MUMFORD-SHAH MODEL WITH FAST ALGORITHM ON LATTICE

Lu YU^{*+}*, Qiao WANG***, Lenan WU***, and Jun XIE*$^{++}$

Department of Radio Engineering, Southeast University, Nanjing, China*
Institute of Communications Engineering, PLA University of Science and Technology, Nanjing, China^{+}
Institute of Command Automation, PLA University of Science and Technology, Nanjing, China^{++}

ABSTRACT

Mumford-Shah piecewise smooth functional is a variational PDE model widely used in image segmentation and smoothing. An analogous discrete model which models image as an MRF has also been built. In this paper, we propose another discrete Mumford-Shah piecewise smooth model on lattice from a different perspective. We present a discrete objective functional, as well as the method to find the solution. Only two simple and deterministic optimization techniques, that is, derivation and greedy algorithm are used in the model to seek the solution. Compared with traditional continuous model, the model in this paper is much simpler and the approach is much easier and faster.

1. INTRODUCTION

Mumford-Shah piecewise smooth model introduced by Mumford and Shah [1] in 1989 is a widely used variational PDE model for image smoothing and segmentation. The model is often implemented by level set approach [2]. The Mumford-Shah functional in level set approach is

$$\begin{aligned} F(u^+,u^-,\phi) &= \iint_\Omega (u^+-u_0)^2 H(\phi)d\chi \\ &+ \iint_\Omega (u^- - u_0)^2(1-H(\phi))d\chi + \mu \iint_\Omega |\nabla u^+|^2 H(\phi) d\chi \\ &+\mu \iint_\Omega |\nabla u^-|^2(1-H(\phi))d\chi + \nu \iint_\Omega |\nabla H(\phi)| d\chi, \end{aligned} \tag{1}$$

where u_0 denotes the image on Ω to be segmented and smoothed, Ω is partitioned by a surface ϕ whose zero level set is Γ into two areas: $\phi > 0$ and $\phi < 0$ on which u^+ and u^- are defined respectively. u^+ and u^- are the smooth images approximating u_0. μ and ν are positive parameters. $H(.)$ is the Heavyside function.

The following Euler-Lagrange functions and gradient flow are the results in [2]:

$$\begin{aligned} u^+ - u_0 = \mu \triangle u^+ \text{ on } \phi > 0,\ \frac{\partial u^+}{\partial \overrightarrow{n}} = 0 \text{ on } \phi = 0, \\ u^- - u_0 = \mu \triangle u^- \text{ on } \phi < 0,\ \frac{\partial u^-}{\partial \overrightarrow{n}} = 0 \text{ on } \phi = 0, \end{aligned} \tag{2}$$

where $\partial/\partial\overrightarrow{n}$ denotes the partial derivative in the normal direction $\overrightarrow{n}$ on the boundary $\phi = 0$.

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \delta(\phi)[\nu\nabla\cdot(\frac{\nabla\phi}{|\nabla\phi|}) - (u^+-u_0)^2 \\ &+(u^- - u_0)^2 - \mu|\nabla u^+|^2 + \mu|\nabla u^-|^2], \end{aligned} \tag{3}$$

where $\delta(\cdot)$ is the Dirac function.

Mumford-Shah functional has nice formulations based on perfect theory of analysis. However, there are many topics in implementation of the model, such as how to handle Neumann boundary condition in (2), how to choose the difference schemes in order to capture the viscosity solution, how to extend u^+ on $\phi < 0$ and u^- on $\phi > 0$, how to reinitialize ϕ in order to keep the surface smooth, etc. Therefore, it is not easy to carry out Mumford-Shah piecewise smooth model.

The difficulties in implementation are mainly caused by the fact that we regard images as functions on continuous domain while digital images are defined on lattice. Why not establish model on discrete space directly? In fact, a discrete Mumford-Shah piecewise smooth model which models image as an MRF has been proposed in [3] and some stochastic optimization techniques were used to find solution. However, that model is complex and the approaches are time costly.

In this paper, we propose another discrete Mumford-Shah model on lattice from a different perspective. Instead of modeling image as an MRF, we model image as a deterministic function on lattice and choose a two-valued surface function ψ with values +1 and -1 only. The idea of discrete level set function was built in [4] and [5]. Both papers aimed at accelerating the evolution of level set while we focus on providing an entirely discrete Mumford-Shah model on lattice. The method in [5] was to update values of the level set function according to the sign of velocity. While we adopt a simple optimization technique known as greedy algorithm to evolve the surface. That is, change the sign of ψ if such operation can decrease the energy. This evolution policy is different from [5] since computation of velocity field is unnecessary. And for Mumford-Shah piecewise smooth model, it takes less time to calculate the change of energy than to compute velocity field. The same technique is used in [4] to implement the Chan-Vese segmentation model [6]. The main difference between this paper and [4] is that we present a discrete energy functional which makes the computation of energy easier. In Mumford-Shah piecewise smooth model, minimization of the energy includes searching a piecewise smooth function to approximate the original image and searching the boundary curve. In our formulation, we present explicit linear equations to solve the piecewise smooth-like function instead of PDEs with boundary conditions. Another difference is that we evolve only boundary points while [4] sweeps every pixel on the image. Thus we can propose a fast method to accelerate the procedure of solving the piecewise smooth-like function.

This work is supported by Open Fund from Key Lab of Image Processing and Image Communication, Nanjing University of Post and Telecommunications.

We'll first introduce the discrete functional, after that, the approach to find the optimal solution is proposed, then all the details of implementation are elaborated. After some experiment results are illustrated, the comparison on various aspects with traditional Mumford-Shah model, as well as the two methods in [4] and [5] are made.

2. MODEL ON LATTICE

We start with the objective functional of the new model on lattice, including its formulations and the approach to seek solution.

2.1. objective functional on lattice

A digital image defined on a $M \times N$ lattice can be represented as $g_{ij}, 1 \le i \le M, 1 \le j \le N, i, j \in Z$. In this paper,we translate the segmentation problem of g into seeking f and ψ on $M \times N$ lattice which minimize

$$E(f,\psi) = \sum_{i,j}(f_{ij}-g_{ij})^2 + \nu \sum_{ij} \sum_{(p,q)\in S_{ij}} (1-\psi_{ij}\psi_{pq}) + \mu \sum_{i,j} \sum_{(p,q)\in S_{ij}} (f_{ij}-f_{pq})^2(1+\psi_{ij}\psi_{pq}), \tag{4}$$

in which g denotes the digital image to be segmented, f denotes the piecewise smooth-like image approximating g. S_{ij} denotes the 8-pixels neighbor around pixel (i,j). The ψ is a two-valued function with values +1 and -1 only and its evolution can follow motion of the segmenting curve Γ and handle topological change naturally.

The first term in (4) requires that f approximate g, we call it *fidelity term*. The $M \times N$ lattice is partitioned by ψ into two areas: $\psi = 1$ and $\psi = -1$. The third term requires that f does not vary too much within each area. For the pixel (i,j), if its neighbor (p,q) is in the same area as (i,j), the difference between f_{ij} and f_{pq} will be counted. Otherwise, the difference between the two will not be counted. That is, no smoothing of f is done across the boundary. This term is called *smoothness term* in this paper. Before explaining the second term, we present a definition frequently used in this paper.

Definition: Let function ψ takes values ± 1 on $M \times N$ lattice, we say that pixel (i,j) is a boundary point if there exists $(p,q) \in S_{ij}$ such that $\psi_{ij} \ne \psi_{pq}$.

In traditional Mumford-Shah model, the last term in functional (1) suggests that Γ, the zero level set of ϕ, as short as possible. In discrete model, it is difficult to compute the curve length, so we use the second term in functional (4) to approximate the curve length. A similar formulation has been used in [1] to calculate the length of a piecewise linear curve on lattice, while the choice of 4-neighbors system in [1] makes the meaning of the formulation is entirely different from ours. Among the neighbors of (i,j), only those whose ψ values are different from ψ_{ij} can attribute to the sum. In fact, it is a weighted sum of all the boundary points on lattice. Experiments show that it is a good approximation to the curve length and a closed curve will shrink to a point with this term alone. Although this term is not the geometric length, we call it *length term* to correspond to the last term in traditional Mumford-Shah functional.

Although the functional (4) and the energy function in [3] have similar forms, there are some underlying differences. First, the latter works within Bayesian framework, the gray levels and edges in it are stochastic variables on MRF, while f and ψ in our model are deterministic functions. Second, the gray levels in [3] vary in a discrete finite set, while the values of f in (4) change continuously. The continuity of f values leads to our very simple optimization method. Furthermore, in [3], the segmenting curve might be unclosed, while in our model, the level set approach ensures the closure of the curve.

2.2. optimization of the functional

For a fixed ψ, optimizing functional (4) over all $f_{mn}(1 \le m \le M, 1 \le n \le N)$ can be implemented by setting $\frac{\partial E}{\partial f_{mn}} = 0$, then

$$2\mu \sum_{(p,q)\in S_{mn}} (f_{mn}-f_{pq})(1+\psi_{pq}\psi_{mn}) + f_{mn} = g_{mn}. \tag{5}$$

According to (5), at least two linear equation systems each correspond to a connected component of ψ can be generated easily. And equation systems can be solved by any iterative method, such as Conjugated Gradient.

For a fixed f, minimizing $E(f,\psi)$ with respect to ψ is solved by greedy algorithm. That is, for each boundary point (m,n), if changing sign of ψ_{mn} can decrease the energy functional (4), then do it, otherwise, do nothing.

3. SOME ISSUES OF IMPLEMENTATION

Now we go to the implementation of functional (4).

3.1. what will change when ψ_{mn} changes

For every boundary point (m,n) we should test whether total energy will decrease or not when ψ_{mn} alters its sign. Then, what will change when ψ_{mn} changes?

Strictly speaking, when ψ_{mn} alters its sign, almost every value of f on lattice will change because the partition areas $\psi = +1$ and $\psi = -1$ have been modified. Due to time constraint, we can't solve equation (5) when every ψ_{mn} changes its sign. And in fact, except f_{mn}, other values of f changes only slightly. So, for ordinary simple images, it is convenient and feasible for us to assume that other values of f do not change. To estimate the changed value of f_{mn}, we write the diffusion equation at (m,n) after ψ_{mn} has altered its sign,

$$2\mu \sum_{(p,q)\in S_{mn}} (\hat{f}_{mn}-f_{pq})(1-\psi_{pq}\psi_{mn}) + \hat{f}_{mn} = g_{mn}. \tag{6}$$

in which there is only one unknown, $\hat{f}_{mn}$.

And in functional (4), the items including ψ_{mn} or f_{mn} would be changed correspondingly.

3.2. how to escape from local minimums

Local minimums are often encountered in optimization. In our model, because approximation to the curve length is not precise enough, local minimums appear more frequently. For example, for a pixel (m,n), when $L = \sum_{(p,q)\in S_{mn}} \psi_{mn}\psi_{pq} = 0$, changing sign of ψ_{mn} does not affect L, but the actual curve length might change. In this case, internal force will not take effect, and if external force happens to be very small, then the evolution might be trapped in a local minimum because of equivalent length term. To avoid this sort of local minimums, we modify L to a negative number when $L = 0$ to enhance the internal force. Experiment results show that with this policy, the evolution can escape from many local minimums successfully.

3.3. how to eliminate isolated point

The isolated points of ψ are not welcome in evolution, especially for images contaminated by salt and pepper noise. To eliminate the isolated points, $L = \sum_{(p,q)\in S_{mn}} \psi_{mn}\psi_{pq}$, again plays an important role. Note that $L = 8$ implies ψ_{mn} and all its eight neighbors take the same value (Here we assume that ψ_{mn} is not at the edge of the image.). In this case, ψ_{mn} will become an isolated point when it changes its sign. Thus we can modify L to a larger positive number to remain ψ not changing. Analogously, $L = -8$ means the pixel (m, n) is an isolated point which should be eliminated, then we assign L a large negative number to enhance the internal force to shrink.

3.4. how to test convergence

As we discussed in section 3.2, we assign L a negative number when $L = 0$ to escape from local minimums. The side effect is that there will be some oscillations at the end of evolution. Namely, ψ of a few points will change from $+1$ to -1 periodically. This can be controlled by adding a counter to record the times of changing signs for each point. A point ψ_{mn} is not permitted to change again if its counter attains a pre-assigned maximum.

3.5. fast method to solve equation systems

The runtime of solving equation systems (5) is the main cost in each iteration. In this paper, we accelerate the procedure by decreasing the number of the pixels appearing in (5). In fact, since we only evolve boundary points, only gray values of the pixels near the curve would change much while the others change only a a little. So in most iterations, we only need to solve the equation on a narrow band along the curve and assume gray values of other pixels unchanged. Experiments show that it is an effective method.

3.6. algorithm

After discussion of implementation details, we present the entire algorithm.

Algorithm:

step0: Initialize ψ according to the initial curve and set $count_{ij} = 0, (1 \le i \le M, 1 \le j \le N)$ and $maxChange = 10$.

step1: Find all the boundary points according to **Definition**.

step2: Solve equation systems (5) by the fast method described in section 3.5 to obtain $f_{ij}, 1 \le i \le M, 1 \le j \le N$.

step3: For every boundary point (m, n), do:

(1) calculate $\hat{f}_{mn}$ according to equation (6)

(2) calculate

$$L = \sum_{(p,q)\in S_{mn}} \psi_{mn}\psi_{pq}$$

$$Len = \begin{cases} -1 & \text{if } L = 0, \\ 200 & \text{if } L = 8, \\ -200 & \text{if } L = -8, \\ L & \text{otherwise} \end{cases} \tag{7}$$

$$oldE = (f_{mn} - g_{mn})^2 - 2\nu Len + 2\mu \sum_{(p,q)\in S_{mn}} (f_{mn} - f_{pq})^2 (1 + \psi_{pq}\psi_{mn}) \tag{8}$$

$$newE = (\hat{f}_{mn} - g_{mn})^2 + 2\nu Len + 2\mu \sum_{(p,q)\in S_{mn}} (\hat{f}_{mn} - f_{pq})^2 (1 - \psi_{pq}\psi_{mn}) \tag{9}$$

where $oldE$ denotes the energy relevant to ψ_{mn} and $newE$ denotes the corresponding energy after changing sign of ψ_{mn}.

(3) if $newE < oldE$ and $count_{mn} < maxChange$, then change ψ_{mn} to $-\psi_{mn}$ and f_{mn} to $\hat{f}_{mn}$ and increase $count_{mn}$ by one.

step4: if some change has made in step3, go to step1; else finish.

4. EXPERIMENT RESULTS

In this section, some experiment results of the model will be illustrated. For the convenience of comparison, we choose almost the same test images and initial curves as those used in [7].

Fig.1 shows the evolution of three simple shapes with internal force alone. We can see that the evolution of shapes is not symmetric. The process depends on the order in which we visit the boundary points. But the final results are the same, that is, the shapes will shrink until they disappear.

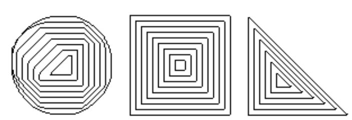
Fig.2 and Fig.3 show the evolution of curve with different initial position. The evolution of curve can capture the boundary automatically from arbitrary initial position. However, for different initial curve, there are some difference in parameter setting which is the same as in continuous model.

Comparing our results to those in [7], we find there are extraneous curves around the corners in Fig.5 of [7] because its algorithm has settled on to a local minimum. Because some special procedures have been adopted as described in section 3.2, we have successfully escaped from the local minimums as shown in Fig.3.

Fig.4 shows the evolution of curve on an image with four distinct foreground regions. There are skew stripes in background of the image which makes the segmentation difficult. Other models such as Mumford-Shah piecewise constant will be helpless for such image.

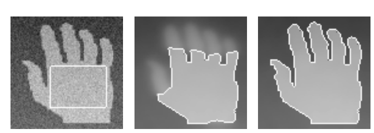
The image in Fig.5 has both outer and interior boundaries which can not be detected with only one curve, so two initial rectangles are used.

As shown in above results, although the segmenting curve has a tendency of beeline or polygon in the process of evolution, the segmentation result is accurate and the final curve is smooth.



(a) (b) (c)

Fig. 1. evolution of some simple shapes with internal force



(a) (b) (c)

Fig. 2. evolution from inside to outside with $\mu = 2, \nu = 150$ and (c) is the result after 37 iterations

(a) (b) (c)

Fig. 3. evolution from image corner to center with $\mu = 15, \nu = 150$ and (c) is the result after 122 iterations

(a) (b) (c)

Fig. 4. evolution with $\mu = 2, \nu = 150$ and (c) is the result after 44 itertions

For the convenience of comparison, we have also implemented traditional Mumford-Shah piecewise model. We use narrow band evolution method. And in order to reduce the running time, reinitialization and extension of velocity are done only on a narrow band. All experiments are run on a 2.4GHz Intel Pentium4 CPU with 256M memory and the programs are written in C language together with Matlab. The comparison is shown in Table1.

Table1: Comparison of segmentation time

Fig	Size	traditional model	discrete model
2	92*92	44.60s	1.90s
3	92*92	143.78s	8.93s
4	91*91	216.3s	3.77s
5	160*140	189.60s	35.37s

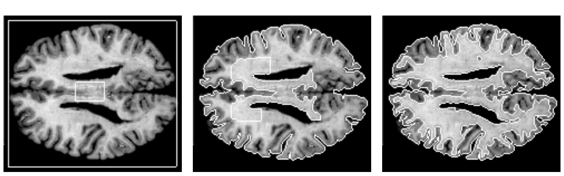
5. CONCLUSION

As a conclusion, we will make a comparison on various aspects with traditional continuous model and the two methods in [4] [5].

As shown in Table2, different from the work in [4] and [5], we have presented an entirely discrete Mumford-Shah piecewise smooth model on lattice. There is neither variational method nor PDEs in the model. The complete analysis and comparisons would be given in a forthcoming paper.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr Yonggang Shi for informing us the similarity between the first version of this paper and the work in [4] which results in this revised version.



(a) (b) (c)

Fig. 5. evolution with $\mu = 7, \nu = 80$ and (c) is the result after 140 iterations

Table2

	traditional model	method in [5]	method in [4]	discrete model
energy functional	continuous functional(1)	continuous functional(1)	continuous functional(1)	degraded functional(4)
smooth function	Euler-Lagrange function(2)	Euler-Lagrange function(2)	Euler-Lagrange function(2)	linear function(5)
calculation of velocity	yes	yes	no	no
extension of velocity	yes	yes	no	no
calculation of changed energy	no	no	yes	yes
level set function	continuous	discrete	discrete	discrete
evolution of level set	evolve by (3)	policy based	policy based	policy based
reinitialization	yes	no	no	no

7. REFERENCES

[1] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Commun. Pure Appl. Math.*, vol. 42, no. 4, pp. 577–685, 1989.

[2] T. Chan and L. Vese, “A level set algorithm for minimizing the Mumford-Shah functional in image processing,” in *Proc. IEEE workshop on variational and level set methods in computer vision 2001*, 2001, pp. 161–168.

[3] D. Mumford, “Pattern theory: a unifying perspective,” in *Proc. First European Congress of Mathematics*. 1994, pp. 187–224, Springer-Verlag.

[4] B. Song and T. Chan, “A fast algorithm for level set based optimization,” *Technical report, UCLA*, vol. CAM 02-68, 2002.

[5] Yonggang Shi and William Clem Karl, “A fast level set method without solving PDEs,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, vol. 2, pp. 97–100.

[6] T. Chan and L. Vese, “Active contours without edges,” *IEEE Trans. Image Processing*, vol. 10, pp. 226–277, Feb 2001.

[7] A. Tsai, A. Yezzi, and A. Willsky, “Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification,” *IEEE Trans. Image Processing*, vol. 10, pp. 1169–1186, Aug 2001.