterov momentum of our PDE scheme. This strategy is applied to the central difference scheme, to obtain the following semi-implicit update scheme, where the first and second steps are separated.

(60)

(61)

To obtain the stability CFL condition, we employ the Von Neumann analysis. The DFT of above two equations:

(62)

(63)

Then substitute in the () into () we could obtain

(64)

Then start to analysis the CFL condition based on *Root Amplitude Lemma.*

For the first condition, , we have . After simplification, the expression is . It is only satisfied between its two roots. As , its negative root is smaller than 0. Thus,

However, this expression is dependent on damping, which is not acceptable. To obtain an expression which is independent from damping, set and solve the upper bound for . The partially derivative of on damping is

As () is negative when and () is positive when , while is always positive, reaches its upper bound when . Substitute it into :

(64)

Then to start analysis the second condition,

When is small, is positive. Then the equation becomes and it is always satisfied. When is large, is negative, the equation becomes

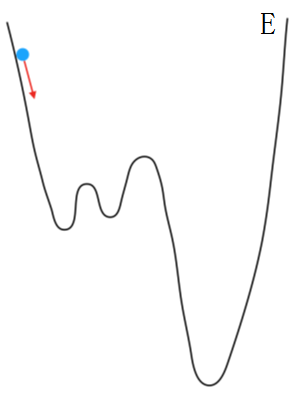
Set . The partially derivative of on damping is . As it is always positive, thus when , reaches its upper bound. Finally the second stability condition is

(65)

Notice that (65) is stricter than (64). Thus the CFL condition for semi-implicit is (65).

* 1. Physical interpretation of PDE acceleration

The searching for the minimum value of a potential functional can be viewed as a physical ball rolling down a valley guided by gravity. The gravitational potential energy of the ball is the potential functional and its kinetic energy is .

In gradient descent, the velocity of that ball is proportional to the slope of current position, which is the gradient . In contrast, the speed is accumulated in kinetic energy in the PDE acceleration scheme.

There are two benefits using acceleration optimization. First, previous momentum is integrated into current momentum. So even when the current gradient is small, previous momentum could energize the ball to move forward. Second, momentum will prevent the optimization process from being trapped in local minimum, which is a very intractable problem. If the current gradient is zero, the momentum will still move the ball forward to look for potentially better minimizer.

Figure : Accelerated descent physics interpretation

* 1. Gradient flow propagation form

In the experiment, the evolvement of the start from the curve and propagate to the whole surface. In this process, the gradient flow is the driven force behind the evolvement.

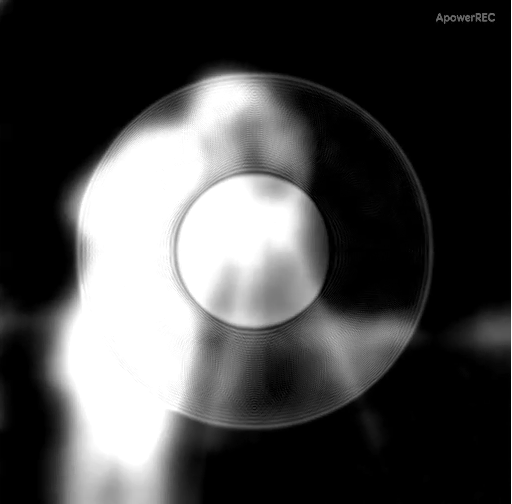


Figure : Diffusion Figure : Wave equations

We discover that when the gradient flow is gradient descent, which indicates that the CFL condition is proportional to the second order of the space step , the propagation takes the form as diffusion. While when the gradient flow is acceleration, which indicates that the CFL condition is proportional to the first order of the space step , the propagation takes the form as wave equations. As the grid is normalize to 1, the space step , where *m* is the length of the image. Therefore the second order of the time step is significantly smaller than its first order. The time step is released when the propagation takes the form of wave equation. That is the reason why the convergence speed of PDE acceleration is considerably faster than gradient descent.

1. Accelerated PDE for Fuzzy region competition

4.1 Level set update scheme

In the original Fuzzy region competition model[], an old version of primal dual algorithm is utilized to search for the solution. The convergence speed has already been enhanced through the improvement of level set form. In addition, we could further optimize the process by replacing the old version of primal dual algorithm with our PDE Acceleration scheme to find the local minimum. Primal dual is another state of the art algorithm, which has been developed for couple of years. Later in this report, we will also use the latest version of primal dual algorithm in the Fuzzy region competition model as a comparison experiment.

Substitute the maximum gradient amplifier, damping coefficient and gradient flow of both Beltrami regularization and TV regularization into accelerated framework we have

Beltrami:

Explicit scheme with Forward difference:

(66)

Explicit scheme with Central difference:

(67)

Semi-Implicit scheme:

(68)

Total Variation:

Explicit scheme with Forward difference:

(69)

Explicit scheme with Central difference:

(70)

Semi-Implicit scheme:

(71)

* 1. Search for the optimal damping

Then we have to find the optimal damping coefficient for accelerated PDE. Damping coefficient plays a critical role in the performance of momentum. Large damping coefficient will lead to decline of the convergence speed, while small damping coefficient will result in instability.

According to [], for a linear Dirichlet problem

(72)

where *L* denotes a uniformly elliptic second order partial differential operator, *b* and *c* are the weights for the second order term and the first order term respectively, *f* denotes the constant term. The problem is defined in the domain . The optimal damping coefficient is[]

(73)

where is the first the first Dirichlet eigenvalue of *L*. Now we apply this analysis to accelerated PDE motion function of Fuzzy region competition with Beltrami regularization. Rewrite () in a linear form

(74)

We notice that there is no first order term in the expression, therefore The coefficient for the second term and the second order elliptic operator .

To derive the first Dirichlet eigenvalue of , the denominator is throw away first. Then the elliptic operator becomes , which is a Laplacian operator. From [] the first Dirichlet eigenvalue for Laplacian is . To deal with the denominator, the average value of it is taken over the whole image and then multiply it with .Remind that in our setting , so the first Dirichlet eigenvalue here is . The optimal damping coefficient of Fuzzy region competition with Beltrami regularization

(75)

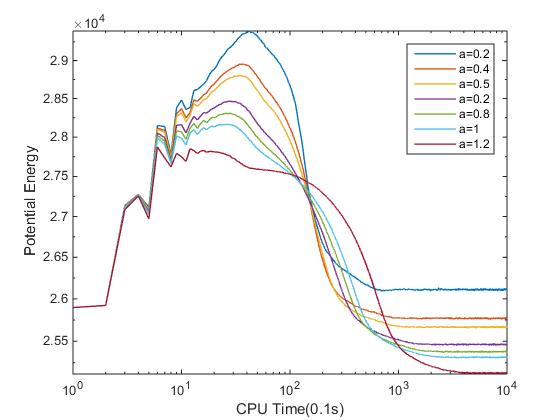
For TV regularization, there is no optimal damping coefficient. It is because we plug in quantization level Q to substitute the minimum acceptable gradient between points, therefore the TV regularization is no longer a strictly PDE problem. 

Figure : TV regularization energy plot

The analysis of energy plot confirms our supposition. As shown in the energy plot above, the total energy increases rapidly at first. The reason for that is in the TV regularization, there is a certain degree of distortion of the surface, which contributes to the regularization energy. The final energy might be higher than initial energy, but we only care about where zero level set lays, which decides the segmentation result. Therefore the upshot of total energy is not an error, as the smoothness of the surface is not in our consideration.

In most cases, we set .

1. Primal dual method

Primal-dual optimization is another successful technique and it has been widely utilized by many algorithms to solve PDE problems. These methods[] are easy and simple to implement and converge considerable faster than gradient descent. Primal-dual TV algorithm was first proposed by Chambolle[] and further developed by [] to make it more efficient. Recently, it has been extended to Beltrami regularization on inverse problems.

The primal-dual algorithm is to find the solution for a saddle point problem, which is established using Legendre-Fenchel transform

(76)

where and are convex functional, is a continuous linear operator, is the convex conjugate of , and are the primal and dual variable respectively. For detailed proof please check [].

In this section, we apply both primal-dual TV algorithm and primal-dual Beltrami algorithm to level set segmentation problem.

5.1 Beltrami regularization

Now we derive our Beltrami segmentation model [] guided by primal-dual Beltrami-denoising optimization problem developed by []. By comparison our potential energy functional[] with the equivalence [], we get the following expression

(77)

(78)

where the linear operator , integral domain , .

We set . The Legendre transform of is

(79)

Hence we obtain the convex conjugate of

(80)

where . Plug and into we get our primal-dual Beltrami segmentation problem:

(81)

Further, we integrate the by part and rewrite the problem as

(82)

We now use the projected gradient method to solve the problem

(83)

To avoid the denominator become zero, we multiply the second term with .Then projection algorithm is used to yield the update scheme

(84)

where and are the update time steps for prime and dual variables respectively. To achieve the constrain on the variable domain we have

(85)

Plug into () we finally obtain the primal-dual Beltrami segmentation algorithm

(86)

* 1. TV regularization

By comparison TV regularization potential energy functional [] with the equivalence [], we get the following expression

(87)

(88)

where the linear operator , integral domain ,

We set . The Legendre transform of is

(89)

Hence we obtain the convex conjugate of

(90)

where . Plug and into we get our primal-dual TV segmentation problem:

(91)

Further, we integrate the by part and rewrite the problem as

(92)

We now use the projected gradient method to solve the problem

(93)

Then projection algorithm is used to yield the update scheme

(94)

To achieve the constrain on the variable domain we have

(95)

Plug into () we finally obtain the primal-dual TV segmentation algorithm

(96)

* 1. Connection to PDE acceleration

As we observed, there is a close relationship between our PDE acceleration scheme and primal dual method. It is proposed in [] that a damping wave numerical scheme is equivalent to primal dual method on Dirichlet problem. We further developed their discovery to the obstacle problem and we explore that there is an numerical PDE interpretation of primal dual, which guide us to improve the performance of primal dual.

To get the connection between primal dual and PDE acceleration scheme, we just take a brief overview of the derivation of primal dual algorithm. The general form of regularization energy functional of level set segmentation problem is:

(97)

where is the regularization term, whether it can be TV regularization or Beltrami regularization. Then take the Legendre-Fenchel transform of that term we obtain

(98)

Integrate by part we get

(99)

The searching of solution of that problem could be seen as alternatively update the primal variable *u* and dual variable *p*. Therefore the primal dual algorithm could be interpreted as jointly operating gradient ascent on *p* and gradient descent on *u*.

(100)

The coefficient is the ratio between the time steps of primal and dual variable. From the above equations we eliminate the dual variable to get the second order update equation:

(101)

Compare it with our PDE acceleration damping wave equation

The form of two update schemes share a high similarity and this PDE numerical form of primal dual algorithm is a new observation so far. We could see that the ratio of primal and dual update time step occupy the same position as damping coefficient , which serves as an guidance on how to choose time steps . We could also notice that the damping coefficient in primal dual algorithm could not be independently separated from the gradient flow.

The choice of primal and dual time step has long been a problem confusing the researchers. The convergence analysis in [Chambolle-Pock 2010] shows that there is an upper limitation for the multiply of two steps,

(102)

However, we still could not confirm the time steps. Previously, most of the researchers attempt to find the optimal sets of time steps manually. However, this is time consuming and we could not make the algorithm operate at the best performances in most cases.

Inspired by the similarity between primal dual and acceleration in terms of PDE numerical scheme, the optimal damping coefficient is the optimal ratio of time steps. With the new constrain we could confirm the optimal set of time steps.

In Beltrami regularization, , and . Therefore

(103)

In TV regularization, , . Notice that there is no optimal damping in TV regularization, therefore we choose

(104)

1. Soft penalization

The acceleration method takes advantage of integrating previous momentum into current momentum. However, in the Fuzzy region competition model we hard clip the level set surface that extend the boundary, which suddenly reduce the momentum and result in the interruption of acceleration process. We assume that hard clip process may have deteriorating effect on the performance of accelerated descent.

We substitute hard-clip with soft penalization to further increase the performance of PDE acceleration on obstacle problem. We consider the *L2* penalization(quadratic)

(105)

Where the second term is the penalization for level set above one and the third term is the penalization for level set below zero. and are weights of two penalization terms. and are the subspace of Ω. and Apply the new energy functional to the update scheme() we can obtain

(106)

Apply Euler discretization framework(15) to the update scheme

(107)

Then we get the explicit solution

(108)

The CFL condition for this scheme is only determined by and is unrelated to the penalty and . And we have proved that the accelerated optimization for obstacle problem is independent from the value of and . Hence we take the limit of and

(109)

Substitute the limitations into the update scheme and therefore we could obtain a simpler scheme

(110)

Surprisingly, this indicates that the soft penalization is equivalent to hard-clip in terms of mathematically. Therefore the soft penalization method would not contribute to the enhancement of algorithm speed.

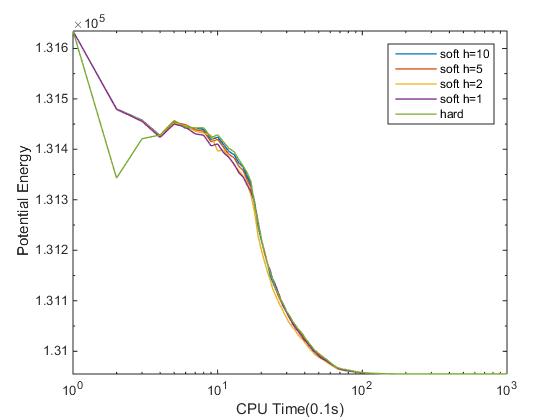


Figure : Soft penalization

This assumption is confirmed by the experiment result above. The convergence speed does not increased by soft penalization. However, the soft penalization is slightly slower than hard-clip for the reason that it is more complicated.

1. Experiment result

7.1 Impact of parameters on the segmentation result

7.1.1 Beltrami coefficient for Beltrami regularization

**  **

Figure : Segmentation result for Beltrami regularization with λ=0.05(Left β=0.5, Middle β =1, Right β=2) on 512×512 noisy cameraman image

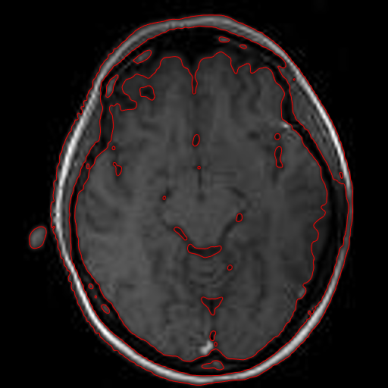
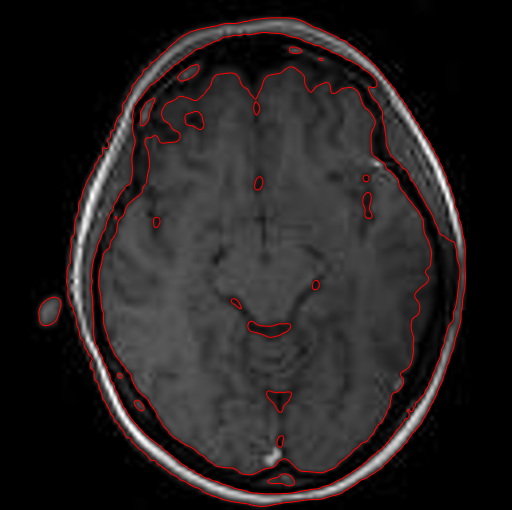
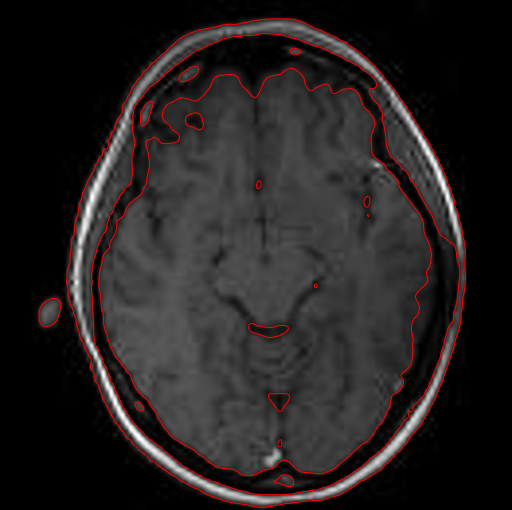
  

Figure : Segmentation result for Beltrami regularization with λ=5(Left β=0.5, Middle β =1, Right β=2) on 512×512 brain image

The image segmentation could be treated as a data-fidelity optimization problem. The fidelity energy term here measures the fidelity to the segmentation contour merely based on pixel value and the regularization term smooth the contour. There are two regularization methond that widely applied in the research:

However, the segmentation contour produced by *H1* regularization is overly smooth, while TV regularization feature-preserving but introduces staircasing artifacts. Therefore Beltrami regularization is introduced to balance the fidelity and regularity.

The Beltrami coefficient is to control the compromise between feature fidelity and regularity. As we could see from the experiment results above, when Beltrami coefficient increases, the feature preserving effect diminishes and contour smoothness enhances.

7.1.2 Fidelity coefficient

Figure : Segmentation result for Beltrami regularization with β=2(Left λ=0.01, Middle λ=0.05, Right λ=0.1) on 512×512 noisy cameraman image

Figure : Segmentation result for TV regularization(Left λ=0.5, Middle λ=1, Right λ=2) on 512×512 noisy cameraman image

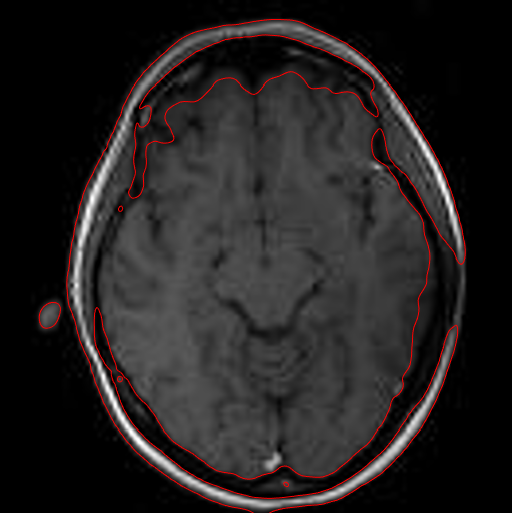
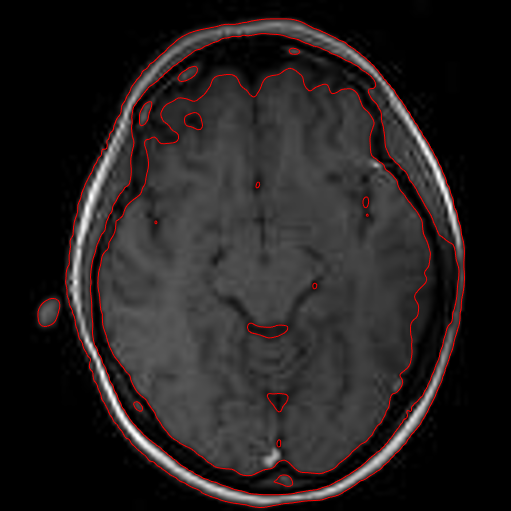
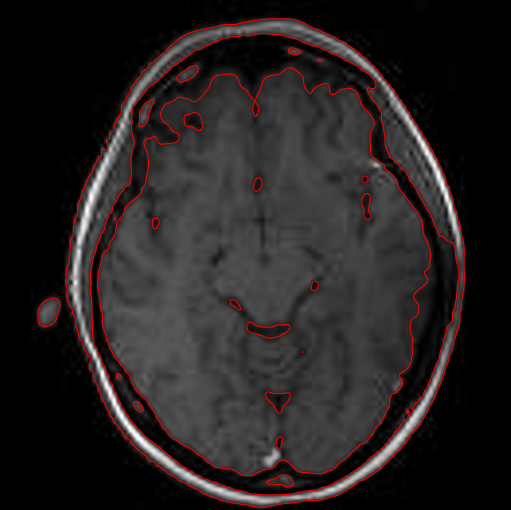
  

Figure : Segmentation result for Beltrami regularization with β=2(Left λ=0.01, Middle λ=0.05, Right λ=0.1) on 512×512 noisy cameraman image

The fidelity weight is to regulate the weight between fidelity term and regularity term. The regularization term *g* is always set as one, because only the ratio between two weights that matters. As the fidelity weight increases, the ability for algorithm to capture the detail inceases, and the contour becomes increasely complex. The experiment result is consistency with the function of fidelity weight in theory.

* 1. Comparison between three differential scheme

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *β* = 0.5 | | *β* = 1 | | *β* = 2 | |
| Time | Iterations | Time | Iterations | Time | Iterations |
| Forward | 9.29s | 257 | 9.89s | 281 | 11.28s | 325 |
| Central | 9.05s | 257 | 9.75s | 280 | 11.40s | 327 |
| Implicit | 24.02s | 376 | 26.21s | 413 | 31.62s | 502 |

Table : Comparison among three differential scheme with λ=0.05 on 512x512 noisy cameraman image

*Table 1* shows the runtimes for forward, central, implicit differential scheme respectively. All the experiments are done on a noisy 512×512 cameraman image. Compared to clean image, the task for noisy image is much more challenging because of the local minimum. The algorithm runs on MatLab with 1.9GHz Intel core i5-4300U CPU of Windows 7. The convergence condition is the maximum absolute grid value between current and previous level set is smaller than 10-4.

The explicit scheme with forward and central difference is considerably faster than implicit scheme for Beltrami regularization. Therefore we will give no further consideration to implicit scheme for Beltrami regularization. It is also worth notice the iteration number for forward and central difference is close, which I believe there is connection. I will confirm the connection in the further work.

* 1. Impact of parameters on the convergence time

PDE acceleration result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *λ* = 0.01 | | *λ* = 0.05 | | *λ* = 0.1 | |
| Time | Iterations | Time | Iterations | Time | Iterations |
| *β* = 0.5 | 12.70s | 364 | 8.97s | 257 | 8.94s | 253 |
| *β* = 1 | 23.15s | 666 | 10.01s | 280 | 8.82s | 252 |
| *β* = 2 | 19.96s | 558 | 11.59s | 327 | 8.93s | 256 |

Table : Runtimes for PDE acceleration Beltrami segmentation on 512x512 noisy cameraman image.

Primal dual result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *λ* = 0.01 | | *λ* = 0.05 | | *λ* = 0.1 | |
| Time | Iterations | Time | Iterations | Time | Iterations |
| *β* = 0.5 | 29.90s | 680 | 18.54s | 375 | 30.94s | 675 |
| *β* = 1 | 34.06s | 737 | 20.61s | 435 | 12.69s | 284 |
| *β* = 2 | 19.54s | 424 | 31.85s | 655 | 17.22s | 361 |

Table : Runtimes for primal dual Beltrami segmentation on 512x512 noisy cameraman image.

We carefully explore the effect of fidelity weight *λ* and Beltrami coefficient *β* on the convergence time. Central difference of the first order explicit scheme is applied to and backward difference of the first order explicit scheme is applied to div. Damping coefficient is set as here, from the linear analysis in the section 4.2. The choice of primal and dual time step is from the analysis in section 5.3.

*Table 2* and *Table 3* show the comparison of runtime between Beltrami PDE acceleration and Beltrami primal dual algorithm. From the result above, our PDE acceleration algorithm is faster than primal dual algorithm with Beltrami regularization in most cases.

* 1. Impact of image scale

PDE acceleration result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *β* = 0.5 | | *β* = 1 | | *β* = 2 | |
| Time | Iterations | Time | Iterations | Time | Iterations |
| 128×128 | 0.37s | 196 | 0.46s | 253 | 0.51s | 282 |
| 256×256 | 0.87s | 161 | 1.27s | 239 | 2.46s | 455 |
| 512×512 | 9.03s | 257 | 9.78s | 280 | 11.46s | 327 |
| 1024×1024 | 62.71s | 453 | 65.86s | 462 | 66.58s | 473 |

Table : Runtimes for PDE acceleration Beltrami segmentation with difference image scale (λ=0.05)

Primal-dual result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *β* = 0.5 | | *β* = 1 | | *β* = 2 | |
| Time | Iterations | Time | Iterations | Time | Iterations |
| 128×128 | 0.61s | 300 | 0.62s | 298 | 0.65s | 317 |
| 256×256 | 1.93s | 329 | 2.71s | 466 | 4.63s | 771 |
| 512×512 | 15.63s | 357 | 19.25s | 433 | 29.26s | 652 |
| 1024×1024 | 87.98s | 494 | 95.18s | 534 | 76.25 | 430 |

Table : Runtimes for primal-dual Beltrami segmentation with difference image scale (λ=0.05)

*Table 4* and *Table 5* shows the comparison between PDE acceleration and primal dual algorithm for solving Fuzzy region competition model with Beltrami regularization on 512×512 noisy cameraman image. Our PDE acceleration is roughly 1.5x faster than primal-dual. For small scale of image, PDE acceleration takes smaller numbers of iteration than primal-dual to find the solution. For large scale of image, the iteration number is relative close. However, due to the simplicity of our algorithm, the convergence speed is still faster than primal dual.