Snake based Unsupervised Texture Segmentation using Gaussian Markov Random Field Models

Sasan Mahmoodi[1], Steve Gunn[1]

[1]School of Electronic and Computer Science, Building 1, Southampton University, Southampton, SO17 1BJ, UK,
Emails: {sm3,srg}@ecs.soton.ac.uk

Abstract: A functional for unsupervised texture segmentation is investigated in this paper. An auto-normal model based on Markov Random Fields is employed here to represent textures. The functional investigated here is optimized with respect to the auto-normal model parameters and the evolving contour to simultaneously estimate auto-normal model parameters and find the evolving contour. Experimental results applied on the textures of the Brodatz album demonstrate the higher speed of convergence of this algorithm in comparison with a traditional stochastic algorithm in the literature.

Keywords: *Unsupervised Texture Segmentation; Gaussian Markov Random Field Model; Mumford-Shah Model; Auto-normal Model; Maximum Likelihood;*

I. Introduction

Unsupervised texture segmentation is a challenging and demanding task in computer vision and image processing. This is due to the fact that image segmentation and the estimation of model parameters must be performed simultaneously. One of the most popular models employed in texture segmentation is Markov random field (MRF) models. There are numerous unsupervised texture segmentation algorithms based on Markov random fields in the literature (e.g. see [1] [2] [3] [4] [5] [6] [7]). Gaussian Markov Random Field (GMRF) model is used in [1] for unsupervised texture segmentation. A multi-resolution scheme is also employed in the literature (e.g. see [2] [4] [3]) to improve the quality of the texture segmentation. The most likely number of classes of textures in an image is estimated in [5] according to maximum a posteriori (MAP) criterion. Double Markov Random Fields (DMRF) is employed in [6] in a Bayesian framework for unsupervised segmentation of textures. The work of Kato *et al.* [7] is an example of methods combining color and texture features in a Bayesian framework via simulated annealing [8].

In this paper, we focus on GMRF models used for the unsupervised segmentation of texture images. Snake algorithm via gradient descent method is exploited here to perform segmentation. Maximum likelihood (ML) is also employed to estimate the GMRF model parameters. The rest of the paper is structured as follows. The theory is discussed in section II. Section III presents experimental results. Conclusions are drawn in section IV.

II. Texture Model and Segmentation

A functional based on Mumford-Shah (M-S) model [9] is proposed here to minimize the contour length and to maximize likelihood functions inside and outside the evolving contour. We assume that the lattice inside and outside the contour are Markov random fields [10]. The evolving contour is represented implicitly by shapes as described in [11]. A GMRF model also known as auto-normal model (for further details see e.g. [1]) is considered for textures here. The model parameters are therefore estimated by a maximum likelihood scheme (ML) in each iteration. The functional is also minimized with respect to the evolving contour to produce a flow to iteratively derive the contour to the desirable solution. Let us initially consider a site i in the input texture image. We assume that the image is Markovian, i.e. for a neighborhood $\mathrm{N}_i \subset R^i = \Omega - \{(x_i, y_i)\}$ of this site (where Ω is image domain i.e. the set of all sites in the image), the conditional probability density P only depends on the pixels in the neighborhood N_i, i.e.:

$$P_N(g(x_i, y_i)) = P(g(x_i, y_i) \big| g(x_{\mathrm{N}_i}, y_{\mathrm{N}_i})) \qquad (1)$$

A second order neighboring system is assumed for N_i in this paper. Pair-wise cliques are also assumed in this neighboring system. For such pair-wise cliques, the conditional probability density is given by equation (2):

$$P_N\left(g(x_i,y_i),\boldsymbol{\alpha},\lambda,\sigma\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{\left(g(x_i,y_i)-\lambda-\sum_{j\in N_i}\alpha_j\left(g(x_i^j,y_i^j)-\lambda\right)\right)^2}{2\sigma^2}\right] \quad (2)$$

where σ is the conditional standard variation, $\boldsymbol{\alpha}=(\alpha_j)$ for $j\in \mathrm{N}_i$ and λ is the mean value of pixels. In equation (2), the j^{th} neighboring pixel location for the i^{th} pixel is represented by (x_j,y_j). We therefore propose to find the minimum of the following functional for texture segmentation:

$$E(\boldsymbol{\alpha},\lambda,\sigma,\boldsymbol{\beta},\nu,\rho,\chi)=\iint_\Omega\{\mu|\nabla\chi|-\ln P_N(g(x,y),\boldsymbol{\alpha},\lambda,\sigma)\chi(x,y)$$
$$-\ln P_N(g(x,y),\boldsymbol{\beta},\nu,\rho)(1-\chi(x,y))\}\,dxdy \quad (3)$$

where $\chi:\Omega\to\{0,1\}$ and μ are the shape characteristic function representing the evolving contour and a constant respectively. The first term in functional (3) is responsible for contour length minimization [9,12]. The second and third terms, i.e. $-\ln P_N(g(x,y),\boldsymbol{\alpha},\lambda,\sigma)$ and $-\ln P_N(g(x,y),\boldsymbol{\beta},\nu,\rho)$ are minimized with respect to parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, λ, ν, σ, and ρ. This indicates that we are looking for a probability distribution function giving the observed data the greatest probability (maximum likelihood). In functional (3), an image is assumed to consist of two textures. Functional (3) can also be generalized for images containing more than two textures by employing a multi-phase shape scheme similar to the multi-phase level set framework proposed in [12]. All model parameters are assumed constant over all sites inside and outside of the evolving contour. In order to estimate the model parameters, functional (3) is minimized with respect to the model parameters $\boldsymbol{\alpha},\boldsymbol{\beta},\lambda,\nu,\sigma,$ and ρ leading to a maximum likelihood strategy. This is achieved by assuming that the first derivative of functional (3) with respect to the model parameters vanishes. This leads to a system of equations whose solution is the desirable model parameters in each iteration, i.e. if the input image is considered as an $M\times N$ discrete grid, then the system can be written as:

$$MN\left(1-\sum_{j\in N_i}\alpha_j\right)\lambda=\sum_i\left(g(x_i,y_i)-\sum_{j\in \mathrm{N}_i}\alpha_j g(x_j,y_j)\right) \quad (4)$$

$$-\lambda\sum_i g(x_k,y_k)+\sum_i g(x_k,y_k)\left(\sum_{j\in \mathrm{N}_i}\alpha_j\left(g(x_j,y_j)-\lambda\right)\right)$$
$$=\sum_i g(x_i,y_i)g(x_k,y_k),\qquad k\in \mathrm{N}_j \quad (5)$$

$$\sigma^2=\sum_i\left(g(x_i,y_i)-\lambda-\sum_{j\in \mathrm{N}_i}\alpha_j\left(g(x_j,y_j)-\lambda\right)\right)^2 \quad (6)$$

We notice that equations (4) to (6) are derived for the model parameters of the texture inside the contour. Similar equations can be derived for the model parameters of the texture outside the contour. We assume that model parameters are constant over the region inside or outside of the contour. But the model parameters inside the contour are different from outside ones. It is important to note that equations (4)-(6) are not linear. To find the solution of equations (4)-(6), we employ an iterative method by initially calculating the mean of inside and outside of the evolving contour to find an initial estimate for λ. Then equations (5) are linearly solved for α_j s ($j=1,2,...,\mathrm{N}_j$). Equation (4) is then solved to find better estimates for λ. Equations (5) are again solved to update α_j s according to the latest value of λ. This iterative method ends when the distance between the calculated parameter values in two consecutive iterations becomes less than a certain threshold. Finally the conditional variance is calculated by using equation (6). To calculate the evolving contour iteratively, functional (3) is minimized with respect to χ by arriving at Euler-Lagrange equation:

$$\frac{\partial\chi}{\partial t}=\mu\nabla\cdot\frac{\nabla\chi}{|\nabla\chi|}+\ln\left(\frac{P_N\left(g(x,y),\boldsymbol{\alpha},\lambda,\sigma\right)}{P_N\left(g(x,y),\boldsymbol{\beta},\nu,\rho\right)}\right) \quad (7)$$

$\chi(x,y,0)=\chi_0(x,y)$ in Ω and $\frac{1}{|\nabla\chi|}\frac{\partial\chi}{\partial\mathbf{n}}=0$ on $\partial\Omega$.

where t is the virtual time along which the contour evolves to the desirable solution.

In each iteration, equations (4) to (6) are employed to update the model parameters and equation (7) is exploited to calculate the evolving contour for the corresponding iteration according to the model parameters updated in (4) to (6). For regularization, a Dirac filter is applied on shape characteristic function in each iteration [11], i.e. the following C^∞ regularizing function is convolved with the shape characteristic function in every iteration (see [11] for more details):

$$G_\varepsilon(x,y)=\frac{\varepsilon}{\pi\left(x^2+y^2+\varepsilon^2\right)} \quad (8)$$

where ε is the regularizing parameter. A semi-implicit finite difference scheme [12] is also employed to discretize equation (7).

III. Experimental Results

Let us consider the texture image consisting of two textures from Brodatz album as shown in figure (1). The unsupervised texture segmentation method proposed here is applied to this image. Figures (1-a) to (1-f) depict a few iterations of this algorithm. In this experiment, the time increment (in contour evolution) $\Delta t = 1$, $\mu = 0.08$ and a 10 x 10 Dirac filter with a regularizing parameter $\varepsilon = 4$ for regularization is chosen. We also observe that the final result of segmentation does not depend on the initial values of the auto-normal model parameters. However the final segmentation result depends on the initial contour. The algorithm may fall into local minima for some initial contours. We compare our algorithm with a multi-resolution Gaussian Markov Random Field (GMRF) with a second order neighborhood (auto-normal model)[2][3][4]. A simulated annealing algorithm is employed to maximize *a posterior* (MAP) probability [8]. In such an algorithm, the likelihood function is a probability function based on the auto-normal model (Gaussian Markov random field). The *prior* probability is a Gibbs distribution function modeling the segmented image (labels) assumed to be a hidden Markov random field. In such a framework, labels are considered to be an Ising model. The *posterior* probability is then calculated in a Bayesian framework. The simulated annealing technique is therefore used to maximize this *posterior* probability. The segmentation target of the texture image of figure (1) is depicted in figure (2-a). The segmentation result obtained by the multi-resolution GMRF algorithm using a simulated annealing optimization method in which virtual temperature decreases logarithmically [8], is shown in figure (2-b). The result shown in figure (2-b) is obtained after 18000 iterations (10000 iterations with half resolution and 8000 iterations with full resolution). In a PC with a 2.00 GHz microprocessor, each iteration for the simulated annealing with half and full resolution takes 3.9 and 16.2 seconds respectively. The total time elapsed by the CPU is therefore 168600 seconds. As shown in figure (2-b), there are some pixels which are incorrectly segmented in simulated annealing algorithm. This is due to the fact that the algorithm is stopped before the temperature drops to absolute zero. We notice that the temperature approaches zero logarithmically, it would therefore take several days for the simulated annealing to get to zero temperature. This is the main disadvantage of the simulated annealing algorithm. The segmentation result of the snake algorithm proposed in this paper is illustrated in figure (2-c) in a binary format. It takes 49 iterations (each iteration 11.2 seconds) for the snake-based algorithm proposed here to segment the textured image. The total time spent by CPU to implement the snake-based algorithm is therefore 548.8 seconds. Only single resolution scheme (full resolution) is used to implement the minimization of functional (3). In this paper, the segmentation error between the segmentation target and the binary segmented image obtained by both algorithms is calculated by the following term:

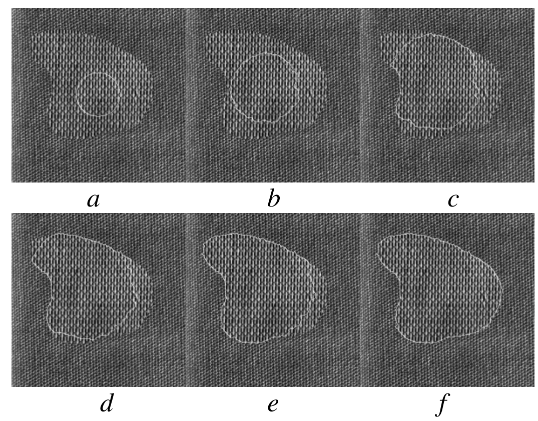


Figure 1: Contour evolutions of the algorithm proposed here applied on a 200 x 200 texture image from Brodatz album a) Initial contour, b) 11th iteration, c) 21st iteration d) 31st iteration e) 41st iteration f) 49th iteration

$$Error = \sum_x \sum_y \left| T(x,y) - S(x,y) \right|^2 \qquad (9)$$

where $T(x,y)$ and $S(x,y)$ are the segmentation target and the binary segmented image obtained by the algorithms investigated in this paper respectively. The average error terms calculated by equation (9) and normalized by the total number of pixels in the image for segmented images over a range of Brodatz texture images are tabulated in table 1.

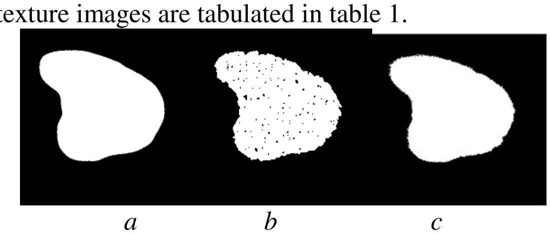


Figure (2): Segmentation results in binary format, a) segmentation target b) auto-normal model using multi-resolution simulated annealing c) auto-normal model using snake

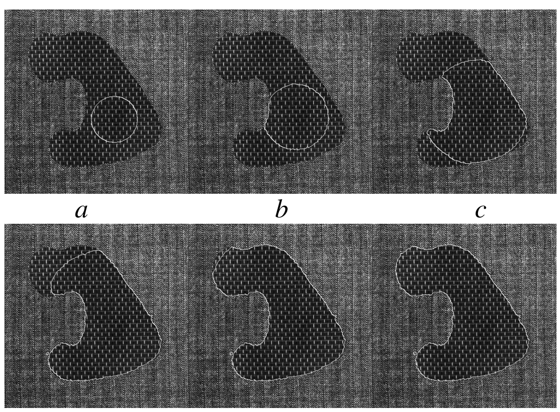
Figure (3) shows the contour evolutions of the snaked-based unsupervised texture segmentation algorithm proposed here to segment a 400 x 400 textured image consisting of two other textures from Brodatz album. The algorithm converges to the solution after 87 iterations. The superiority of the snake-based algorithm proposed here is demonstrated by the higher speed of segmentation as explained earlier.

Table 1: A quantitative comparison in the normalized segmentation error over a range of texture images for the simulated annealing and snake-based algorithm

	Simulated Annealing	Snake
Normalized Average Error (%)	1.88%	1.51%

IV. Conclusion

A snake-based Gaussian Markov Random Field (auto-normal) model is proposed in this paper for unsupervised texture segmentation. The method proposed here performs the texture segmentation much faster than the traditional stochastic auto-normal model implemented by a multi resolution scheme based on simulated annealing. Only a single resolution scheme is required for the snake-based segmentation method to converge.



d *e* *f*

Figure (3): Contour evolutions of the unsupervised texture segmentation algorithm proposed here to segment a 400 x 400 textured image a) first iteration, b) 11th iteration, c) 31st iteration, d) 51st iteration, e) 71st iteration, f) 87st iteration.

V. Acknowledgement

This work was supported in part by the IST program of the European Community, under the PASCAL2 Network of Excellence, the IST-2007-216886 and PinView project with grant number 216529.

VI. References

[1] B.S. Manjunalh, and R. Chellapa, "Unsupervised Texture Segmentation using Markov Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 13, No. 5, pp. 478-482, 1991.

[2] S. Krishnamachari, and R. Chellapa, "Multi-resolution Gauss Markov Random Field Models for Texture Segmentation", *IEEE Transactions on Image Processing,* Vol. 6, No. 2, pp. 251-267, 1997.

[3] R.Wilson, and C.T. Li, "A Class of Discrete Multi-Resolution Random Fields and its Application to Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, No.1, pp. 42-56, 2002.

[4] M.L. Corner, and E.J. Delp, "Segmentation of Textured Images using a Multi-Resolution Gaussian Autoregressive Model", *IEEE Transactions on Image Processing,* Vol. 8, No. 3, pp. 408-420, 1999.

[5] S.A. Barker, P.J.W. Rayner, "Unsupervised Image Segmentation using Markov Random Field Models", *Pattern Recognition,* Vol. 33, pp. 587-602, 2000.

[6] D.E. Melas, S.P. Wilson, "Double Markov Random Fields and Bayesian Image Segmentation", *IEEE Transactions on Signal Processing,* Vol. 50, No. 2, 2002

[7] Z. Kato, T.C. Pong, "A Markov Random Field Segmentation Model for Color Textured Images", *Image and Vision Computing,* Vol. 24, pp. 1103-1114, 2006.

[8] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. Vol. 6, No. 6, pp. 721-741, 1984.

[9] D. Mumford, and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems", *Communications for Pure and Applied Mathematics,* Vol. 42, No. 4, pp. 577-688, 1989.

[10] G. Winkler, "*Image Analysis, Random Fields and Dynamic Monte Carlo Methods",* Springer-Verlag, Berlin, 1995.

[11] S. Mahmoodi, "Shape-based Active Contours for Fast Video Segmentation", *IEEE Signal Processing Letters,* Vol. 16, No. 10, pp.857-860, 2009.

[12] L.A. Vese and T.F. Chan, "A Multiphase Level Set Framework for Image Segmentation using the Mumford-Shah Model" *International Journal of Computer Vision,* Vol. 50, No. 3, pp. 271-293, 2002.