Reducing the Computational Complexity of Markov Random Fields within an Arbitrarily Large Texture Label Space

Chang-Tsun Li
Department of Electrical Engineering
Chung Chen Institute of Technology
Tahsi, Taoyuan, Taiwan, 33509 ROC.
ctli@cc04.ccit.edu.tw

Abstract

This work proposes a novel idea, called SOIL, for reducing the computational complexity of the *maximum a posteriori* optimization problem using Markov random field by exploiting the local characteristics so that the searching in a virtually infinite label space is confined in a small finite space. Globally the number of labels allowed is as many as the number of image sites while locally the optimal label is sampled from a space consisting of the labels assigned to the 4-neighbour plus a random one. Neither the prior knowledge about the number of classes nor the estimation phase of the class number is required in this work. The proposed method is applied to the problem of texture segmentation and the result is compared with those obtained from conventional methods.

Keywords: Markov Random Field, Stochastic Relaxation, Simulated Annealing, Texture Segmentation, Bayesian Estimation.

1. Introduction

Due to its local characteristics, also known as *Markovianity*, Markov Random Field (MRF) has become one of the most popular approaches to optimization problems in a wide variety of areas in general [6][7][10][12], and texture segmentation in particular [1][3][8][9][11][13]. The main attraction of the local characteristics of MRF resides in its merit allowing a global optimization problem to be simplified and solved locally, whereby the computational cost is minimized. In the context of texture segmentation, the optimization is basically a process seeking the optimal labeling of the image pixels/sites. *Markovianity* allows the label selection of a site to be conditioned explicitly on the local interaction between the site and its neighbors within a well-defined neighborhood system without involving all the sites of the images.

For an image of size $M \times M$ pixels/sites, if we define some symbols as follows:

$S = \{(i,j) \mid (i,j)$ is the coordinate of a site in the image; $1 \le i, j \le M\}$

$\Gamma = \{\gamma \mid \gamma$ is a class label; $1 \le \gamma \le L\}$

$\lambda = \{\lambda_s \mid s \in S$ and $\lambda_s \in \Gamma$, is a class label of site $s.\}$

$\Lambda = \{\lambda \mid \lambda$ is a label configuration of the image$\}$,

then the image can be seen as a sequence of Markov Random Fields, i.e., a family of random variables $\lambda = \{\lambda_s \mid s \in S\}$ with respect to a neighborhood system N_s, in which each random variable λ_s takes a value in Γ, if and only if the following two properties are satisfied:

$$P(\lambda) > 0 \qquad \forall \lambda \in \Lambda \tag{1}$$

$$P(\lambda_s \mid \lambda_{S-\{s\}}) = P(\lambda_s \mid \lambda_{N_s}) \tag{2}$$

where $\lambda_{S-\{s\}}$ is the configuration of the set $S-\{s\}$ and λ_{N_s} is the local configuration of neighborhood N_s of site s. Besag [2] reasoned that if Equation (1), the prior, is satisfied, the joint probability $P(\lambda)$ of any random field is uniquely determined by its local conditional probability. This is characterized by Equation (2), which is the local characteristic of Markov Random Fields, called *Markovianity*. From Equation (2) we can see that a Markov Random Field is a conditional probability measure of a random variable at a site depending only on the interactions with its neighbors. This local characteristic implies that the statistical structure of the image is essentially localized within the neighborhood system.

The Hammersley and Clifford expansion [2] has established the theorem of MRF-Gibbs equivalence that λ is *a sequence of Markov random fields on S with respect to* N_s *if and only if* λ *is a set of Gibbs random fields on S with respect to* N_s. Readers are referred to [2] for the proof of the theorem. The importance of this theorem is that it provides a simple way of formulating the joint probability by specifying the clique potential functions appropriately chosen for desired behavior of the system. In the context of texture segmentation, we can treat an image as a sample of a set of Gibbs random fields with respect to a neighborhood system N_s if and only if its label configuration obeys a Gibbs distribution [4][10]. A Gibbs distribution is denoted, for a given set of observed data X, as

$$P(\lambda) = \frac{1}{Z} e^{-\frac{U(\lambda,X)}{T}} \tag{3}$$

where T is the *pseudo temperature* and the normalizing factor Z is the *partition function* defined as

$$Z = \sum_{\lambda \in \Lambda} e^{-\frac{U(\lambda,X)}{T}} \tag{4}$$

For each configuration λ, associated interaction energy $U(\lambda, X)$ is defined as

$$U(\lambda, X) = \sum_{c \in C} V_c(\lambda, X) \tag{5}$$

where $V_c(\lambda, X)$, called *potential*, is the interaction among the sites in clique C. The higher the potential is, the more repulsive the sites within the clique are against each other, i.e. it is more likely that they belong to different classes, and vice versa.

While a Gibbs random field is characterized by its global property, the Gibbs distribution, a MRF model is characterized by its local characteristic in Equation (2). With the theorem of MRF-Gibbs equivalence and *markovianity*, instead of formulating $P(\lambda_s \mid \lambda_{S-\{s\}})$, the optimization problem is approached with a MRF model specified as

$$P(\lambda_s \mid \lambda_{N_s}) = \frac{1}{Z_s} e^{-\frac{U(\lambda_s, \lambda_{N_s}, X_{N_s})}{T}} \tag{6}$$

where the *partition function* Z_s is defined as

$$Z_s = \sum_{\lambda_s \in \Gamma} e^{-\frac{U(\lambda_s, \lambda_{N_s}, X_{N_s})}{T}} \tag{7}$$

and

$$U(\lambda_s, \lambda_{N_s}, X_{N_s}) = \sum_{c \in C} V_c(\lambda_s, \lambda_{N_s}, X_{N_s}) \tag{8}$$

where X_{N_s} represents the observed data at the sites within the neighborhood N_s. Equation (6) suggests that the estimate of the class label, λ_s, at site s is determined locally through the interaction between site s and its neighborhood N_s.

MRF models are often used in conjunction with statistical estimation, such as stochastic relaxation, so as to formulate objective functions in terms of established optimization principles, e.g., *Maximum a Posteriori* (MAP). Relaxation is an iterative process for reducing the ambiguity of labeling by minimizing an energy function in which contextual constraints are encoded [14]. MAP is one of the most often adopted statistical criteria for optimization and in fact, has been the most popular choice in MRF texture analysis [1][3][4][7][9][13]. Equation (6) also suggests that the distribution function depends on temperature T. At low temperatures, the posterior distribution concentrates on the label which associates with the lowest interaction energy $U(\lambda_s, \lambda_{N_s}, X_{N_s})$, whereas at high temperature the posterior distribution is essentially flat and no specific label is particularly favored.

Stochastic relaxation, in a sense of simulated annealing, is carried out by reducing the temperature T iteratively. According to Equation (6), as the temperature T decreases at each iteration, the probabilities of the configurations associated with lower energy become larger and those with higher energy are reduced. Eventually (hopefully), the system will settle in the configuration with lowest energy. Geman and Geman [4] suggested

$$T = \frac{C}{\log(1+i)} \tag{9}$$

as the annealing schedule, where i is the number of iterations, C is a sufficiently large constant and T is the temperature function of i. Since the 'energy landscape' over Markov random fields is usually non-convex, in order to avoid local minima it is preferable to start at high temperature and as the relaxation proceeds, T is decreased gradually in the manner of physical annealing. This MAP-MRF labeling scheme is known as Gibbs Sampler when employed in conjunction with simulated annealing [4].

Despite the advantages of Markov Random Fields, when they are used in conjunction with stochastic relaxation scheme, the convergence rate requires a serious consideration, because slow convergence rate incurs high computational complexity. Therefore, it is our intention to investigate this problem and propose a solution to it. The rest of the work is organized as follows. Issues regarding the computational complexity of MRF's are addressed in Section 2. Section 3 reviews two related works and discusses their advantages and drawbacks. A new approach to reducing the computation complexity of MAP-MRF framework is also proposed and the performance of it is compared with those of the related two. Application of the proposed approach to texture segmentation is demonstrated in Section 4. Section 5 concludes the work.

2. Issues about the Computational Complexity of MRF's

As far as the convergence rate is concerned, there are two major issues call for attention:

Issue 1: Starting temperature T

There is a dilemma regarding the choice of the starting temperature T of Equation (6). To avoid converging to local minima, higher starting temperature is preferred since it provides a more moderate cooling schedule. However, this incurs computational penalty and slows the convergence rate down. On the other hand, to accelerate convergence, lower starting temperature is more desirable since it provides a more rapid cooling schedule. However, this is more likely to make the algorithm premature and get trapped in local minima. Therefore, trade-off has to be made between fast convergence rate and high segmentation quality by choosing an appropriate starting temperature. In the context Gibbs sampling, the starting temperature is dictated by the constant C in Equation (9). Unfortunately, deciding the best value of C is non-trivial and heuristic, and the range of the interaction energy $U(\lambda_s, \lambda_{N_s}, X_{N_s})$ and the number of labels have to be taken into account so that C can be better chosen.

Issue 2: Size of label space L

From the partition function in Equation (7), we know that the posterior probability has to be calculated for all L texture class labels in Γ. Also from Equation (6), it is easy to see that the larger L is, the smaller the probability of each label to be selected is. When L is relatively large, even for the label with lowest interaction energy (cost), its opportunity of being selected is slim because the *posteriori* probability of it is overwhelmed by the big population of labels. This means that if L is large, the algorithm takes more iterations or longer time before the optimal configuration can gradually emerge from the rest. In its extreme case when $L \to \infty$, the distribution in Equation (6) becomes exactly uniform and none of the labels is preferred, therefore, the label of each site will change randomly. On the other hand, if L is given a value smaller than the actual number of texture regions, the image will inevitably be under-segmented.

Another observation we made is that when a single site s is

significantly different from all its neighbors, all the labels *not* assigned to any of its neighbor will have the same probability which is significantly larger than those of the labels assigned to any of its neighbor. As a result, the label of s will be very likely to change randomly among the labels *not* assigned to any neighbor of s and the convergence rate is therefore retarded.

Based on the above two observations, it is therefore desirable to keep the size of label space to its minimum without compromising segmentation quality and reducing the convergence rate.

Unfortunately, the number of texture classes contained in the underlying image is usually unknown. This uncertainty complicates the choosing of value for L. The solution may be either to pick a relatively large value for L or to perform an estimation with a pre-process. However for the former, the drawback has been clearly stated. For the later, estimating the number of textures in most cases requires prior knowledge about the textures or hypothesis. Basically, the estimation itself is an ill-posed problem and usually heuristic, thus is unreliable. Therefore some measure is needed to tackle this issue.

As discussed in *Issue 1*, deciding the best value of the starting temperature involves the consideration of the range of the interaction energy $U(\lambda_s, \lambda_{N_s}, X_{N_s})$. This is not the emphasis of this work. The reason we raise this issue is because that it is somehow related to *Issue 2*, and there is always an uncertainty involved in deciding the starting temperature T or constant C of Equation (9) for different values of L. For example, either when $L \to \infty$ while C is kept constant or when $C \to \infty$ while L is kept constant, the distribution in Equation (6) becomes exactly uniform with none of the labels preferred. Therefore, for larger L, it may seem sensible to adopt lower value for C so as to associate the optimal label with a higher probability. This is because lower value of C (or T) magnifies the difference between $U(\cdot)$'s of different configuration, therefore, reduces the influence of the majority of the less favored lables on the optimal one. However, by doing so, the algorithm is more likely to premature and gets trapped at local minima. Also, depending on applications, the value of L may be image-dependent. This requires a new tuning of C when a new image with different value of L is presented. It is therefore desirable to divorce C and L so that the value of them can be decided independently. This is achieved in the present work and will be discussed in Section 3.2.

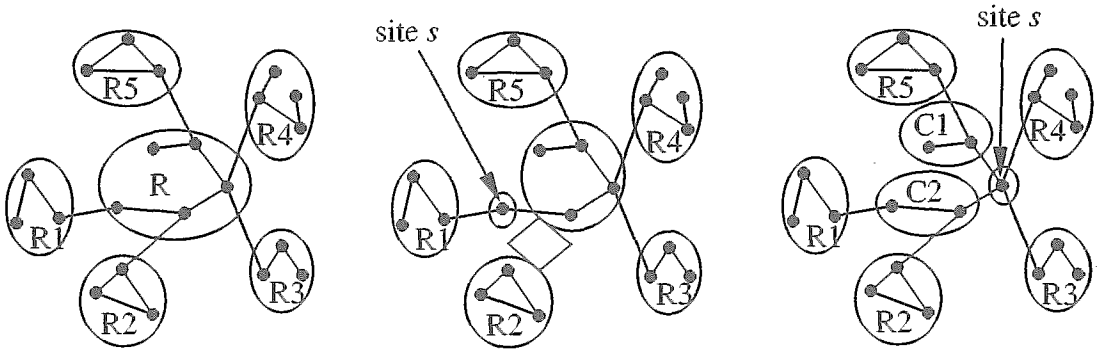
3. Approach to Reducing the Computational Complexity

Although the heavy complexity in calculating the partition function in Equation (7) is well recognized, few researchers proposed solutions to the problem [1][13].

3.1 Previous Approaches and Their Drawbacks

Andrey and Tarroux [1], in order to reduce the computational complexity, have proposed a method to calculate the normalizing factor by summing over all the terms (energy functions) of the partition function, *each approximated by a simpler mathematical expansion*. However, their approach does not fully exploit the local characteristics of the MRF's. Instead, their calculation of the partition function involves all the sites within the underlying image and all the elements in the label space Λ. The saving in computational cost is only achieved by finding a simpler expression of the energy function which requires less computational operations, not through a reduced but sufficient set of configurations.

Wang proposed a labeling method [13] which restricts the sampling on a reduced configuration space $C(S)$ consisting of a set of Partitions with Connected Components (PCC). A PCC is a configuration in which each region of homogeneous features is connected. Wang considers an image grid as a general graph with the image pixels being the sites of the graph. Figure 1 illustrates three PCCs, each represents one partition (configuration) of the graph (image grid). In Figure 1, sites (nodes), representing image pixels, connected with edges are neighbors. Connected sites confined in the ellipses form homogeneous regions, therefore, there are 6 regions partitioned in Figure 1(a) while there are 7 and 8 regions in Figure 1(b) and (c), respectively. At the first step of the segmentation process, Wang's algorithm is run directly on the graph of image pixels S attempting to minimize the energy function over $C(S)$ – a subset of Λ, and will arrive at a local minimum partition PCC_0. Then in order to merge and grow the partitioned regions of PCC_0, a new graph S_1 is defined by considering each connected region (component) of PCC_0 as a single site of S_1. The segmentation algorithm is then repeated on graph S_1 attempting to minimize the energy function by searching the minimum partition in the new and reduced configuration space $C(S_1)$ — a subset of $C(S)$, until it converges to a new minimum partition PCC_1. This process repeats until the algorithm can no longer obtain any change. The main idea of this algorithm is that, by considering each region of PCC_k formed at



step k as a single site at step $k+1$.

Now let us look at how $C(S_k)$s are formed. Like most segmentation algorithms, Wang's method also updates the configuration a single site at a time. Therefore, for each visit to a site, the candidate configurations, called the *set of elementary moves (SEMs)* by Wang, varies slightly only on the position of the site being visited.Wang defines two types of set of elementary moves: *SEM of the first type* referred to as *SEM1* and *SEM of the second type* referred to as *SEM2*.To define *SEM1* and *SEM 2*, Wang first defines a particular partition $[\lambda]_s$, associated with partition λ, as follows. Let site $s \in S$ and $\lambda = (R, R_1, R_2, \Lambda, R_n) \in C(S)$, where R is the region containing s. If isolating s divides R into connected components $\{s\}$, C_1, C_2,..., C_t, then $[\lambda]_s$ is defined as $[\lambda]_s = \{\{s\}, C_1, C_2, \Lambda, C_t, R_1, R_2, \Lambda, R_n\}$. Figure 1(b) and (c) are two different partitions $[\lambda]_s$s associated with partition λ in Fgiure 2(a). Now *SEM1* and *SEM2* can be defined as follows.

Definition of SEM1: for site $s \in S$ and $\lambda \in S(C)$, let R denotes the region containing site s,

a) if R contains only s or $R - \{s\}$ is connected (as shown in Figure 1(b)), then *SEM1* is the set consisting of $[\lambda]_s$ and the PCCs obtained from $[\lambda]_s$ by combining s with the regions adjacent to s in $[\lambda]_s$ and leaving other regions unchanged (e.g., Figure 1(b)).

b) if $R - \{s\}$ is not connected (as shown in Figure 1(c)), then $SEM1 = \{\lambda\}$.■

Definition of SEM2: for site $s \in S$ and $\lambda \in S(C)$, *SEM2* is the set consisting of $[\lambda]_s$ and the PCCs obtained from $[\lambda]_s$ by combining s with the regions in $[\lambda]_s$ adjacent to s, such that there is no pair of regions among them which are neighbors, and leaving other regions unchanged. ■

For example, let $([\lambda]_s, \{R, R_1, R_2, \Lambda, R_k\})$ denotes the PCC obtained from $[\lambda]_s$ by combining s with the regions R_1, R_2, Λ, R_k, and leaving other regions unchanged, then the *SEM2* of Figure 1(c) is

$$SEM2 = \{\lambda, ([\lambda]_s, \{s, C_1\}), ([\lambda]_s, \{s, C_2\}), ([\lambda]_s, \{s, R_3\}), ([\lambda]_s, \{s, R_4\}), ([\lambda]_s, \{s, C_1, R_3\}), ([\lambda]_s, \{s, C_1, R_4\}), ([\lambda]_s, \{s, C_2, R_3\}), ([\lambda]_s, \{s, C_2, R_4\}), ([\lambda]_s, \{s, C_1, C_2, R_3\}), ([\lambda]_s, \{s, C_1, C_2, R_4\})\}$$

Therefore, when a site is being visited, instead of sampling the label of site s over L labels, only the labels such that, for all PCCs, PCC $\in SEM1 \cup SEM2$, are involved in the sampling. However, although *SEM1* is small, which incurs less computational cost, the relatively limited choices is likely to yield undesirable partitions. On the other hand, although *SEM2* provides more choices which allows the algorithm to avoid undesirable partitions, it is relatively large and its construction is complex as shown in the above example, which certainly imposes more overhead on the algorithm. It is important to note that the interaction energy function depends not only on the configuration but also on the observed data (features). However, when the shape of regions changes, the regional features (e.g., mean gray level, variance, etc.) have to be re-evaluated. This requirement consumes computational resources too. Therefore, apart from the complication of constructing its configuration elements, another problem regarding the use of SEM2 is that each element of it adds an additional computation load for feature re-evaluation on the algorithm. To optimize the performance, at the early steps of the segmentation when the number of sites is large and the size of the connected regions is small, only *SEM1* is considered so as to form bigger regions and reduce the number of sites for the next graph as soon as possible. After a few steps when the number of sites in the new graph is reduced to some extent, *SEM2* is used to replace *SEM1* to offer more choices so as to bring out more changes and better segmentation.

To further reduce the computational complexity, based on the *conjecture of four-color problem* that four colors are sufficient to color a planar map of any number of regions [5], Wang suggests to allow only five labels being used throughout the segmentation process, therefore, the label space is reduced from L ($L > 5$) to 5.The reason he allows five, instead of 4, labels is because the conjecture of four-color problem has not been proved mathematically (although it has been proved by means of an intricate computer analysis in 1976 by Appel and Hagen [5]. This idea does set an upper bound of 5 on the number of labels to be involved in the segmentation process, however, the drawbacks are obvious: First, it does not attempt to assign the same label to the disjoint regions of the same features. Secondly, it does not try to avoid assigning the same label to the disjoint region of the different features. For example, if there are six significantly different regions contained in the image, at least two of the distinct regions will be assigned the same label.

In the next section, aiming at overcoming the aforementioned drawbacks of Wang's algorithm, a simpler, yet more flexible method, which fully exploits the local characteristics of the MRF's, is proposed to minimize the computational complexity within an infinite label space.

3.2 Our Method Using *SOIL*

To enable the algorithm to execute without supervision and to avoid estimating the number of textures contained in the underlying image, we allow as many labels as the number of sites to be used, i.e., in our method, $L = |S|$. This is virtually equivalent to adopting an infinite label space. However, for any neighborhood system N_s (4- and 8-neighborhood, etc.) employed, it is observed that actually, for all λ_s not used in the *4-neighborhoods* of site s, $P(\lambda_s \mid \lambda_{N_s})$'s are all equal. It does not matter what label, which is not in the 4-neighborhood of site s, is assigned to s. Therefore, there is no need to calculate $P(\lambda_s \mid \lambda_{N_s})$ for more than *one* λ_s which are not used in the 4-neighborhood of site s. Thus, *for any site S, we only calculate the probabilities of the current labels in the 4-neighborhoods of s and one randomly picked from the rest of the labels which are not assigned to the 4-neighbors.* That is to say that the candidate labels to be assigned to any site S is sampled from the Set Of Indispensable Labels (SOIL) denoted as

$$SOIL = \lambda_{4N_s} \cup \{\text{any one } \lambda_s \notin \lambda_{4N_s}\} \qquad (10)$$

where $4N_s$ stands for the 4-neighborhood of site s and λ_{4N_s} is the set of the current labels in the 4-neighborhood of s.

Although, globally, there are $|S|$ labels available, labels over any specific homogeneous region will unify through local interactions among sites as relaxation proceeds.

The advantages of this idea is threefold:

- It saves computational cost and solves the second issue mentioned in the previous section by minimizing the number of labels to be involved. For example, for the worst case in the first order neighborhood, assuming all the four neighbors have different labels from each other, there are only *five* (four used by the 4 neighbors plus a random one) labels involved. For the best case wherein all the neighbors are associated with the same label, only *two* labels are involved. This idea minimizes the computational complexity in two aspects: *locally*, the number of labels involved is reduced to the minimum; *globally*, the convergence rate is minimized because the growing rate of the optimal labels at different site is higher than the methods involving more than one labels which are not used in the neighborhood.
- Knowing the maximal number (five) of labels to be involved also helps us in deciding an appropriate value of constant C in Equation (9), since now C can be decided independent of L and based purely on the range of the interaction energy $U(\lambda_s, \lambda_{N_s}, X_{N_s})$. Therefore, *Issue 1* mentioned in the previous section is *partly* solved.
- This idea makes the prior knowledge about the number of textures or estimation phase of the classes number unnecessary, and enables the algorithm to work without supervision.

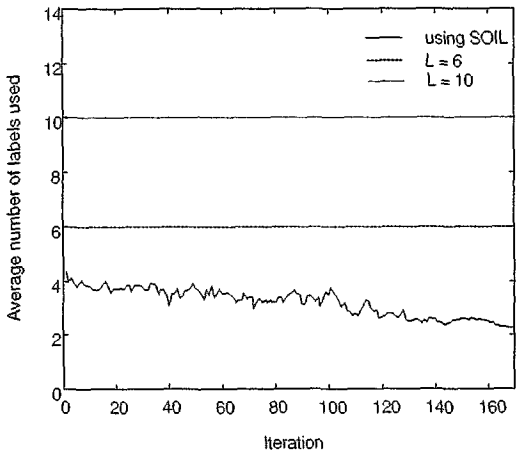


Figure 3. The average number of labels used at each iteration.

3.3 Performance Comparisons between PCC and SOIL

The performance comparisons between Wang's PCC method and our SOIL approach can be made as follows.

- Allowing at most 5 labels to be used *locally* in Wang's algorithm reduces computational cost, however the maximal number of labels allowed *globally* is also fixed at five. This limits the flexibility of the algorithm in 1) differentiating disjoint regions of different features and 2) assigning the same label to the disjoint regions with homogeneous features. For our SOIL method, the maximal number of labels allowed *locally* is five while it is L ($L = |S|$) *globally*. The superiority of SOIL to PCC is obvious: the computational cost is kept minimal while the flexibility is maintained.
- Apart from the flexibility, the computational cost of Wang's PCC algorithm using *SEM1* is equivalent to that of SOIL. However, *SEM1* is only used in the first few steps while the larger and more complicatedly constructed *SEM2* is used throughout the rest of the segmentation process. For SOIL, the number of labels involved at any point of the segmentation process is always less than or equal to 5 (actually, as we will see in the next section, the number of labels involved approaches 2 as the algorithm using SOIL iterates toward the final configuration). In addition, SOIL requires no complicated configuration construction like *SEM2* does.

4. Application of SOIL to Texture Segmentation

To demonstrate the merits of the proposed algorithm using SOIL, the proposed method and two conventional methods with $L = 6$ and $L = 10$, respectively, are applied to segment the textured image in Figure 2 (The white lines in Figure 2 are the superposed boundaries detected at the coarsest scale). The only difference among the three algorithms is the number of labels involved during the sampling at each site. Since SOIL can be used in conjunction with different energy functions and the main emphasis of this work is to demonstrate the advantages of using SOIL, therefore, the definition of energy function adopted in the three algorithms is not presented. Readers are referred to [9] for the detailed description of the definition. The average number of labels involved at each site with respect to iteration is plotted in Figure 3. The solid line is the average number of labels using SOIL while the other two are associated with the conventional methods. Without adopting SOIL, the number of labels remains constant throughout the optimization process, while adopting the proposed method, the algorithm is adaptive to the local configurations, therefore, the solid curve in Figure 3

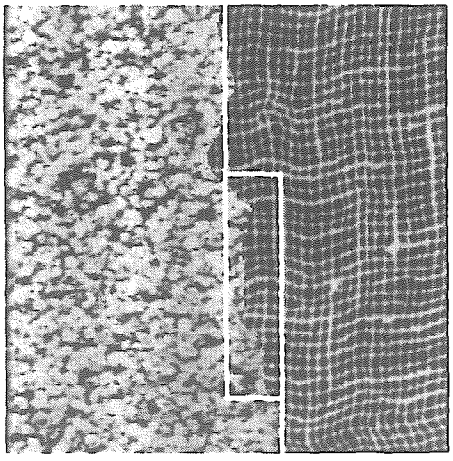


Figure 2. Segmentation results of a textured image at one scale.

Table 1. Performance of three different algorithms applied to the image in Figure 2. The number of sites is 64 (8×8).

Algorithms	using SOIL	L = 6	L = 10
Average iterations	167	2156	5080
Average total number of labels involved	31,423	827,904	3,251,200
Times more costly than using SOIL	1	26.3	103.5

reflects the declining tendency of the number of labels involved. As expected, the solid curve approaches 2 when the segmentation algorithm settles. Note the solid curve does not converge to 2 because the local configuration of the sites along texture boundaries consists of more than one label.

To compare the overall performance, some statistics are collected in Table 1 after the three algorithms are tested. Because of the nature of stochastic relaxation, a single run of the algorithms does not necessarily reveal the performance difference. Therefore, each algorithm repeats for 100 runs on the same input image in Figure 2 with the same values of C and interactive energy function defined in [9]. In Table 1, the item *Average iterations* tells that the third algorithm with L equal to 10 have to scan through the image for the most iterations in average, thus requires the longest time. The second algorithm places itself at the second rank of performance while the first algorithm using SOIL outperforms the other two. However, *Average iterations* does not actually tell us how much cost is saved because, for different algorithms, the sum of the number of labels involved in each iteration is different. This is in turn because, for different algorithms, the number of labels involved at each site is different. The item *Average total number of labels involved* in Table 1 tells the total number of labels involved throughout a single run of the algorithms in average. Note that involving more labels requires more time of calculation of Equation (6). Item *Times more costly than using SOIL* in Table 1 indicates how many times more costly of each methods than SOIL.

The above experiments are conducted at a single scale or the coarsest scale of a multiple-scale structure, where the initial configuration is set randomly. Within a multiple-scale structure, segmentation at finer scales is usually conditioned on the segmentation result of the coarser scales, i.e., the initial configuration is the coarse segmentation of the previous scale. For the conventional methods, the number of the segmented regions at the coarser scale can now be used as the value of L. For the proposed method using SOIL, the average |SOIL| is close to 2 from the first iteration at finer scale because the initial configuration is not random. So the advantages of using SOIL is still significant at finer scales.

5. Conclusions

In this work, two issues about the computational complexity of a *maximum a posteriori (MAP)* optimization problem using Markov random field (MRF) are addressed. Some previous approaches to reducing the computational complexity are reviewed and their advantages and disadvantages are also analyzed. A new novel idea, called SOIL, for reducing the computational complexity, which fully exploit the local characteristics of MRF's, is proposed. Application of the idea to image segmentation shows the merits and feasibility of it.

In summary, the main contributions of this method are:

- It requires neither prior information about the number of texture classes in the underlying image nor the estimation phase of class number.
- The searching for the global minimum in a virtually infinite label space (arbitrarily large number of labels) is confined in a small finite space without losing flexibility. The convergence rate is therefore significantly accelerated.
- It is simple and straightforward, thus, requires no complicated construction of the set of candidate configurations.
- It allows the starting temperature T to be decided independent of the size of label space L.

Acknowledgments

This work is supported by National Science Council, Taiwan, ROC, under grant no. NSC 88-2213-E-014-015

References

[1] P. Andrey and P. Tarroux, "Unsupervised Segmentation of Markov Random Field Modeled Textured Images Using Selectionist Relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 252-262, 1998.

[2] J. Besag, "Spatial Interaction and the statistical Analysis of Lattice Systems," *Journal of the Royal Statistical Society (Series B)*, vol. 36, pp. 192-236, 1974.

[3] C. A. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," *IEEE Transactions on Image Processing*, vol. 3, pp. 162-176, 1994.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.

[5] R. P. Grimaldi, *Discrete and Combinatorial Mathematics- an Applied Introduction*, Addison Wesley, 1994.

[6] R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications*, American Math. Society, 1980.

[7] James M. Kroll, Nam C. Phamdo, "Analysis and Design of Trellis Codes Optimized for a Binary Symmetric Markov Source with MAP Detection," *IEEE Trans. Info. Theory*, vol. IT-44, pp. 2977 - 2987, November 1998.

[8] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov Random Field Models for Texture Segmentation," *IEEE Transactions on Image Processing*, vol. 6, no.2, pp 251-267, 1997.

[9] C. T. Li, *Unsupervised Texture Segmentation Using Multiresolution Markov Random Fields*. PhD Thesis, Department of Computer Science, The University of Warwick, UK, 1998.

[10] S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, 1995.

[11] B. S. Manjunath and R. Chellappa, "Unsupervised Texture Segmentation Using Markov Random Field

Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 478-482, 1991.

[12] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 360-370, 1999.

[13] J. -P. Wang, "Stochastic Relaxation on Partitions with Connected Components and Its Application to Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 619-626, 1998.

[14] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, 1982.