Pattern Recognition, Vol. 25, No. 10, pp. 1217 1229, 1992
Printed in Great Britain

0031-3203/92 $5.00 + .00
Pergamon Press Ltd
Pattern Recognition Society

COLOR IMAGES' SEGMENTATION USING SCALE SPACE FILTER AND MARKOV RANDOM FIELD

CHUNG-LIN HUANG,† TAI-YUEN CHENG† and CHAUR-CHIN CHEN‡

† Department of Electrical Engineering, National Tsing-Hua University, Hsin-Chu, Taiwan 30043, Republic of China

‡ Department of Computer Science, National Tsing-Hua University, Hsin-Chu, Taiwan 30043, Republic of China

(*Received* 30 *July* 1991; *in revised form* 23 *January* 1992; *received for publication* 14 *February* 1992)

Abstract—A new hybrid method is presented that combines the scale space filter (SSF) and Markov random field (MRF) for color image segmentation. The fundamental idea of the SSF is to use the convolution of Gaussian functions and image-histogram to generate a scale space image and then find the proper interval bounded by the local extrema of the derivatives. The Gaussian function is with zero mean and varied standard deviation. Using the SSF the different scaled histogram is separated into intervals corresponding to peaks and valleys. The MRF makes use of the property that each pixel in an image has some relationship with other pixels. The basic construction of an MRF is a joint probability given the original data. The original data is the image that is obtained from the source and the result is called the label image. Because the MRF needs a number of segments before it converges to the global minimum, the SSF is exploited to do coarse segmentation (CS) and then MRF is used to do fine segmentation (FS) of the images. Basically, the former is histogram-based segmentation, whereas the latter is neighborhood-based segmentation. Finally, experimental results obtained from using SSF alone, MRF using iterated conditional mode (ICM), and MRF using Gibbs sampling are compared.

Color image segmentation   Markov random field (MRF)   Scale space filter (SSF)
Gibbs distribution   Iterated conditional modes (ICM)   Gibbs sampling (simulated annealing)

1. INTRODUCTION

Segmentation is a process of grouping image pixels into units that are homogeneous with respect to one or more characteristics. It is an important task in image analysis. However, many researchers have focused their attention on the monochrome image segmentation whose goal is the initial separation of the individual objects in the perception of the scene. A common problem in segmentation of a monochrome image occurs when an image has a background of varying gray level such as gradually changing shades, or when collecticns we would like to call regions or classes assume some broad range of gray scales. This problem is inherent since intensity is the only available information from monochrome images.

It has long been recognized that the human eye can detect only in the neighborhood of one or two dozen intensity levels at any one point in a complex image due to brightness adaptation, but can discern thousands of color shades and intensities. Color is a perceptual phenomenon related to human response to different wavelengths in the visible electromagnetic spectrum. Three psychological attributes, namely hue, saturation, and intensity, are generally used to represent color. Compared to a monochrome image, a color image provides, besides intensity, additional information in the image. In fact, human beings intuitively feel that color is an important part of their visual experience, and color is useful or even necessary for powerful processing in computer vision. Researchers have attempted to utilize this additional information. Thus applications with a color image are becoming increasingly prevalent nowadays.

From the image segmentation point of view, color image segmentation is basically a three-dimensional (3D) image histogram clustering technique[1,2] which is a computational expensive process. Instead of using a multidimensional histogram thresholding technique to segment the color image, our solution to this problem is to project the feature space onto its lower dimensional subspace (one-dimensional (1D) histogram). The clustering operation on the 1D histogram is basically a monochrome image segmentation, however, the clusters generated in each dimension have to be merged to form 3D clusters. The segmentation process will generate many more regions for the color images than for monochrome images because of the abundant information in the color images. However, using only the feature clustering may generate some random noise and very small regions which do not indicate the natural color regions. In this paper, we introduce a method to remove these pseudo regions.

Here, we use the scale space filter (SSF) to separate the histogram into some intervals that are bounded by the local extrema of its derivative and this operation is called the coarse segmentation (CS). Scale space filter can separate some pixels which are in the area closing the positions of peaks and those pixels which are close to the zones of valleys. Then, after the scale space filtering, Markov random field (MRF) operates on pixels and this operation is called the fine segmentation (FS). The coarse-to-fine step will find the segment number by itself. The CS is basically a histogram-based segmentation, it does not consider the spatial correlation among pixels. The FS is a neighborhood-based segmentation. Markov random field uses the a priori knowledge of neighboring pixels for segmentation.

The idea of filtering across a continuum of scale using Gaussian filters has been explored for the purpose of constructing symbolic descriptions of signals[3] and shapes.[4] Stansfield[5] motivated by using multiple Gaussian filters for edge detection[6] observed that the zero-crossing locations of 1D signals traced out contours in a two-dimensional (2D) space called scale space. Witkin[3] developed a multi-scale representation based on ternary trees and described a methodology for extracting the perceptually salient features of the signal. Babaud *et al.*[7] proved that the Gaussian is the only linear filter which has the desired property of not creating zero crossings as the scale decreases.

Multi-scale descriptions in terms of the location of zero crossings of the second derivatives of the signal in scale space have become known as fingerprints. By using fingerprints, events can be detected at coarse scales and localized by tracking zero-crossing contours in scale space down to fine scale. In this paper, because we are interested in extracting peaks and valleys in a 1D waveform of the histogram, zero crossings in the second derivative can be detected at a scale where only the more significant variations in the waveform remain. The detected zero crossings can then be tracked down to a lower scale where the precise location of the peak or valley can be determined.[8]

Then, we use MRF to segment the color image. Recently researchers have investigated the applications of MRF. Markov random field was originally used by Ising[9] who tried to model the structure of a crystal. Besag[10] proposed the formulation of conditional probability models for finite systems of spatially interacting random variables and applied the Hammersley–Clifford theorem to establish the relation between MRF and the Gibbs distribution. Geman and Geman[11] adopted a Bayesian approach and a restoration algorithm for computing the maximum a posterior (MAP) to estimate the original image given the degraded image. Subrahmonia *et al.*[12] used a similar method for 3D primitive model recognition, parameter estimation, and segmentation from a sequence of images taken by one or more calibrated cameras.

Markov random field models can generally be used for the reconstruction of a function starting from a set of noisy sparse data, such as intensity, stereo, or motion data. The essence of the MRF model is that the probability distribution of the configuration of the fields, given a set of data, is a Gibbs distribution. The model is then specified by an "energy function", that can be modeled to embody a priori information about the system. The basic idea of MRF is that each pixel must have some relations with its neighboring pixels. Therefore, we wish to establish the model that can extract "information" from neighboring pixels to get better results.

2. COARSE SEGMENTATION USING SCALE ŞPACE FILTER (SSF)

In this section we develop an SSF which can detect the positions of intervals containing only peaks or valleys at different scales for histogram analysis. The positions of peaks and valleys and their derivatives (and intervals bounded by extreme) usually indicate the existence of a smooth area and the edges. If we can exploit the information embedded in the histogram more effectively, we can analyze the image more accurately. The question is how to extract the meaningful peaks and valleys from it. Because the curves of most histograms are not smooth and they contain many rugged peaks, the computer cannot analyze the curve effectively. Witkin[13] proposed an approach that uses the Gaussian function to smooth the curve and then detects the area of peaks and valleys. We find this method is very effective for image segmentation.

This method uses different *scale* Gaussian functions to smooth the specific histogram and generates, a so-called, scale space image of which one coordinate is gray level and the other is the *scale*. When the *scale* value is large, the curve is smoother than smaller ones. When scale increases, the small peaks of the histogram will disappear and finally the curve becomes a horizontal line. In Fig. 1, we find the original peaks and valleys are smoother as scale value increases. What is the last scale to partition the histogram to obtain proper segments? The answer is that no one is proper for all the range of the histogram. In each smaller range of the histogram, there is a better scale value which can divide the histogram to better result. In other words, each interval bounded by a local extreme of deviation of the histogram might be generated according to different scale value. Let the histogram be interpreted by $f(x)$ where x is the value of the gray level and $F(x,\sigma)$ the scale space image function[13]

$$F(x,\sigma) = f(x) * g(x,\sigma) = \int_{-\infty}^{\infty} f(u)\frac{1}{\sigma\sqrt{(2\pi)}}\exp\left[-\frac{(x-u)^2}{2\sigma^2}\right]\mathrm{d}u \quad (1)$$

where "$*$" denotes convolution with respect to x. It is obvious that the scale space image is the result of convolution of the histogram and Gaussian function. The standard deviation of σ is the scale. F defines a surface in the (x,σ) plane, the surface swept out as the Gaussian's standard deviation is smoothly varied. We call the (x,σ) plane, *scale space*, and the surface F defined in equation (1), the *scale space image* of f. Figure 1 demonstrates the sequence of Gaussian smoothing at increasing σ ($=2, 5, 10$, and 22), which are constant-σ profiles from the scale space image. Figure 2 portrays the scale space image as a surface in perspective.

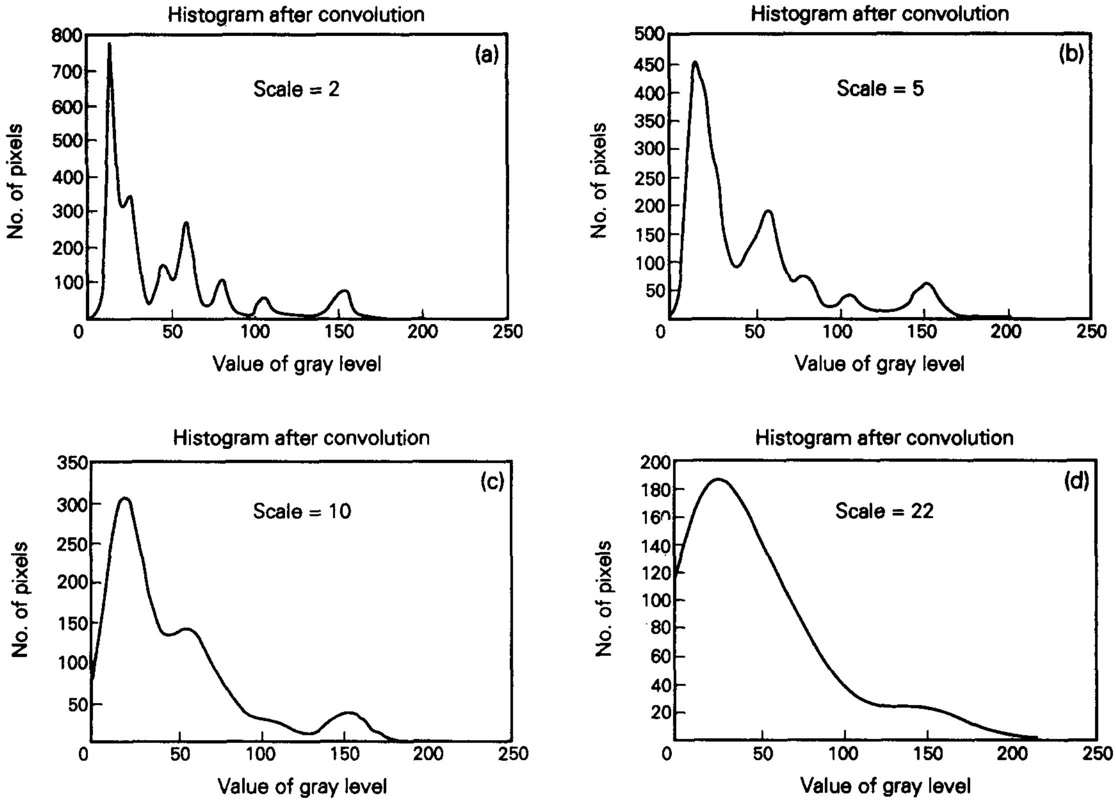


Fig. 1. Histograms after convolution with four different scales ($\sigma = 2, 5, 10, 22$).

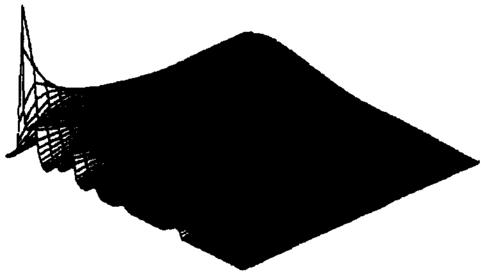


Fig. 2. Scale space image.

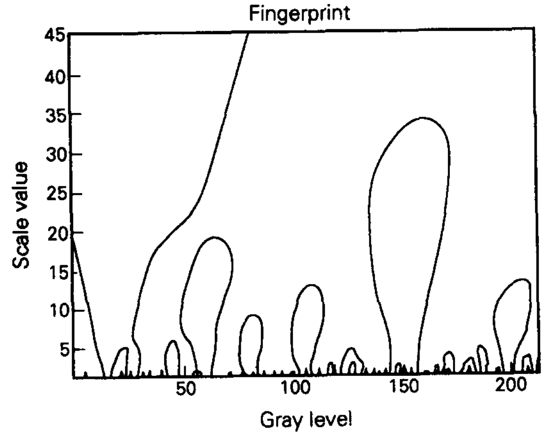


Fig. 3. Fingerprint.

Although, conceptually, we are interested in the extremes, working with the zero crossings is more convenient. Evaluating the partial at any fixed σ, the zero crossings in $\partial F/\partial x$ are local minima and maxima in the smoothed signal at that σ, and those in $\partial^2 F/\partial x^2$ are extremes of slope. The second deviation of the scale space image with respect to x is shown in Fig. 3. The horizontal axis is gray value and the vertical axis is the scale value. The curve lines are the positions of the zero crossing of the second deviation with respect to x. Figure 3 shows that the fixed-scale zeroes in fact lie on *zero-crossing contours* through scale space. The two arms of each arch form a complementary pair, crossing zero with opposite sense. As we sweep across the apex of an arch, with σ increasing, the pair approach each other with increasing velocity, then collide and are annihilated.

2.1. *Algorithm*

Firstly, we modify equation (1) to a discrete type. Secondly, we take the second deviation and find the positions of zero crossing. Thirdly, we find the positions of zero crossing, search the pairs of positions that have the same scale with a specific apex, and establish the connections between each pair. The position of zero crossing is called mark. Each mark has a record that contains the positions of itself and the apex. We use the position of the apex to distinguish the marks. After

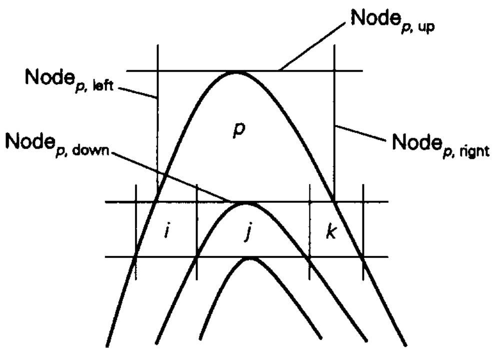


Fig. 4. Node generation.

we found all the marks and established the relation between zerò crossings, the initial image of second-order deviation can be thought of as the set of points and there are some wires connecting them. Each point has only one wire connected to it and the points connected with the same wire belong to the same apex, just like the fingerprint.

Given the fingerprint, the next step is to construct the initial node tree. Because node is rectangular, we use a four-tuple, ($Node_{\text{p,up}}$, $Node_{\text{p,down}}$, $Node_{\text{p,left}}$, $Node_{\text{p,right}}$), to represent the boundaries of node p, where $Node_{\text{p,up}}$ and $Node_{\text{p,down}}$ correspond to the scale value and $Node_{\text{p,left}}$ and $Node_{\text{p,right}}$ correspond to the gray level. The width of node p ($= Node_{\text{p,right}} - Node_{\text{p,left}}$) is the peak interval corresponding to the specific scale. The value of $Node_{\text{p,up}}$ is just the scale value of the apex. Each time the apex is detected, a new node is generated and its scale value is memorized for future use. The $Node_{\text{p,down}}$ is the scale value where the first child apex is found. So, the discovery of a new apex means that some new children nodes will be generated for the current parent node. The $Node_{\text{p,left}}$ and $Node_{\text{p,right}}$ are the crossing positions of the horizontal line touching the first child-apex and the two arms of the parent apex, shown in Fig. 4. In Fig. 4, $node_i$ and $node_k$ are also the children nodes of parent $node_p$ but not the apex node. From this figure we can understand that if the contour has a sub-contour, the parent at least has three children, one is an apex child and the other two are not.

After the nodes have been established, we select the active nodes to partition the histogram. We use the suggestion from Witkin[13] to choose the most stable nodes which have the longest distance between $Node_{\text{p,up}}$ and $Node_{\text{p,down}}$ than the mean distance of its children. However, to implement the algorithms for finding the active nodes from the second-order deviation of the scale space image, we face three unexpected problems: *out of range problem, overlapping-discontinuous problem,* and *node-selection problem.*

(1) *Out-of-range problem.* The pair of zeroes of the fingerprint never vanish when we move to a finer scale, however, they can be out of the range of the histogram gray scale. Therefore, the number of zeroes may decrease. There are two reasons for this phenomenon. Firstly, the scale space image is a continuous function resulting from the convolution operation of the histogram and Gaussian function. Because the Gaussian function expands over $(-\infty, \infty)$, the results also have an infinite expansion. Secondly, after the second-order deviation, the position of *zero crossing* may change with respect to different scales. Especially when the histogram has some sharp peaks, the zero crossings of the second deviation of *scale space image* will slide when the value of scale (σ) changes. In Fig. 1, we can easily find the phenomenon. Its left-hand side has a very sharp peak. After convolving with the Gaussian function, the histogram may have the width of the peaks extended, and the positions of zero crossing (which is also the reflection point) changed.

(2) *Overlapping-discontinuous problem.* Because the scale values are different in each interval and the positions of zero crossings may slide, intervals may be overlapped or disconnected, as in Fig. 5. The histogram cannot be segmented properly. We have to resize the length and relocate the position. We separate the overlapping interval to two segments and combine them to the two neighboring nodes. The segmentation is based on the different properties (children number) of two neighboring nodes which is mentioned in the following equation:

$$D = (Node_j \mathrm{l} - Node_i \mathrm{r}) * \frac{child\text{-}num_{nodei}}{child\text{-}num_{nodei} + child\text{-}num_{nodej}}$$

$$Node_i \mathrm{r} = D + Node_i \mathrm{r}, \quad Node_j \mathrm{l} = Node_i \mathrm{r}; \quad i < j \qquad (2)$$

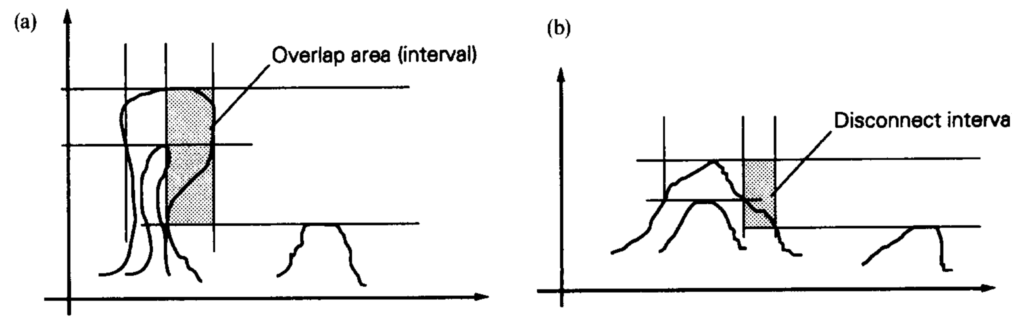


Fig. 5. (a) Overlapped interval. (b) Disconnected interval.

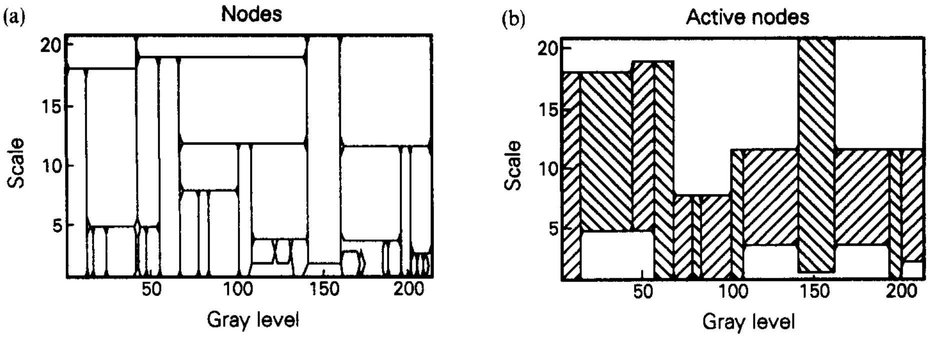


Fig. 6. (a) Interval tree of nodes. (b) Active nodes.

where $i < j$ means that node i is on the left-hand side of node j and $Node_i$ r means the right boundary of node i the same as $Node_j$ l. We use *node* instead of *interval* because we adjust the width of the node which is also the length of the interval. It is obvious that the interval length to be partitioned depends on the number of children nodes under parent nodes. We allow the node which has more children to get a larger part. After resizing, the intervals are all connected (see Fig. 6(a)).

(3) *Node-selection problem*. How to choose the *active nodes* (see Fig. 6(b)) is very important in the scale space method. According to Witkin's[13] suggestion, the active nodes are the nodes which can make the interval tree achieve the most stable state. We choose the node which has the longest distance between up-scale and down-scale of the node as the active node. In most cases, it generates good results.

3. FINE SEGMENTATION USING MARKOV RANDOM FIELD (MRF)

In this section, we present the basic definitions of MRF, and the formulation of a particular Gibbs distribution (GD) that is used in our image models. For the scene modeled by an MRF, a stochastic relaxation procedure for approaching the scene of maximum probability has been suggested by Geman and Geman.[11] Markov random fields possess several characteristics that make them useful in image segmentation. Properties such as smoothness and continuity of color regions over an entire image can be enforced using only dependencies among local neighbors. Discontinuities which separate regions of constant color may be computed, and then smooth regions can be found. In addition, the inclusion of both the prior and posterior distributions (through Bayes' rule) establishes a relationship between the noisy observed image and the color segmentation results.

Details of MRF theory can be found in references (11, 14). Briefly, an MRF is a lattice of sites; for example, an image of pixels. Since MRFs are stochastic processes, the pixels in an image may take on any of their allowed values, which mean that all images can be generated. In addition, the conditional probability of a particular pixel having a certain value is only a function of the neighboring pixels, not of the entire image. The Hammersley–Clifford theorem[10] establishes the equivalence between the conditional probabilities of the local characteristics in the MRF and local energy potentials in a GD. Therefore, the a priori probability that the MRF in a particular state can be calculated by summing the local energies over the entire image. We are interested in obtaining the MRF state that maximizes the a posteriori probability of the final segmentation given the observed data. From additional theorems,[11] the a priori energies can be added to an a posteriori energy term involving the difference between the actual observed data and the current MRF state (or predicted image).

We focus our attention on discrete 2D random fields defined over a finite $N_1 * N_2$ rectangular *lattice* of points (corresponding to pixels in a digital image) which is defined as

$$L = \{(i,j): 1 \le i \le N_1, 1 \le j \le N_2\}. \tag{3}$$

We start out by defining a neighborhood system on this rectangular lattice L.

Definition 1. A collection of subsets of L described as

$$\eta = \{\eta_{ij}: (i,j) \in L, \eta_{ij} \subset L\} \tag{4}$$

is a "neighborhood system" on L if and only if η_{ij}, the "neighborhood" of pixel (i,j) is such that: (1) $(i,j) \notin \eta_{ij}$, and (2) if $(k,l) \in \eta_{ij}$ then $(i,j) \in \eta_{kl}$ for any $(i,j) \in L$. We can now define an MRF with respect to the neighborhood system η defined over the lattice L.

Definition 2. Let η be a neighborhood system defined over lattice L. A random field $X = \{X_{ij}\}$ defined over lattice L is an MRF with respect to the neighborhood system η if and only if

$$\begin{aligned} &P[X_{ij} = x_{ij} | X_{kl} = x_{kl}, (k,l) \in L, (k,l) \ne (i,j)] \\ &\quad = p[X_{ij} = x_{ij} | X_{kl} = x_{kl}, (k,l) \in \eta_{ij}] \end{aligned} \tag{5}$$

for all $(i,j) \in L$, and $P(X = x) > 0$ for all x.

We note that capital letters denote random variables and random fields, and lower case letters denote specific realizations. A hierarchically ordered sequence

5	4	3	4	5
4	2	1	2	4
3	1	i,j	1	3
4	2	1	2	4
5	4	3	4	5

Fig. 7. Hierarchically arranged neighborhood systems η^m.

of neighborhood systems that are commonly used in image modeling is $\eta^1, \eta^2, \eta^3, \ldots$. $\eta^1 = \{\eta^1_{ij}\}$ is such that for each $(i,j) \in L$ (except for pixels in the boundaries) η^1_{ij} consists of the four pixels neighboring pixel (i,j). $\eta^2 = \{\eta^2_{ij}\}$ is such that η^2_{ij} consists of the eight pixels neighboring (i,j). The neighborhood structure for η^1 and η^2, as well as for η^3, η^4 and η^5 are shown in Fig. 7. The neighborhood system η^m is called the mth order neighborhood system.

Due to the finite lattice used, the neighborhoods of pixels on the boundaries are necessarily smaller unless a toroidal (periodic) lattice structure is assumed. It also should be pointed out that the neighborhood systems that can be defined over L are not limited to the hierarchically ordered sequence of neighborhood systems described above, nor do they have to be isotropic or homogeneous. Therefore, MRF is characterized by the conditional distributions called the "local characteristics" of the random field. There are inherent difficulties with this definition of MRF in terms of the local characteristics. Specifically, these difficulties are the consistency problems concerning the joint distribution and the unavailability of the joint distribution. These difficulties are alleviated by characterizing MRF through GD.

3.1. *Gibbs random field* (*GRF*)

To define a GRF, firstly, it is necessary to define the "cliques" associated with a lattice–neighborhood system pair (L, η).

Definition 3. A clique of the pair (L, η), denoted by c, is a subset of L such that: (1) c consists of a single pixel, or (2) for $(i,j) \neq (k,l), (i,j) \in c$ and $(i,j) \in c$ implies that $(i,j) \in \eta_{kl}$. The collection of all cliques of (L, η) is denoted by $C = C(L, \eta)$.

The types of all cliques associated with η^1 and η^2 are shown in Fig. 8. Now, GD can be defined as follows.

Definition 4. Let η be a neighborhood system defined over the finite lattice L. A random field $X = \{X_{ij}\}$ defined on L has Gibbs distribution (GD) or equivalently is a Gibbs random field (GRF) with respect to η if and only if its joint distribution is of the form

$$P(X = x) = \frac{1}{Z} e^{-U(x)} \qquad (6)$$

where

$$U(x) = \sum_{c \in C} V_c(x) \text{ is the energy function at } X = x, \qquad (7)$$
$$V_c(x) \text{ is the potential associated with clique } c$$

and

$$Z = \sum_x e^{-U(x)} \text{ is the partition function.} \qquad (8)$$

The partition function Z is simply a normalizing constant, so that the sum of the probability of all realizations x, add to unity. The only condition on the otherwise totally arbitrary "clique" potential $V_c(x)$ is that it depends only on the pixel values in clique c. Therefore, the GD energy function consists of two parts, one describing the interaction potential between neighbors, and the other associated with the difference between the predicted image and the actual observed data. Several methods of minimizing the energy function over the image (i.e. maximizing the probability) can be used, among them simulated annealing, deterministic procedures, and network solutions.

The origins of GD lie in the physics and statistical mechanics literature. Ising[9] used a special GD, now known as the "Ising model", to describe the magnetic properties of ferromagnetic materials. The source of the revived interest in GD is known as the Hammersley–Clifford theorem. This result proven in the 1970s independently by several researchers,[10,15] establishes a one to one correspondence between MRFs and GRFs. Unlike the MRF characterization, the GD characterization provides the joint distribution of the random field, is free from consistency problems, and provides a more workable spatial model. For the sake of completeness, we present a version of the Hammersley–Clifford theorem,[16] which establishes the equivalence of the MRF and GRF.

Theorem 1. Let η be a neighborhood system on a finite lattice L. A random field X is an MRF with respect to η if and only if its joint distribution is a GD with cliques associated with η.

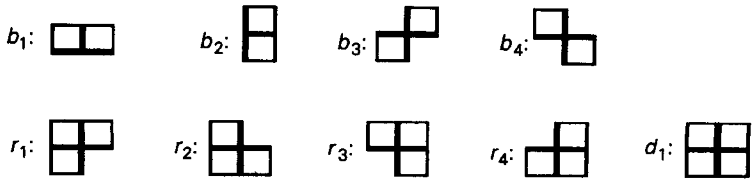


Fig. 8. Clique type.

It should be pointed out that any random field can be viewed as an MRF or a GRF with respect to a large enough neighborhood system, e.g. $\eta_{ij} = L$ for all $(i, j) \in L$. But our extensive experiments with these random fields indicate that MRF (or GRF) models even with the smallest neighborhood systems, e.g. η^1 and η^2, are very flexible and powerful. Furthermore, the positive condition in the definition of MRF does not pose a significant restriction. In the following, we present the general theory of MRFs on graphs, focusing on the aspects and examples that figure in image segmentation. Although the MRF initially is used in texture image, it also can be used in other images after some modification.

3.2. *Sampling discrete random fields*

Sampling is the process of generating a realization of a random field, given a model whose parameters have been specified or estimated. The intractability of the partition function Z in the denominator of the Gibbs density function implies that the standard statistical procedures for sampling random variables such as rejection–acceptance and conditional probability decomposition cannot be brought to bear on the problem of sampling a GRF or MRF. Instead, we must rely on relaxation-type algorithms.

The following algorithm[14] samples a GRF and eliminates the need for computing the partition function. It simulates a Markov chain whose states are G^M possible colorings of the image and is similar to simulated annealing.[10]

Algorithm 1 for sampling a GRF

```
(1) Initialize the $N \times N$ lattice by assigning a color randomly from $\mathbf{A} = \{0, 1, \ldots, G-1\}$ to each site. Call this initial coloring $\mathbf{x}$.
(2) For each site $s$ from 1 to $M = N^2$:
    (a) choose $g \in \mathbf{A}$ at random and let $y_s = g$. Let $y_t = x_t$ for all $t \neq s$;
    (b) let $p = \min\{1, P(\mathbf{X} = \mathbf{y})/p(\mathbf{X} = \mathbf{x})\}$;
    (c) replace $\mathbf{x}$ by $\mathbf{y}$ with probability $p$.
(3) Repeat (2) $N_{\text{iter}}$ times.
```

The ratio of likelihood in step 2(b) can be computed in practice because the ratio does not depend on the partition function Z, and only those cliques involving site s need to be included. Convergence of the algorithm is assured if N_{iter} is large enough ($N_{\text{iter}} \approx 50$).

A second sampling algorithm called the Gibbs sampler[11] is also a raster-scan algorithm.

Algorithm 2 for sampling a GRF

```
(1) Initialize the $N \times N$ lattice by assigning a color randomly from $\mathbf{A} = \{0, 1, \ldots, G-1\}$ to each site. Call this initial coloring $\mathbf{x}$.
(2) For each site $s$ from 1 to $M = N^2$:
    (a) compute probabilities $\{p_g\}$ for $g = 0, 1, \ldots, G-1$ where $p_g = \mathbf{P}(\mathbf{X}_s = g | \mathbf{X}_{\partial s} = \mathbf{x}_{\partial s})$ and $\mathbf{x}_{\partial s}$ is the current set of colors in the neighborhood of site $s$;
    (b) set the color of site $s$ to $g$ with probability $p_g$.
(3) Repeat (2) $N_{\text{iter}}$ times.
```

Algorithm 2 requires that G exponential functions be computed in step (2). Thus Algorithm 2 will take more time than Algorithm 1 and is subject to underflow. Both algorithms use a randomly chosen initial state, although theory says that the algorithm will converge for any initial state.

3.3. *Pixel labeling*

Decision making problems encountered in image segmentation require that labels be assigned to pixels in an image based on a degraded version of the true image. In this section, we mention the labeling problem with emphasis on the role of random field models. In the next section, we will mention the relaxation algorithms for segmentation that use MRF models.

An image labeling problem is specified in terms of a set of objects and a set of object labels $L = \{l_1, l_2, \ldots, l_G\}$. The objects can be individual pixels, edge elements, or segments. For simplicity, we take the objects to be the M pixels. In image segmentation, the labels denote the pattern classes in the image. For instance, if the image contains an object placed in the background, $G = 2$ labels could be used, one for object and the other for the background. The object itself might contain several gray values. The true pixel labeling is denoted by $\mathbf{x}^* = \{x_1^*, x_2^*, x_M^*\}$. The objective of all segmentation algorithms discussed here is to estimate $\mathbf{x}^*$.

The set of observable random variables is denoted by $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_M\}$, where Y_t is the feature vector associated with the tth pixel. Contextual information enters the labeling problem through an MRF model of the statistical dependent among the neighboring pixels. The true labeling $\mathbf{x}^*$ is viewed as realization of an MRF imposed on $\mathbf{X} = \{X_1, X_2, \ldots, X_M\}$ where each random variable X_t takes on values on L. This MRF serves a prior distribution for the labeling being estimated.

Given a set of observed feature vectors, $\mathbf{Y} = \mathbf{y}$, and the contextual information as an MRF, $P(\mathbf{X} = \mathbf{x})$, the problem is to find the *optimal* estimate of the true labeling $\mathbf{x}^*$. Traditionally, each pixel was labeled based on y alone and then the assigned label as updated iteratively by the discrete relaxation process. The current trend is to combine these two steps using the Bayesian formulation. The MAP method chooses the estimates $\hat{\mathbf{x}}$ that maximizes the posterior probability of $\mathbf{X} = \hat{\mathbf{x}}$, given $\mathbf{Y} = \mathbf{y}$.

The model relating observation $\mathbf{y}$ to labeling $\mathbf{x}$ is chosen to ensure that the posterior distribution of $\mathbf{X}$, given $\mathbf{Y} = \mathbf{y}$ is also an MRF. The sampling algorithms for MRFs can then be exploited in the estimation process. Requiring conditional independence of the observed random variables, given the true labels, is sufficient to ensure that the posterior distribution is an MRF. Although the marginal conditional distribution of Y_i given the true label $X_i = x_i$ can have any form, this distribution is often assumed to be normal with mean μ_{xi} and variance σ_{xi}. The observed random variable is, thus, continuous rather than discrete for the purposes of this analysis. For instance, if the prior distribution is taken to be a Derin–Elliott MRF,[16] the posterior distribution of **X** given **Y** = **y** is

$$P(\mathbf{X}=\mathbf{x}\,|\,\mathbf{Y}=\mathbf{y}) = \mathrm{e}^{-U(x|y)}/Z_{x|y} \tag{9}$$

where $U(x|y)$ is the energy function and $Z_{x|y}$ the normalizing constant.

3.4. *Relaxation algorithms*

The posterior distribution $P(\mathbf{X}=\mathbf{x}\,|\,\mathbf{Y}=\mathbf{y})$ is a powerful tool for image analysis; in principle we can construct the optimal (Bayesian) estimator for the original image or examine the image sampled from $P(\mathbf{X}=\mathbf{x}\,|\,\mathbf{Y}=\mathbf{y})$. Our job here is to find the values of **x** which maximize the posterior distribution for a fixed **y**, i.e. minimizing the energy function. Since the size of **X** is at least 2^{4000}, corresponding to a small (64 × 64) binary image, the identification of even a near-optimal solution is extremely difficult. Therefore, there are many types of "relaxation" methods invented. Here, we use two MAP procedures: Gibbs sampling (or simulated annealing)[10,17] and ICM[17,18] and compare the results.

(1) *Gibbs sampling (simulated annealing).* Gibbs sampling (or simulated annealing) is a method of function optimization that tries to avoid the pitfalls inherent in other methods for optimizing functions of many variables. It makes no assumptions about the smoothness of functions to be optimized, but imposes severe computational requirements. Simulated annealing is in the class of stochastic relaxation algorithms and is based on the classical Metropolis *et al.* method. The parameter T called "temperature" in equations (19) and (20) is used to supervise the annealing process. The probability density function of the states in each site may be smoother when the parameter T is large; otherwise, the probability density function is sharp which can enhance the result when we use the Gibbs sampling.

Algorithm for MAP estimation by simulated annealing

```
(1) Choose an initial temperature $T$.
(2) Initialize $\hat{\mathbf{x}}$ by choosing $x_t$ as the color $\hat{x}_t$ that maximizes $P(Y_t = y_t | X_t = x_t)$ for each pixel $t$.
(3) Perturb $\hat{\mathbf{x}}$ into $\hat{\mathbf{z}}$. Let $\Delta = U(\hat{z}|y) - U(\hat{x}|y)$
    if $\Delta > 0$ then replace $\hat{\mathbf{x}}$ with $\hat{\mathbf{z}}$
    else replace $\hat{\mathbf{x}}$ by $\hat{\mathbf{z}}$ with probability $\mathrm{e}^{\Delta/T}$.
(4) Repeat (3) $N_{\text{inner}}$ times.
(5) Replace $T$ by $f(T)$ where $f$ is a monotonically decreasing function.
(6) Repeat (3)–(5) until frozen.
```

To achieve the mono-decreasing function of $f(T)$, we let $f(T) = T(k)$ and denote $T(k)$ as

$$T(k) = \frac{C}{\log(1+k)}, \quad 1 \le k \le K \tag{10}$$

where $T(k)$ is the temperature, K the total iteration number, and C a constant which is arbitrary.

(2) *Iterated conditional modes (ICMs).* Besag[18] recognizes the inherent difficulty in computing the MAP estimate and proposes the ICM method as a computational feasible alternative. The ICM method is an attempt to estimate pixel labels in a computational simple manner while retaining the MRF as a model of prior information and to avoid the tendency of an MRF to degenerate into a single color image (phase transition).

The key to the ICM method is the following equation of proportionality for the probability of the label at pixel t, given the observed image y and the current estimates of the labels of all pixels in the neighborhood of pixel t

$$P(X_t = x_t | Y = y,\ X_{S|t} = x_{S|t}) \propto P(Y_t = y_t | X_t = x_t) P(X_t = x_t | X_{\partial t} = x_{\partial t}). \tag{11}$$

The ICM algorithm is described below. The M-vector **y** is given and an M-vector of estimate pixel $\hat{x}$ is computed.

Algorithm for estimating pixel labels by ICM method

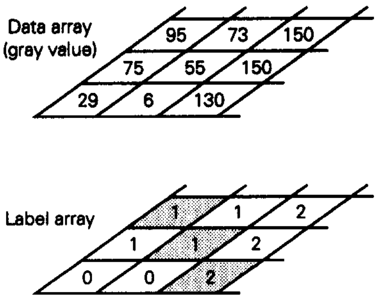
```
(1) Choose an MRF model for the true labels $\mathbf{X}$.
(2) Initialize $\hat{\mathbf{x}}$ by choosing $x_t$ as the color $\hat{x}_t$ that maximizes $P(Y_t = y_t | X_t = x_t)$ for each pixel $t$.
(3) For $t$ from 1 to $M$, update $\hat{x}_t$ by the value of $x_t$ that maximizes equation (11).
(4) Repeat (3) $N_{\text{iter}}$ times.
```

Experience has shown that about 10 raster scans of an image are sufficient for convergence. The computation is a few orders of magnitude faster than the simulated annealing approach to finding the MAP estimators of the labels.

4. HYBRID SYSTEM AND EXPERIMENTAL RESULTS

In this section, we will illustrate the whole system and some results will be shown. The color images are separated to three components RGB. Our hybrid system operates on each component image independently. The three component images are segmented individually and then combined as a segmented color image. Scale space filter can segment the initial image and get the segment number which is used by MRF. An image can be categorized into three types, i.e. texture, smooth area, and boundary. Images have most of the pixels belonging to the smooth area and few pixels related to edge and texture. The pixels in the smooth area with similar gray value are segmented using SSF, whereas, the rest of the pixels are to be classified using MRF.

Our method is divided into two parts. The first part, the SSF, coarsely partitions the images which is called coarse segmentation (CS). Scale space filter can partition the histogram into several intervals which is bounded by zero crossing of the second deviation of the scale space image. The following MRF, does fine



Label	Gray value	Interval
0	31	[26, 36]
1	79	[69, 89]
2	163	[148, 178]
–	–	–
–	–	–
–	–	–

Fig. 9. Relation between image array and label array. The masked block in the label array is temporarily assigned a label value.

segmentation (FS) to smooth the coarsely-segmented images. In order to compare the difference between using ICM and Gibbs sampling, we have used both approaches for image segmentation.

The pixels located in the interval containing the peaks of the histogram are labeled during the process of SSF and the other pixels in the valley-interval are ambiguous. Before FS we build a label array. The reason is that the change of state is on the label array. The ambiguous pixels are temporarily assigned the nearest label value (Fig. 9). The masked blocks (Fig. 9) are the temporary label numbers which may be changed in the process of MRF. The representative of peak-interval is at the middle position of the interval.

The MRF model processes a label array with the same size as the original image. Originally the label is unrelated with the true color. The color, after segmentation, needs to be similar to the initial image color. We make two color maps to keep the major colors of the segmented image as similar as possible to the original image color. Firstly, we built the map which contains the label number corresponding to the gray level such that any gray value can find its corresponding label number through the array (illustrated in Fig. 10). Those pixels in the valley-interval are temporarily assigned the nearest label in the gray level. Secondly, another map is needed to do the inverse mapping for the segmented label image. Since the first map transfers an interval of the label to a single label, the second map needs to select a representative from each interval of levels for each label and does the reverse operation. In Fig. 10, we find the representative of a peak interval is not necessarily the same as the peak value because the representative is selected from the middle position of the interval. Before FS, it uses the first map to transfer the gray value to label number, and after FS, it uses the second map to replace the labels with gray-level values.

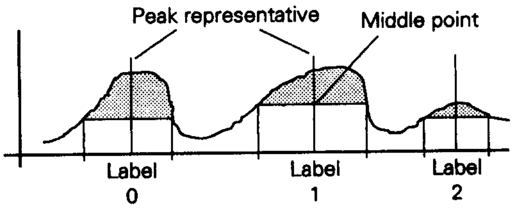


Fig. 10. The mapping from gray value to label number. The gray value under the masked area is mapped to the number under the horizontal axis.

In the segmentation process, we map the clustering pixels to a single color, and assume no noise exists. Here the only operation is to measure the distance between the original color and the label color. We cannot use the gray value as the distance because the differences among label colors are different for different images. The label colors are determined at intermediate positions of the intervals obtained from SSF (see Fig. 10). Each label color is just the delegate of some similar pixels. We measure the distance according to the distance between the ambiguous pixel and the label. If the gray value of a pixel is located in the middle between two labels, the distances between the pixel and the two labels are both one. The distances of the next two outer labels are two, and so on (see Fig. 11). According to the distance measurement described above, we define the posterior probability as

$$P(X_s = x_s | X_r = x_r,\ r \in \eta_s) = \frac{\exp\left[\alpha \sum_{i=1}^{8} d_N(x_s, r_i) + \beta d_D(x_s, g_s)\right]}{\sum_{x_s \in \Lambda} \exp\left[\alpha \sum_{i=1}^{8} d_N(x_s, r_i) + \beta d_D(x_s, g_s)\right]} \tag{12}$$

where η_s is the neighbor of site s, d_N and d_D the distance measures between neighbors and given image data. We use first order, four neighbors, so that the summation

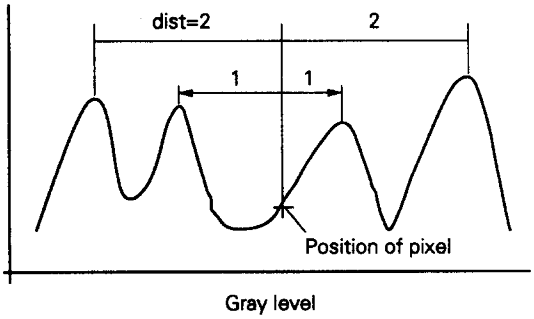


Fig. 11. Distance of label.

of d_N is from 1 to 4. Λ is the set of all possible states S of the label. α and β are constants which control the weight of neighbors and given data. Controlling α, we can emphasize the importance of neighbor-relation or given data (i.e. when the initial image has a lot of noise, the given data is no good). We may increase weight α to use the relation of neighbors to improve the result. If the original image is good, we can decrease the value of α to get better results. However, if α is too small, the term d_N will make less influence than the other term d_D, so that the image will change very slowly. In our experiments, α is fixed to 0.3, $\beta = 1$ and the iteration number (ICM) is 10.

Figure 12(a) shows the initial image and Fig. 12(b) illustrates the outlines of the segmented regions after SSF which has some very small regions. Figures 12(c) and (d) are the outlines of the segmented regions after MRF using ICM (10 iterations) and Gibbs sampling (400 iterations), respectively. Comparing Figs 12(c) and (d), we can find that both results are similar except Fig. 12(c) has smaller regions. It occurs because of the random choice of states. Figures 13 and 14 are other examples which also show similar results.

From the experimental results we find the SSF is very effective. The benefit of the SSF is that it can determine the number of labels. The SSF dominates the segmentation process. However, if some peaks or valleys of the histogram are misclassified by SSF, the final result might be worse. The following MRF compensates the misclassified pixels. The major disadvantage of the MRF using simulated annealing is that the convergent rate is too slow (about 4 h in our Solbourne 24 MIPS workstation). The ICM algorithm makes very good improvement, it takes less than 10 min in our Solbourne workstation, and the results are not very different from those using simulated annealing. Translating the RGB to another axis with linear translation or modifying the rule of selecting active nodes may avoid the problems.

5. CONCLUSION

In this paper we present a new hybrid method for color image segmentation. Although the algorithm based on MRF does not restrict the state number in each site, the results show that the state number should

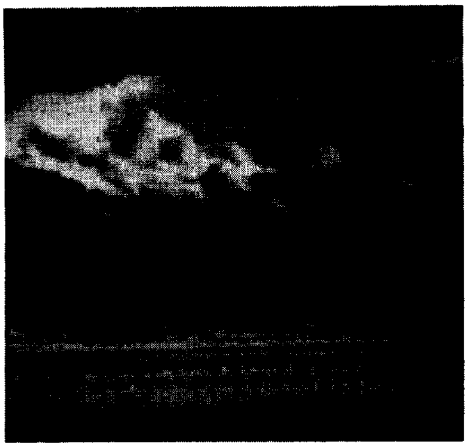


Fig. 12(a). Original image.

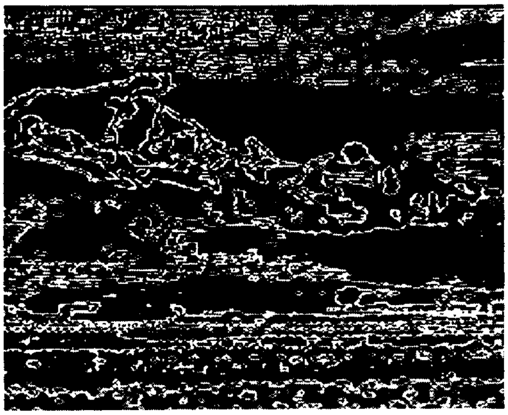


Fig. 12(b). Segmented image after SSF.

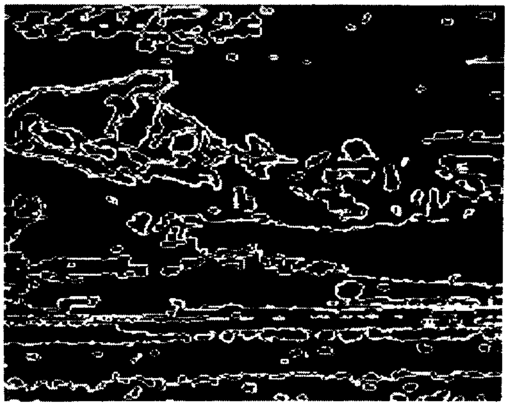


Fig. 12(c). Segmented image using ICM.

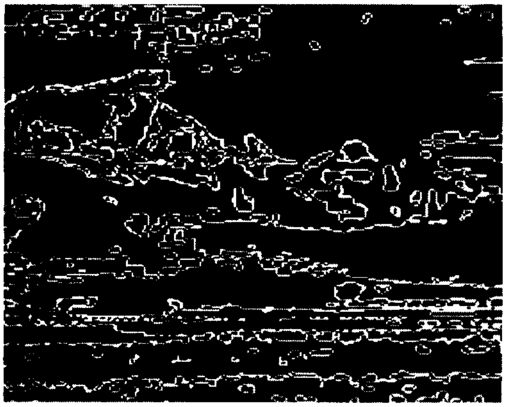


Fig. 12(d). Segmented image using Gibbs sampling after 400 iterations.

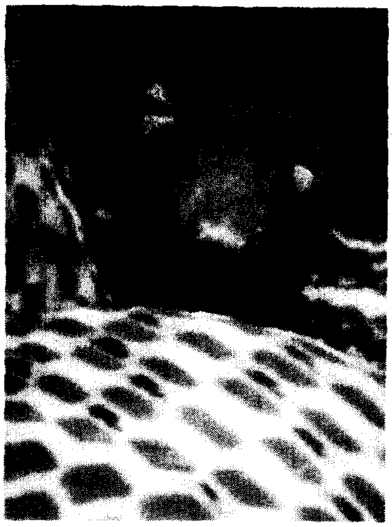


Fig. 13(a). Original image.

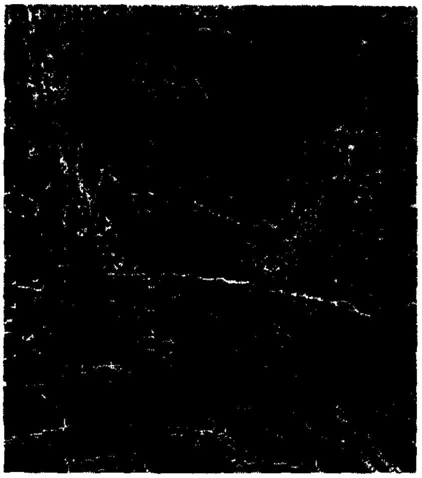


Fig. 13(b). Segmented image after SSF.

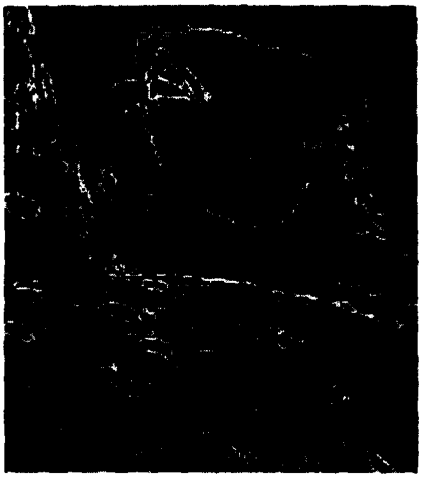


Fig. 13(c). Segmented image using ICM.

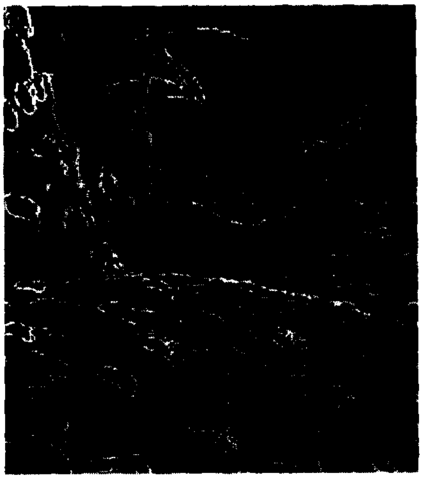


Fig. 13(d). Segmented image using Gibbs sampling after 500 iterations.

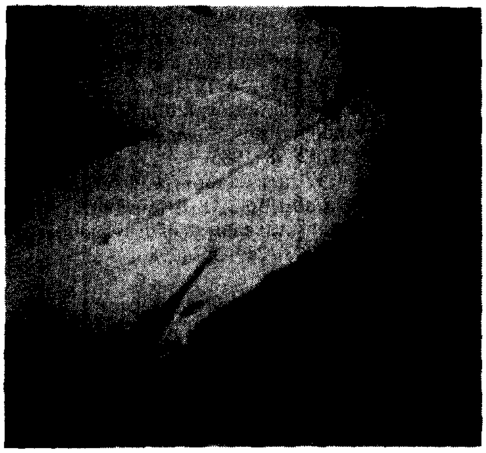


Fig. 14(a). Original image.

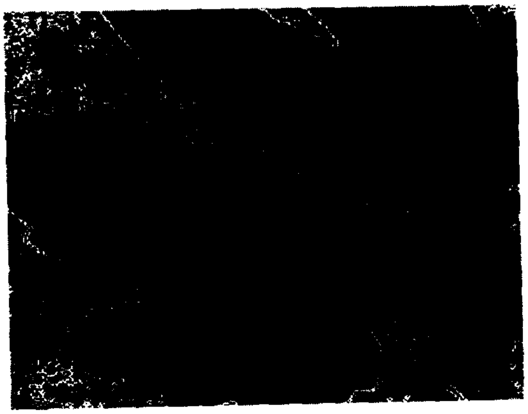


Fig. 14(b). Segmented image after SSF.

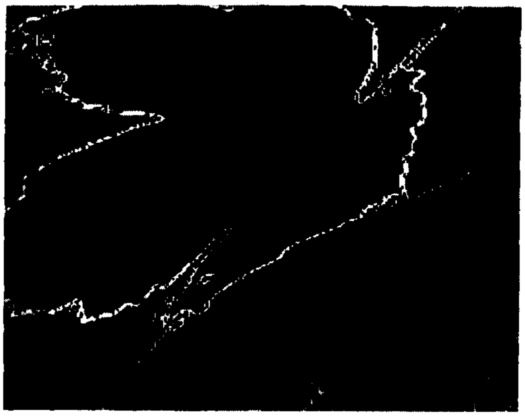


Fig. 14(c). Segmented image using ICM.

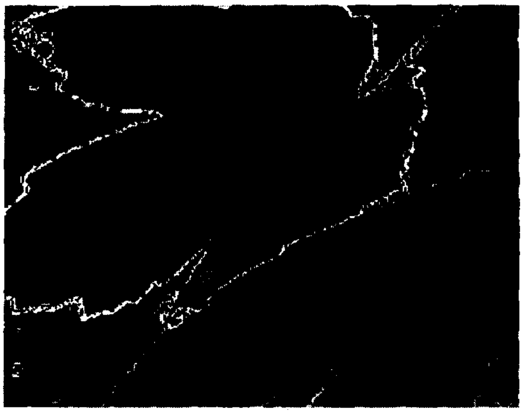


Fig. 14(d). Segmented image using Gibbs sampling after 500 iterations.

be limited. If there are too many states for each site, the convergence rate will become very slow, and the segmented image is unstable (trapped in the local minima). We first translate the real image into the label image for further segmentation. The label image is obtained by segmenting images by human-like histogram-based segmentation called SSF.

Scale space filter uses a Gaussian function to smooth the histogram and uses the zero crossings of the second deviation associated with x to define the intervals containing peaks or valleys. Finally it chooses the most stability nodes to set the segmental intervals. The final step, i.e. choosing most stable intervals, is very important and its rule is flexible. Although Witkin declares that the longest node is the most stable node, we found that the final result does not always correspond to the best one. It may lose some peaks or valleys or capture too much detail. We are working on developing a general method that can get optimum segmentation of the histogram to different scale intervals.

Here, we have implemented algorithms for color image segmentation which proves to be a very successful process. The color images provide, besides intensity, additional information in the image. The color image segmentation is a more challenging task than its monochrome counterpart. Its applications are becoming increasingly significant nowadays.

REFERENCES

1. Y. Lim and S. Lee, On the color image segmentation algorithm based on thresholding and the fuzzy c-means techniques, *Pattern Recognition* **23**, 935–952 (1990).
2. M. Celenk, A color clustering technique for image segmentation, *Comput. Vision Graphics Image Process.* **52**, 145–170 (1990).
3. A. P. Witkin, Scale-space filtering, *Proc. IJCAI-83*, pp. 1019–1022, August (1983).
4. H. Asada and M. Brandy, The curvature primal sketch, M.I.T., Cambridge, AI Lab Memo 758 (1984).
5. J. Stansfield, Conclusion from the commodity expert project, M.I.T., Cambridge, AI Lab Memo 601 (1980).
6. D. Marr and E. Hildreth, A theory of edge detection, *Proc. R. Soc. London* **207**, 187–217 (1980).
7. J. Babaud, A. Witkin and R. Duda, Uniqueness of the Gaussian kernel for scale-space filtering, Fairchild TR 645 (1983).
8. M. J. Carlotto, Histogram analysis using a scale-space approach, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-9**(1), 121–129 (January 1987).
9. E. Ising, *Z. Phys.* **31**, 253 (1925).
10. J. Besag, Spatial interaction and the statistical analysis of lattice systems (with discussion), *J. R. Statist. Soc. Ser. B* **36**, 192–236 (1974).
11. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-6**, 721–741 (November 1984).
12. J. Subrahmonia, Y. P. Hung and D. B. Cooper, Model-based segmentation and estimation of 3D surfaces from two or more intensity images using Markov random fields, *Proc. 10th ICPR*, Atlantic City, 16–21 June (1990).
13. A. P. Witkin, Scale-space filtering: a new approach to multi-scale description, *Image Understanding*, S. Ullman and W. Richards, eds, pp. 79–95. Ablex, New Jersey (1984).
14. C. C. Chen, Markov random field in image analysis, Doctor's thesis, Dept. Comput. Sci., Univ. Michigan State (1988).
15. F. Spitzer, Markov random fields and Gibbs ensembles, *Am. Math. Mon.* **78**, 142–154 (1971).
16. H. Derin and W. S. Cole, Segmentation of texture images using Gibbs random fields, *Comput. Vision Graphics Image Process.* **35**, 72–98 (1986).
17. R. C. Dube and A. K. Jain, Random field models in image analysis, *J. Appl. Statist.* **16**(2), 131–164 (1989).
18. J. Besag, On the statistical analysis of dirty pictures, *J. R. Statist. Soc. Ser. B* **48**(3), 259–302 (1986).

About the Author—CHUNG-LIN HUANG was born in Taichung, Taiwan, in 1955. He received his B.S. degree in nuclear engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1977, and M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1979, respectively. He obtained his Ph.D. degree in electrical engineering from University of Florida, Gainesville, Florida, U.S.A., in 1987. From 1981 to 1983, he was an associate engineer in ERSO, ITRI, Hsin-Chu, Taiwan. From 1987 to 1988, he worked for the Unisys Co., Orange County, California, U.S.A., as a project engineer. Since August

1988 he has been with the Electrical Engineering Department, National Tsing-Hua University, as an associate professor. His research interests are in the areas of image processing, computer vision, and visual communication. Dr Huang is a member of IEEE and SPIE.

About the Author—TAI-YUEN CHENG was born in Chung-Li, Taiwan, in 1964. He received his B.S. degree from Electrical Engineering Department, National Taiwan Institute of Technology, in 1989 and his M.S. from Electrical Engineering Department, National Tsing-Hua University, in June 1991. His research interests are computer vision and image processing.

About the Author—CHAUR-CHIN CHEN was born in Kaohsiung, Taiwan, in 1955. He received a B.S. degree in mathematics from National Taiwan University, Taipei, Taiwan, in 1977, M.S. degrees in mathematics and computer science in 1982 and 1984, and Ph.D. degree in computer science in 1988 from Michigan State University. He is currently an associate professor in the Department of Computer Science at National Tsing-Hua University, Taiwan, Republic of China. His research interests are in the areas of image pattern recognition, computer vision, and image models. Dr Chen is a member of the Pattern Recognition Society, the Association for Computing Machinery, and IEEE Computer Society.