Multiresolution Image Classification by Hierarchical Modeling with Two-Dimensional Hidden Markov Models

Jia Li, *Member, IEEE,* Robert M. Gray, *Fellow, IEEE,* and Richard A. Olshen, *Senior Member, IEEE*

***Abstract*—This paper treats a multiresolution hidden Markov model for classifying images. Each image is represented by feature vectors at several resolutions, which are statistically dependent as modeled by the underlying state process, a multiscale Markov mesh. Unknowns in the model are estimated by maximum likelihood, in particular by employing the expectation-maximization algorithm. An image is classified by finding the optimal set of states with maximum *a posteriori* probability. States are then mapped into classes. The multiresolution model enables multiscale information about context to be incorporated into classification. Suboptimal algorithms based on the model provide progressive classification that is much faster than the algorithm based on single-resolution hidden Markov models.**

***Index Terms*—EM algorithm, image classification, image segmentation, multiresolution hidden Markov model, tests of goodness of fit.**

I. Introduction

RECENT years have seen substantial interest and activity devoted to algorithms for multiresolution processing [16], [37]. One reason for this focus on image segmentation is that multiresolution processing seems to imitate the decision procedure of the human visual system (HVS) [28]. For example, when the HVS segments a picture shown in Fig. 1 into a foreground region (a fox) and a background region, the foreground can be located roughly by a brief glance, which is similar to viewing a low-resolution image. As is shown in Fig. 1(b), the crude decision leaves only a small unsure area around the boundary. Further careful examination of details at the boundary results in the final decision as to what is important in the image. Both global and local information are used by the HVS, which distributes effort unevenly by looking at more ambiguous regions at higher resolutions than it devotes to other regions.

Context-dependent classification algorithms based on two-dimensional hidden Markov models (2-D HMMs) have been developed [14], [24], [25] to overcome the overlocalization of conventional block-based classification algorithms. In this paper, a multiresolution extension of the 2-D HMMs described in [25] is proposed so that more global context information can be used efficiently. A joint decision on classes for the entire image is needed to classify optimally an image based on the 2-D HMM [25]. In real life, however, because of computational complexity, we have to divide an image into subimages and ignore statistical dependence among the subimages. With the increase of model complexity, it is necessary to decrease the size of the subimages to preserve modest computational feasibility. Instead of using smaller subimages, a classifier based on the multiresolution model retains tractability by representing context information hierarchically.

With a 2-D multiresolution hidden Markov model (MHMM), an image is taken to be a collection of feature vectors at several resolutions. These feature vectors at a particular resolution are determined only by the image at that resolution. The feature vectors across all the resolutions are generated by a multiresolution Markov source [35], [18]. As with the 2-D HMM, the source exists in a state at any block at any resolution. Given the state of a block at each particular resolution, the feature vector is assumed to have a Gaussian distribution so that the unconditional distribution is a Gaussian mixture. The parameters of each Gaussian distribution depend on both state and resolution. At any fixed resolution, as with the 2-D HMM, the probability of the source entering a particular state is governed by a second-order Markov mesh [1]. Unlike the HMM, there are multiple Markov meshes at one resolution whose transition probabilities depend on the states of parent blocks.

Many other multiresolution models have been developed to represent statistical dependence among image pixels, with wide applications in image segmentation, denoising, restoration, etc. The multiscale autoregressive model proposed by Basseville *et al.* [3], the multiscale random field (MSRF) proposed by Bouman and Shapiro [7], and the wavelet-domain HMM proposed by Crouse *et al.* [11] is discussed and compared with the 2-D MHMM in Section III after necessary notation is introduced in Section II.

As was mentioned, the human visual system is fast as well as accurate, at least by standards of automated technologies, whereas the 2-D MHMM and other multiresolution models [3] do not necessarily benefit classification speed because information is combined from several resolutions in order to make a decision regarding a local region. However, a 2-D MHMM provides a hierarchical structure for fast progressive classification if the maximum *a posteriori* (MAP) classification rule is

Manuscript received September 1, 1999; revised March 16, 2000. This work was supported by the National Science Foundation under NSF Grant MIP-9706284.

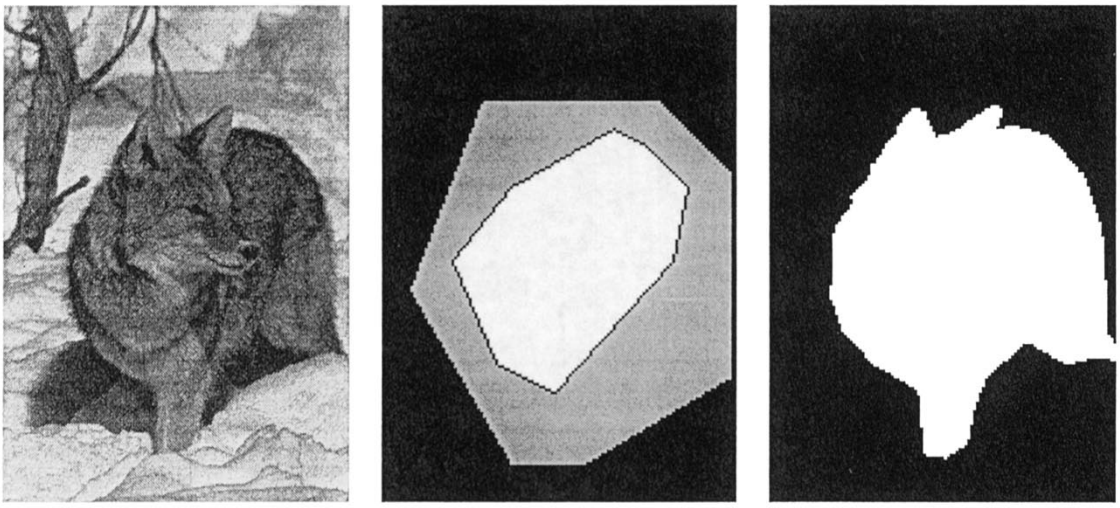
J. Li is with the Xerox Palo Alto Research Center, Palo Alto, CA 94304 USA (e-mail: jiali@isl.stanford.edu).

R. M. Gray is with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: rmgray@stanford.edu).

R. A. Olshen is with the Department of Health Research and Policy, Stanford University, Stanford, CA 94305 USA (e-mail: olshen@stat.stanford.edu).

Communicated by P. Moulin, Guest Editor.

Publisher Item Identifier S 0018-9448(00)05454-7.



(a) (b) (c)

Fig. 1. The segmentation process of the human visual system. (a) Original image. (b) A rough segmentation with the gray region being undecided. (c) The refined segmentation.

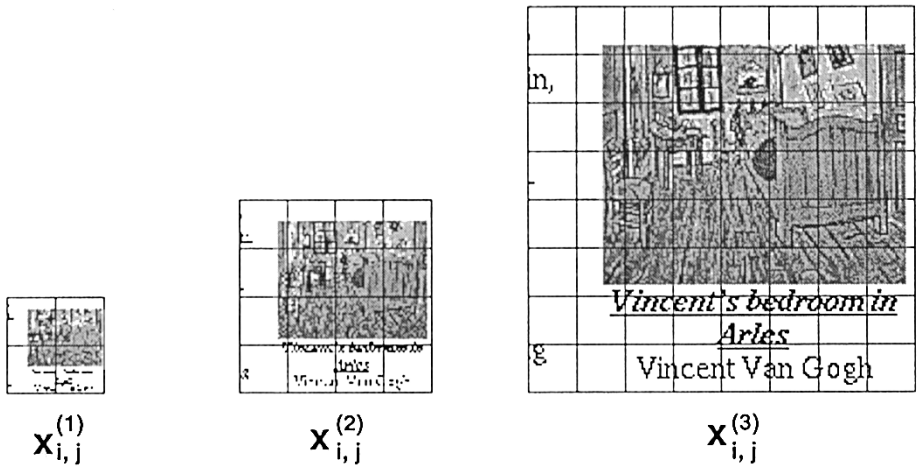


Fig. 2. Multiple resolutions of an image.

relaxed. The progressive classifier is inspired by the human visual system to examine higher resolutions selectively for more ambiguous regions.

In Section II, a mathematical formulation of the basic assumptions of a 2-D multiresolution HMM is provided. Related work on multiresolution modeling for images is discussed in Section III. The algorithm is presented in Section IV. Fast algorithms for progressive classification are presented in Section V. Section VI provides an analysis of computational complexity. Experiments with the algorithm are described in Section VII. Section VIII is about hypothesis testing as it applies to determining the validity of the MHMM. Conclusions are drawn in Section IX.

II. Basic Assumptions of 2-D MHMM

To classify an image, representations of the image at different resolutions are computed first. The original image corresponds to the highest resolution. Lower resolutions are generated by successively filtering out high-frequency information. Wavelet

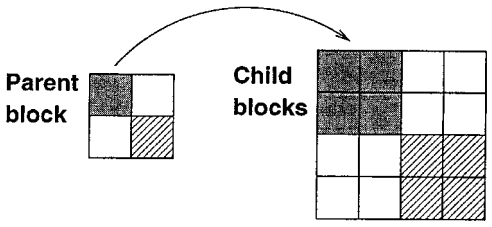


Fig. 3. The image hierarchy across resolutions.

transforms [12] naturally provide low-resolution images in the low-frequency band (the LL band). A sequence of images at several resolutions is shown in Fig. 2. As subsampling is applied for every reduced resolution, the image size decreases by a factor of two in both directions. As is shown by Fig. 2, the number of blocks in both rows and columns is successively diminished by half at each lower resolution. Obviously, a block at a lower resolution covers a spatially more global region of the image. As is indicated by Fig. 3, the block at the lower resolution is referred to as a parent block, and the four blocks at the same

spatial location at the higher resolution are referred to as child blocks. We will always assume such a "quadtree" split in the sequel since the training and testing algorithms can be extended easily to other hierarchical structures.

We first review the basic assumptions of the single-resolution 2-D HMM as presented in [25]. In the 2-D HMM, feature vectors are generated by a Markov model that may change state once every block. Suppose there are M states, the state of block (i, j) being denoted by $s_{i,j}$. The feature vector of block (i, j) is $u_{i,j}$, and the class is $c_{i,j}$. We use $P(\cdot)$ to represent the probability of an event. We denote $(i', j') < (i, j)$ if $i' < i$ or $i' = i$, $j' < j$, in which case we say that block (i', j') is before block (i, j). The first assumption is that

$$P(s_{i,j}|\ \text{context}) = a_{m,n,l},$$
$$\text{context} = \{s_{i',j'},\ u_{i',j'}\colon (i', j') < (i, j)\}$$

where $m = s_{i-1,j}$, $n = s_{i,j-1}$, and $l = s_{i,j}$. The second assumption is that for every state, the feature vectors follow a Gaussian distribution. Once the state of a block is known, the feature vector is conditionally independent of information in other blocks. The covariance matrix Σ_s and the mean vector μ_s of the Gaussian distribution vary with state s.

For the MHMM, denote the collection of resolutions by $\mathcal{R} = \{1, \ldots, R\}$, with $r = R$ being the finest resolution. Let the collection of block indices at resolution r be

$$\mathbb{N}^{(r)} = \{(i, j)\colon 0 \le i < w/2^{R-r},\ 0 \le j < z/2^{R-r}\}.$$

Images are described by feature vectors at all the resolutions, denoted by $u_{i,j}^{(r)}$, $r \in \mathcal{R}$. Every feature vector is labeled with a class $c_{i,j}^{(r)}$. The underlying state of a feature vector is $s_{i,j}^{(r)}$. At each resolution r, the set of states is $\{1^{(r)}, 2^{(r)}, \ldots, M_r^{(r)}\}$. Note that as states vary across resolutions, different resolutions do not share states.

As with the single-resolution model, each state at every resolution is uniquely mapped to one class. On the other hand, a block with a known class may exist in several states. Since a block at a lower resolution contains several blocks at a higher resolution, it may not be of a pure class. Therefore, except for the highest resolution, there is an extra "mixed" class in addition to the original classes. Denote the set of original classes by $\mathcal{G} = \{1, 2, \ldots, G\}$ and the "mixed" class by $G + 1$. Because of the unique mapping between states and classes, the state of a parent block may constrain the possible states for its child blocks. If the state of a parent block is mapped to a determined (nonmixed) class, the child blocks can exist only in states that map to the same class.

To structure statistical dependence among resolutions, a Markov chain with resolution playing a time-like role is assumed in the 2-D MHMM. Given the states and the features at the parent resolution, the states and the features at the current resolution are conditionally independent of the other previous resolutions, so that

$$P\left\{s_{i,j}^{(r)}, u_{i,j}^{(r)}\colon r \in \mathcal{R}, (i, j) \in \mathbb{N}^{(r)}\right\}$$
$$= P\left\{s_{i,j}^{(1)}, u_{i,j}^{(1)}\colon (i, j) \in \mathbb{N}^{(1)}\right\}$$
$$\times P\left\{s_{i,j}^{(2)}, u_{i,j}^{(2)}\colon (i, j) \in \mathbb{N}^{(2)} \middle| s_{k,l}^{(1)}\colon (k, l) \in \mathbb{N}^{(1)}\right\}$$
$$\times \cdots \times P\left\{s_{i,j}^{(R)}, u_{i,j}^{(R)}\colon (i, j) \in \mathbb{N}^{(R)} \middle| s_{k,l}^{(R-1)}\colon (k, l) \in \mathbb{N}^{(R-1)}\right\}. \quad (1)$$

At the coarsest resolution, $r = 1$, feature vectors are assumed to be generated by a single-resolution 2-D HMM. At a higher resolution, the conditional distribution of a feature vector given its state is also assumed to be Gaussian. The parameters of the Gaussian distribution depend upon the state at the particular resolution.

Given the states at resolution $r - 1$, statistical dependence among blocks at the finer resolution r is constrained to sibling blocks (child blocks descended from the same parent block). Specifically, child blocks descended from different parent blocks are conditionally independent. In addition, given the state of a parent block, the states of its child blocks are independent of the states of their "uncle" blocks (nonparent blocks at the parent resolution). State transitions among sibling blocks are governed by the same Markovian property assumed for a single-resolution 2-D HMM. The state transition probabilities, however, depend on the state of their parent block. To formulate these assumptions, denote the child blocks at resolution r of block (k, l) at resolution $r - 1$ by

$$\mathbb{D}(k, l) = \{(2k, 2l), (2k+1, 2l), (2k, 2l+1), (2k+1, 2l+1)\}.$$

According to the assumptions

$$P\left\{s_{i,j}^{(r)}\colon (i, j) \in \mathbb{N}^{(r)} \middle| s_{k,l}^{(r-1)}\colon (k, l) \in \mathbb{N}^{(r-1)}\right\} = \prod_{(k,l)\in\mathbb{N}^{(r-1)}} P\left\{s_{i,j}^{(r)}\colon (i, j) \in \mathbb{D}(k, l) \middle| s_{k,l}^{(r-1)}\right\}$$

where

$$P\{s_{i,j}^{(r)}\colon (i, j) \in \mathbb{D}(k, l) | s_{k,l}^{(r-1)}\}$$

can be evaluated by transition probabilities conditioned on $s_{k,l}^{(r-1)}$, denoted by $a_{m,n,l}(s_{k,l}^{(r-1)})$. We thus have a different set of transition probabilities $a_{m,n,l}$ for every possible state in the parent resolution, and so for the "state process" the resolutions are "minimal" in the sense of Pérez and Heitz [33, Proposition 1]. The influence of previous resolutions is exerted hierarchically through the probability of the states, which can be visualized in Fig. 4. The joint probability of states and feature vectors at all the resolutions in (1) is then derived as

$$P\left\{s_{i,j}^{(r)}, u_{i,j}^{(r)}\colon r \in \mathcal{R}, (i, j) \in \mathbb{N}^{(r)}\right\}$$
$$= P\left\{s_{i,j}^{(1)}, u_{i,j}^{(1)}\colon (i, j) \in \mathbb{N}^{(1)}\right\}$$
$$\times \prod_{r=2}^{R} \prod_{(k,l)\in\mathbb{N}^{(r-1)}} \left(P\left\{s_{i,j}^{(r)}\colon (i, j) \in \mathbb{D}(k, l) \middle| s_{k,l}^{(r-1)}\right\} \times \prod_{(i,j)\in\mathbb{D}(k,l)} P\left\{u_{i,j}^{(r)} \middle| s_{i,j}^{(r)}\right\} \right).$$

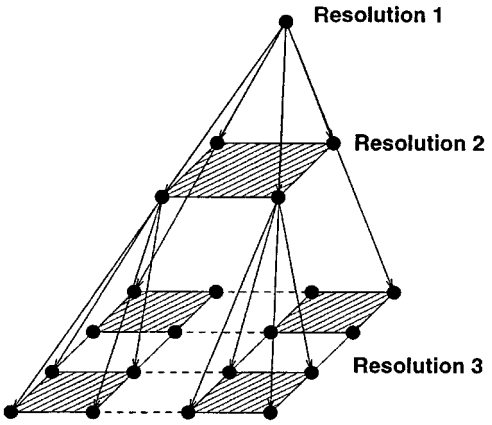


Fig. 4. The hierarchical statistical dependence across resolutions.

To summarize, a 2-D MHMM reflects both the interscale and intrascale statistical dependence. The interscale dependence is modeled by the Markov chain over resolutions. The intrascale dependence is modeled by the HMM. At the coarsest resolution, feature vectors are assumed to be generated by a 2-D HMM. At all the higher resolutions, feature vectors of sibling blocks are also assumed to be generated by 2-D HMMs. The HMMs vary according to the states of parent blocks. Therefore, if the next coarser resolution has M states, then there are, correspondingly, M HMMs at the current resolution. The motivation for having both the inter- and intrascale dependence is discussed in Section III. Experiments in Section VII show the influence of both types of dependence.

III. Related Work

A variety of multiresolution models have been proposed for the purpose of incorporating relatively global information into image classification. One early work in this direction is the multiscale autoregressive model proposed by Basseville *et al.* [3]. Suppose images are represented by R resolutions, with $r = 1$ being the coarsest resolution. Pixel intensities at resolution r are denoted by

$$X^{(r)} = \{X^{(r)}(i, j): (i, j) \in \mathbb{N}^{(r)}\}.$$

Define a coarse-scale shift operator s to reference the *parent* node, and s^k to reference the "ancestor" node k levels higher. Specifically,

$$s^k(i, j) = (\lfloor i/2^k \rfloor, \lfloor j/2^k \rfloor).$$

A homogeneous multiscale autoregressive model has the property that

$$x^{(r)}(i, j) = a_{r-1}x^{(r-1)}(s(i, j)) + a_{r-2}x^{(r-2)}(s^2(i, j)) + \cdots + a_1 x^{(1)}(s^{r-1}(i, j)) + w(i, j), \quad a_k \in \Re$$

where $w(i, j)$ is an independent white driving noise.

As with autoregressive models in other contexts, this model entails a rather constrained dependence, here across resolutions. Recent work has generalized the cross resolution dependence by introducing Gaussian mixture models [7] or hidden Markov models [11]. Bouman and Shapiro proposed the multiscale random field (MSRF) model for images. Suppose an image is described by a random field X. The pixel labels (or classes) at resolution r are $C^{(r)}$, $r = 1, \ldots, R$. The first assumption of the MSRF is the Markovian property across resolutions, i.e.,

$$P\left(C^{(r)} = c^{(r)} \middle| C^{(l)} = c^{(l)}, l < r\right) = P\left(C^{(r)} = c^{(r)} \middle| C^{(r-1)} = c^{(r-1)}\right).$$

The second assumption is the exclusive dependence of X on $C^{(R)}$, that is,

$$P\left(X \in dx \middle| C^{(r)} = c^{(r)}, r = 1, \ldots, R\right) = P\left(X \in dx \middle| C^{(R)} = c^{(R)}\right).$$

For segmentation, the models are restricted to two properties regarding $C^{(r)}$, $r = 1, \ldots, R$. First, the individual classes in $C^{(r)}$ are conditionally independent given the classes in $C^{(r-1)}$. Second, each class in $C^{(r)}$ depends only on classes in a neighborhood at the coarser resolution $r - 1$.

There are three key differences between our 2-D MHMMs and the MSRF models for segmentation [7]. First, in the MSRF, features are observed solely at the finest resolution. The coarser resolutions figure only in prior probabilities of classes. For many applications of image classification [37], [26], it has been found that combining features extracted from several resolutions improves classification. In Section VII, experiments also demonstrate the gain in performance that owes to multiresolution features. Second, states and classes are not distinguished by the MSRF in that every class is considered as one state. At the finest resolution, the conditional distribution of feature vectors given a state is a Gaussian mixture. It is shown [39] that such an HMM is equivalent to a special case of the HMM we assumed, in which every class contains several states, each corresponding to a component of the Guassian mixture. On the other hand, in general, an HMM with multiple states in one class and Gaussian distributions conditioned on states cannot be converted to an HMM with a single state in every class and a Gaussian mixture distribution given each state. Third, the MSRF assumes statistical dependence only across resolutions. In the 2-D MHMM, however, since sibling blocks are dependent given the state of their parent block, interscale and intrascale dependence can be balanced flexibly. With only the interscale dependence, a multiresolution model implies that a parent node completely summarizes context information for its child nodes. However, this assumption need not be true in practice, even approximately. In fact, for many applications, information useful for distinguishing classes is embedded in relatively high-frequency bands. As a result, when the resolution is sufficiently low, a parent node cannot provide any helpful context information.

A 2-D MHMM provides a mechanism to trade interscale and intrascale dependence according to applications. For example, suppose the number of blocks at the finest resolution that a system intends to classify jointly is 8×8. If the HMM assumed for feature vectors at the coarsest resolution examines 2×2 blocks jointly, we need a three-resolution model with quadtree split. If the HMM at the coarsest resolution examines 4×4 blocks jointly, we then need a two-resolution model with quadtree split. Another setup of a two-resolution model might be to replace the quadtree split by a 4×4 split and assume an

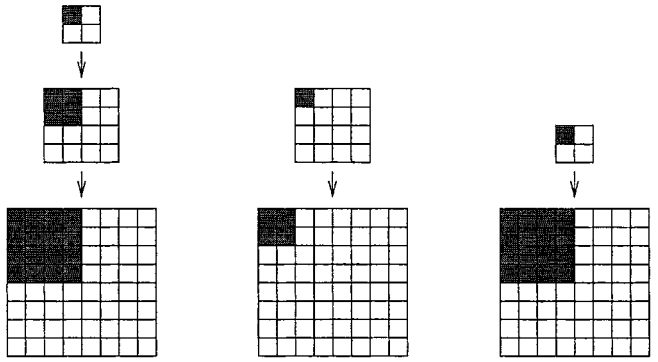


Fig. 5. Three possible structures of MHMMs on an 8×8 grid of blocks. (a) a three-level MHMM with quadtree split and the coarsest resolution modeled by an HMM on a 2×2 grid. (b) a two-level MHMM with quadtree split and the coarsest resolution modeled by an HMM on a 4×4 grid. (c) a two-level MHMM with 4×4 split and the coarsest resolution modeled by an HMM on a 2×2 grid.

HMM on 2×2 blocks at the coarse resolution. The three possibilities of the MHMM are shown in Fig. 5. All the parameters in the model structure setup can be chosen conveniently as inputs to algorithms for training and testing.

Another multiresolution model based on HMMs is the model proposed for wavelet coefficients by Crouse *et al.* [11], where wavelet coefficients across resolutions are assumed to be generated by one-dimensional hidden Markov models with resolution being the time-like role in the Markov chain. If we view wavelet coefficients as special cases of features, the model in [11] considers features observed at multiple resolutions. However, intrascale dependence is not pursued in depth in [11]. This wavelet-domain model is applied to image segmentation [9] and is extended to general features in [29].

The approach of applying models to image segmentation in [9] is different from that of Bouman and Shapiro [7] and ours. States in wavelet-domain HMMs are not related to classes. In particular, there are two states at every resolution, one representing a wavelet coefficient being large and the other small. To segment images, a separate HMM is trained for each class. A local region in an image is regarded as an instance of a random process described by one of the HMMs. To decide the class of the local region, likelihood is computed using the HMM of each class, and the class yielding the maximum likelihood is selected. The whole image is then segmented by combining decisions for all the local regions. It is not straightforward for such an approach to account for the spatial dependence among classes in an image. Furthermore, the wavelet-domain HMMs alone do not provide a natural mechanism to incorporate segmentation results at multiple resolutions. A remedy, specifically context-based interscale fusion, is developed in [9] to address this issue. In Bouman and Shapiro [7] as well as our paper, however, an entire image is regarded as an instance of a 2-D random process characterized by one model, which reflects the transition properties among classes/states at all the resolutions as well as the dependence of feature vectors on classes/states. The set of classes or states with the maximum *a posteriori* probability is sought according to the model. Segmenting an image by combining features at multiple resolutions is inherent in our algorithm based on 2-D MHMMs. As the number of states and the way of extracting features are allowed to vary with resolution, it is flexible enough to incorporate multiscale information for classification using 2-D MHMMs.

In computer vision, there has been much work on learning vision by image modeling [20], [22], [17]. Particularly, in [17], multiresolution modeling is applied to estimate motions from image frames. Bayesian network techniques [5], [32], [19] have played an important role in learning models in computer vision. Theories of Bayesian networks also provide guidance on how to construct models with tractable learning complexity. Exact inference on general Bayesian networks is NP-hard, as discussed by Cooper [10]. Computationally efficient algorithms for training a general Bayesian network are not always available. As we have constructed 2-D MHMMs by extending 1-D HMMs used in speech recognition [39], efficient algorithms for training and applying 2-D MHMMs are derived from the expectation-maximalization (EM) algorithm [13] and related techniques developed for speech recognition.

IV. The Algorithm

The parameters of the multiresolution model are estimated iteratively by the EM algorithm [13]. To ease the enormous computational burden, we apply a suboptimal estimation procedure: the Viterbi training algorithm [39]. At every iteration, the combination of states at all the resolutions with the maximum *a posteriori* (MAP) probability is searched by the Viterbi algorithm [38]. These states are then assumed to be real states to update the estimation of parameters. Because of the multiple resolutions, a certain part of the training algorithm used for the single-resolution HMM [25] is changed to a recursive procedure. For completeness, we present the EM algorithm for estimating the parameters of a 2-D HMM as described in [25]. Next, the EM estimation is approximated by the Viterbi training algorithm, which is then extended to the case of a 2-D MHMM.

For a single-resolution HMM, suppose that the states are $s_{i,j}$, $1 \le s_{i,j} \le M$, the class labels are $c_{i,j}$, and the feature vectors are $u_{i,j}$, $(i, j) \in \mathbb{N}$, a generic index set. Denote

1) the set of observed feature vectors for the entire image by $\boldsymbol{u} = \{u_{i,j}\colon (i, j) \in \mathbb{N}\}$;
2) the set of states for the image by $\boldsymbol{s} = \{s_{i,j}\colon (i, j) \in \mathbb{N}\}$;
3) the set of classes for the image by $\boldsymbol{c} = \{c_{i,j}\colon (i, j) \in \mathbb{N}\}$;
4) the mapping from a state $s_{i,j}$ to its class by $C(s_{i,j})$, and the set of classes mapped from states $\boldsymbol{s}$ by $C(\boldsymbol{s})$; and
5) the model estimated at iteration p by $\phi^{(p)}$.

The EM algorithm iteratively improves the model estimation by the following steps:

1) Given the current model estimate $\phi^{(p)}$, the observed feature vectors $u_{i,j}$, and classes $c_{i,j}$, $(i, j) \in \mathbb{N}$, the mean vectors and covariance matrices are updated by

$$\mu_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i, j) u_{i,j}}{\sum_{i,j} L_m^{(p)}(i, j)} \tag{2}$$

$$\Sigma_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i,j)\left(u_{i,j}-\mu_m^{(p+1)}\right)\left(u_{i,j}-\mu_m^{(p+1)}\right)^t}{\sum_{i,j} L_m^{(p)}(i,j)}. \quad (3)$$

$L_m^{(p)}(i,j)$ is the *a posteriori* probability of block (i,j) being in state m, calculated by

$$L_m^{(p)}(i,j) = \sum_{\boldsymbol{s}} I(m=s_{i,j}) \cdot \frac{1}{\alpha} I(C(\boldsymbol{s})=\boldsymbol{c}) \times \prod_{(i,j)\in\mathbb{N}} a^{(p)}_{s_{i-1,j},s_{i,j-1},s_{i,j}} \times \prod_{(i,j)\in\mathbb{N}} P\left(u_{i,j}\left|\mu^{(p)}_{s_{i,j}}, \Sigma^{(p)}_{s_{i,j}}\right.\right) \quad (4)$$

where $I(\cdot)$ is the indicator function and α is a normalization constant.

2) The transition probabilities are updated by

$$a^{(p+1)}_{m,n,l} = \frac{\sum_{i,j} H^{(p)}_{m,n,l}(i,j)}{\sum_{l=1}^{M}\sum_{i,j} H^{(p)}_{m,n,l}(i,j)}. \quad (5)$$

$H^{(p)}_{m,n,l}(i,j)$ is the *a posteriori* probability of block (i,j) being in state l, $(i-1,j)$ in state m, and $(i,j-1)$ in state n, which is calculated by

$$H^{(p)}_{m,n,l}(i,j) = \sum_{\boldsymbol{s}} I(m=s_{i-1,j},\, n=s_{i,j-1}, l=s_{i,j}) \times \frac{1}{\alpha'} I(C(\boldsymbol{s})=\boldsymbol{c}) \times \prod_{(i,j)\in\mathbb{N}} a^{(p)}_{s_{i-1,j},s_{i,j-1},s_{i,j}} \times \prod_{(i,j)\in\mathbb{N}} P\left(u_{i,j}\left|\mu^{(p)}_{s_{i,j}}, \Sigma^{(p)}_{s_{i,j}}\right.\right) \quad (6)$$

where α' is a normalization constant.

The computation of $L_m(i,j)$ and $H_{m,n,l}(i,j)$ is prohibitive even with the extension of the forward and backward probabilities [39] to only two dimensions. To simplify the calculation of $L_m(i,j)$ and $H_{m,n,l}(i,j)$, it is assumed that the single most likely state sequence accounts for virtually all the likelihood of the observations (MAP rule), which entails the Viterbi training algorithm. We thus aim at finding the optimal state sequence to maximize $P(\boldsymbol{s}|\boldsymbol{u}, \boldsymbol{c}, \phi^{(p)})$, which is accomplished by the Viterbi algorithm. Assume

$$\boldsymbol{s}_{\text{opt}} = \max_{\boldsymbol{s}}^{-1} P(\boldsymbol{s}|\boldsymbol{u}, \boldsymbol{c}, \phi^{(p)}).$$

Then $L_m(i,j)$ and $H_{m,n,l}(i,j)$ are trivially approximated by

$$L_m(i,j) = I\left(m = s_{\text{opt},(i,j)}\right)$$
$$H_{m,n,l}(i,j) = I\left(m = s_{\text{opt},(i-1,j)},\, n = s_{\text{opt},(i,j-1)},\, l = s_{\text{opt},(i,j)}\right).$$

The key step in training is converted to searching $\boldsymbol{s}_{\text{opt}}$ by the MAP rule. To simplify expressions, the conditional variables $\boldsymbol{c}$ and $\phi^{(p)}$ are omitted from $P(\boldsymbol{s}|\boldsymbol{u}, \boldsymbol{c}, \phi^{(p)})$ in the sequel. By default, $\boldsymbol{s}_{\text{opt}}$ is computed on the basis of the current model estimate $\phi^{(p)}$. It is shown [25] that

$$\max_{\boldsymbol{s}}{}^{-1} P(\boldsymbol{s}|\boldsymbol{u}, \boldsymbol{c}) = \max_{\boldsymbol{s}:\, C(\boldsymbol{s})=\boldsymbol{c}}{}^{-1} P(\boldsymbol{s}|\boldsymbol{u}).$$

Therefore, $\boldsymbol{c}$ is omitted from the condition by assuming that $\boldsymbol{s}_{\text{opt}}$ is searched among $\boldsymbol{s}$ satisfying $C(\boldsymbol{s}) = \boldsymbol{c}$, i.e., $C(s_{i,j}) = c_{i,j}$ for all $(i,j) \in \mathbb{N}$. Note that the maximization of $P(\boldsymbol{s}|\boldsymbol{u})$ is equivalent to maximizing $P(\boldsymbol{s}, \boldsymbol{u})$. $P(\boldsymbol{s}, \boldsymbol{u})$ can be expanded

$$\begin{aligned} &P\{s_{i,j}, u_{i,j}: (i,j)\in\mathbb{N}\} \\ &\quad = P\{s_{i,j}: (i,j)\in\mathbb{N}\} \\ &\qquad \times P\{u_{i,j}: (i,j)\in\mathbb{N} | s_{i,j}: (i,j)\in\mathbb{N}\} \\ &\quad = P\{s_{i,j}: (i,j)\in\mathbb{N}\} \times \prod_{(i,j)\in\mathbb{N}} P(u_{i,j}|s_{i,j}) \\ &\quad = P(T_0)\times P(T_1|T_0)\times P(T_2|T_1) \\ &\qquad \times\cdots\times P(T_{w+z-2}|T_{w+z-3}) \prod_{(i,j)\in\mathbb{N}} P(u_{i,j}|s_{i,j}) \end{aligned} \quad (7)$$

where T_d denotes the sequence of states for blocks lying on diagonal d, $\{s_{d,0}, s_{d-1,1}, \times, s_{0,d}\}$, as is shown in Fig. 6.

Since T_d serves as an "isolating" element in the expansion of $P\{s_{i,j}: (i,j)\in\mathbb{N}\}$, the Viterbi algorithm can be applied straightforwardly to find the combination of states maximizing the likelihood $P\{s_{i,j}, u_{i,j}: (i,j)\in\mathbb{N}\}$. What differs here from the normal Viterbi algorithm is that the number of possible sequences of states at every position in the Viterbi transition diagram increases exponentially with the increase in number of blocks in T_d. If there are M states, the amount of computation and memory are both of order M^ν, where ν is the number of states in T_d. Fig. 7 shows an example. Hence, this version of

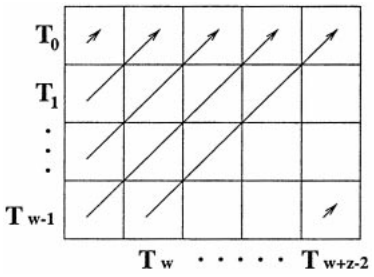


Fig. 6. Blocks on the diagonals of an image.

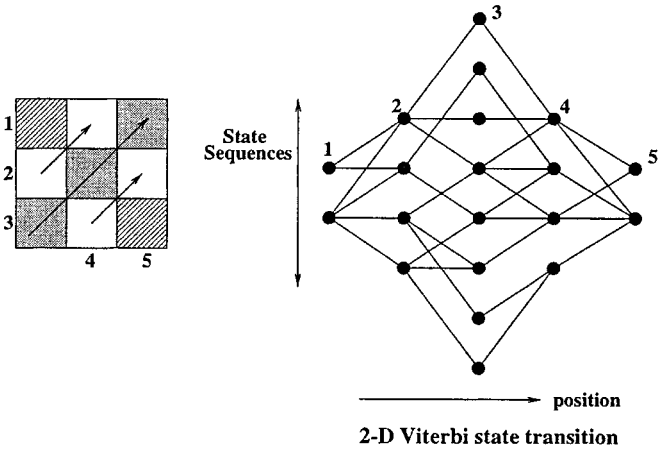


Fig. 7. The variable-state Viterbi algorithm.

the Viterbi algorithm is referred to as a variable-state Viterbi algorithm.

Next, we extend the Viterbi training algorithm to multiresolution models. Denote

$$\boldsymbol{s}^{(r)} = \{s_{i,j}^{(r)}: (i, j) \in \mathbb{N}^{(r)}\}$$
$$\boldsymbol{u}^{(r)} = \{u_{i,j}^{(r)}: (i, j) \in \mathbb{N}^{(r)}\}$$

and

$$\boldsymbol{c}^{(r)} = \{c_{i,j}^{(r)}: (i, j) \in \mathbb{N}^{(r)}\}.$$

If the superscript (r) is omitted, for example, $\boldsymbol{s}$, it denotes the collection of $\boldsymbol{s}^{(r)}$ over all $r \in \mathcal{R}$.

The Viterbi training algorithm searches for

$$\boldsymbol{s}_{\text{opt}} = \max_{\boldsymbol{s}:\, C(\boldsymbol{s})=\boldsymbol{c}}^{-1} P(\boldsymbol{s}|\boldsymbol{u}).$$

As mentioned previously, the set of original classes is $\mathcal{G} = \{1, 2, \ldots, G\}$. The "mixed" class is denoted by $G+1$. At the finest resolution R, $c_{i,j}^{(R)} \in \mathcal{G}$ is given by the training data. At a coarser resolution $r < R$, $c_{i,j}^{(r)}$ is determined by the recursive formula

$$c_{i,j}^{(r)} = \begin{cases} g, & c_{k,l}^{(r+1)} = g \text{ for all } (k, l) \in \mathbb{D}(i, j) \\ G+1, & \text{otherwise.} \end{cases}$$

Equivalent to the above recursion, $c_{i,j}^{(r)} = g$, $g \in \mathcal{G}$ if all the descending blocks of (i, j) at the finest resolution R are of class g. Otherwise, if different classes occur, $c_{i,j}^{(r)} = G+1$. By assigning $c_{i,j}^{(r)}$ in such a way, consistency on the mapping from states to classes at multiple resolutions is enforced in that if $c_{i,j}^{(r)} = g$, $g \in \mathcal{G}$, the probability that $C(s_{k,l}^{(r+1)}) \neq g$, for any $(k, l) \in \mathbb{D}(i, j)$, is assigned 0.

To clarify matters, we present a case with two resolutions. By induction, the algorithm extends to models with more than two. According to the MAP rule, the optimal set of states maximizes the joint log-likelihood of all the feature vectors and states

$$\begin{aligned} &\log P\left\{s_{k,l}^{(r)}, u_{k,l}^{(r)}: r \in \{1, 2\}, (k, l) \in \mathbb{N}^{(r)}\right\} \\ &\quad = \log P\left\{s_{k,l}^{(1)}, u_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\right\} \\ &\qquad + \log P\left\{s_{i,j}^{(2)}, u_{i,j}^{(2)}: (i, j) \in \mathbb{N}^{(2)} \,\middle|\, s_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\right\} \\ &\quad = \log P\left\{s_{k,l}^{(1)}, u_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\right\} \\ &\qquad + \sum_{(k,l) \in \mathbb{N}^{(1)}} \log P\left\{s_{i,j}^{(2)}, u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) \,\middle|\, s_{k,l}^{(1)}\right\}. \end{aligned}$$

The algorithm works backward to maximize the above log-likelihood. First, for each $s_{k,l}^{(1)}$ and each $(k, l) \in \mathbb{N}^{(1)}$

$$\{\overline{s}_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l)\}$$

is searched to maximize

$$\log P\{s_{i,j}^{(2)}, u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) | s_{k,l}^{(1)}\}.$$

Since given $s_{k,l}^{(1)}$, the child blocks at Resolution 2 are governed by a single resolution 2-D HMM with transition probabilities $a_{m,n,l}(s_{k,l}^{(1)})$, the variable-state Viterbi algorithm [25] can be applied directly. In order to make clear that $\overline{s}_{i,j}^{(2)}$ depends on $s_{k,l}^{(1)}$, we often write $\overline{s}_{i,j}^{(2)}(s_{k,l}^{(1)})$. The next step is to maximize

$$\begin{aligned} &\log P\left\{s_{k,l}^{(1)}, u_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\right\} \\ &\quad + \sum_{(k,l) \in \mathbb{N}^{(1)}} \log P\left\{\overline{s}_{i,j}^{(2)}(s_{k,l}^{(1)}), u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) \,\middle|\, s_{k,l}^{(1)}\right\} \\ &= \sum_{\tau} \Bigg[\log P\left(T_{\tau}^{(1)} \,\middle|\, T_{\tau-1}^{(1)}\right) \\ &\qquad + \sum_{(k,l):\, \Delta(k,l)=\tau} \Bigg(\log P\left(u_{k,l}^{(1)} \,\middle|\, s_{k,l}^{(1)}\right) \\ &\qquad + \log P\left\{\overline{s}_{i,j}^{(2)}\left(s_{k,l}^{(1)}\right), u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) \,\middle|\, s_{k,l}^{(1)}\right\} \Bigg) \Bigg] \end{aligned} \tag{8}$$

Equation (8) follows from (7). As in (7), $T_{\tau}^{(1)}$ denotes the sequence of states for blocks on diagonal τ in Resolution 1. We can apply the variable-state Viterbi algorithm again to search for the optimal $s_{i,j}^{(1)}$ since $T_{\tau}^{(1)}$ still serves as an "isolating" element in the expansion. The only difference with the maximization of (7) is the extra term

$$\log P\left\{\overline{s}_{i,j}^{(2)}\left(s_{k,l}^{(1)}\right), u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) \,\middle|\, s_{k,l}^{(1)}\right\}$$

which is already computed and stored as part of the first step.

Provided with the $\boldsymbol{s}_{\text{opt}}$, parameters are estimated by equations similar to (2), (3), and (5). For notational simplicity, the superscripts (p) and $(p+1)$ denoting iterations are replaced by (r) to denote the resolution. At each resolution r, $r \in \mathcal{R}$, the parameters are updated as shown in (9)–(11) at the top of the following page, where (i', j') is the parent block of (i, j) at resolution $r-1$. For quadtree split, $i' = \lfloor i/2 \rfloor$, $j' = \lfloor j/2 \rfloor$.

In the model-testing process, that is, applying a 2-D MHMM to classify an image, the MAP states $\boldsymbol{s}_{\text{opt}}$ is searched first. Because the training algorithm guarantees the consistency on class mapping across resolutions, to derive classes from states, we only need to map the states at the finest resolution, $s_{i,j}^{(R)}$, $(i, j) \in \mathbb{N}^{(R)}$, into corresponding classes. The algorithm used to search $\boldsymbol{s}_{\text{opt}}$ in training can be applied directly to testing. The only difference is that the constraint $C(\boldsymbol{s}_{\text{opt}}) = \boldsymbol{c}$ is removed since $\boldsymbol{c}$ is to be determined.

V. Fast Algorithms

As states across resolutions are statistically dependent, to determine the optimal states according to the MAP rule, joint consideration of all resolutions is necessary. However, the hierarchical structure of the multiresolution model is naturally suited to progressive classification if we relax the MAP rule. Suboptimal fast algorithms are developed by discarding joint considerations and searching for states in a layered fashion. States in the lowest resolution are determined only by feature vectors in this

$$\mu_m^{(r)} = \frac{\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i,j}^{(r)}\right) u_{i,j}}{\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i,j}^{(r)}\right)} \tag{9}$$

$$\Sigma_m^{(r)} = \frac{\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i,j}^{(r)}\right)\left(u_{i,j} - \mu_m^{(r)}\right)\left(u_{i,j} - \mu_m^{(r)}\right)^t}{\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i,j}^{(r)}\right)} \tag{10}$$

$$a_{m,n,l}^{(r)}(k) = \frac{\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i-1,j}^{(r)}, n = s_{i,j-1}^{(r)}, l = s_{i,j}^{(r)}\right)\cdot I\left(k = s_{i',j'}^{(r-1)}\right)}{\sum_{l=1}^{M}\sum_{(i,j):\,(i,j)\in\mathbb{N}^{(r)}} I\left(m = s_{i-1,j}^{(r)}, n = s_{i,j-1}^{(r)}, l = s_{i,j}^{(r)}\right)\cdot I\left(k = s_{i',j'}^{(r-1)}\right)} \tag{11}$$

resolution. A classifier searches for the state of a child block in the higher resolution only if the class of its parent block is "mixed."

As one block at a lower resolution covers a larger region in the original image, making decisions at the lower resolution reduces computation. On the other hand, the existence of the "mixed" class warns the classifier of ambiguous areas that need examination at higher resolutions. As a result, the degradation of classification due to the low resolution is avoided to a certain extent. Two fast algorithms are proposed.

A. Fast Algorithm 1

Use the two-resolution case in the previous section as an example. To maximize

$$\log P\left\{s_{k,l}^{(r)}, u_{k,l}^{(r)}: r \in \{1, 2\}, (k, l) \in \mathbb{N}^{(r)}\right\}$$

the first step of Fast Algorithm 1 searches for $\{\overline{s}_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\}$ that maximizes

$$\log P\{s_{k,l}^{(1)}, u_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\}.$$

For any $\overline{s}_{k,l}^{(1)}$, if it is mapped into the "mixed" class, the second step searches for $\{\overline{s}_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l)\}$ that maximizes

$$\log P\{s_{i,j}^{(2)}, u_{i,j}^{(2)}: (i, j) \in \mathbb{D}(k, l) | \overline{s}_{k,l}^{(1)}\}.$$

Although the algorithm is "greedy" in the sense that it searches for the optimal states at each resolution, it does not give the overall optimal solution generally since the resolutions are statistically dependent.

B. Fast Algorithm 2

The second fast algorithm trains a sequence of single-resolution HMMs, each of which is estimated using features and classes in a particular resolution. Except for the finest resolution, there is a "mixed" class. To classify an image, the first step is the same as that of Fast Algorithm 1: search for $\{\overline{s}_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\}$ that maximizes

$$\log P\{s_{k,l}^{(1)}, u_{k,l}^{(1)}: (k, l) \in \mathbb{N}^{(1)}\}.$$

In the second step, context information obtained from the first resolution is used, but differently from Fast Algorithm 1. Suppose $\overline{s}_{k,l}^{(1)}$ is mapped into class "mixed," to decide $s_{i,j}^{(2)}$, $(i, j) \in \mathbb{D}(k, l)$, we form a neighborhood of (i, j), $\mathbb{B}(i, j)$, which contains $\mathbb{D}(k, l)$ as a subset. We then search for the combination of states in $\mathbb{B}(i, j)$ that maximizes the *a posteriori* probability given features in this neighborhood according to the model at Resolution 2. Since the classes of some blocks in the neighborhood may have been determined by the states of their parent blocks, the possible states of those blocks are constrained to be mapped into the classes already known. The limited choices of these states, in turn, affect the selection of states for blocks whose classes are to be decided.

There are many possibilities to choose the neighborhood. In our experiments, particularly, the neighborhood is a 4×4 grid of blocks. For simplicity, the neighborhood of a block is not necessarily centered around the block. Blocks in the entire image are predivided into 4×4 groups. The neighborhood of each block is the group to which the block belongs.

VI. Comparison of Complexity with 2-D HMM

To show that the multiresolution HMM saves computation by comparison with the single-resolution HMM, we analyze quantitatively the order of computational complexity for both cases. Assume that the Viterbi algorithm without path constraints is used to search for the MAP states so that we have a common ground for comparison.

For the single-resolution HMM, recall that the Viterbi algorithm is used to maximize the joint log-likelihood of all the states and features in an image according to (7)

$$\begin{aligned}\log P\{s_{i,j}, u_{i,j}: (i, j)\in\mathbb{N}\} &= \log P(T_0) + \log P(u_{0,0}|T_0) + \cdots \\ &\quad + \sum_{\tau=1}^{w+z-2}\left(\log P(T_\tau|T_{\tau-1}) + \sum_{(i,j):\,\Delta(i,j)=\tau} P(u_{i,j}|s_{i,j})\right)\end{aligned}$$

where T_τ is the sequence of states for blocks on diagonal τ, and w, or z is the number of rows, or columns in the image. For simplicity, assume that $w = z$. Every node in the Viterbi transition diagram (Fig. 7) corresponds to a state sequence T_τ, and every transition step corresponds to one diagonal τ. Therefore, there are in total $2w - 1$ transition steps in the Viterbi algorithm. Denote the number of blocks on diagonal τ by $n(\tau)$

$$n(\tau) = \begin{cases} \tau + 1, & 0 \le \tau \le w - 1 \\ 2w - \tau - 1, & w \le \tau \le 2w - 2. \end{cases}$$

The number of nodes at step τ is $M^{n(\tau)}$, where M is the number of states.

For each node at step τ, a node in the preceding step is chosen so that the path passing through the node yields the maximum likelihood up to step τ. Suppose the amount of computation for calculating accumulated cost from one node in step $\tau-1$ to one node in step τ is $\gamma(\tau)$. Since $\gamma(\tau)$ increases linearly with the number of blocks on diagonal τ, we write

$$\gamma(\tau) = c_1 n(\tau) + c_2.$$

The computation at step τ is thus $M^{n(\tau)}M^{n(\tau-1)}\gamma(\tau)$. The total computation for the Viterbi algorithm is

$$\sum_{\tau=1}^{2w-2} M^{n(\tau)}M^{n(\tau-1)}\gamma(\tau)$$
$$= ((2w-1)c_1 + 2c_2)\frac{M^{2w+1}}{M^2-1} - 2c_1\frac{M^{2w+1}}{(M^2-1)^2}$$
$$- (2c_2 - c_1)\frac{M}{M^2-1} + 2c_1\frac{M}{(M^2-1)^2} - c_2 M.$$

If M is sufficiently large so that $M^2-1 \approx M^2$ and $(1/M) \approx 0$, we simplify the above to

$$\sum_{\tau=0}^{2w-2} M^{n(\tau)}M^{n(\tau-1)}\gamma(\tau)$$
$$\approx ((2w-1)c_1 + 2c_2)M^{2w-1} - 2c_1 M^{2w-3} - c_2 M.$$

The computation is thus of order $O(wM^{2w-1})$.

For the multiresolution model, considering the two-resolution case, in the first step the Viterbi algorithm is applied to subimages $\mathbb{D}(k, l)$ to search for $\{\overline{s}^{(2)}_{i,j}\colon (i, j) \in \mathbb{D}(k, l)\}$ that maximize

$$\log P\{s^{(2)}_{i,j}, u^{(2)}_{i,j}\colon (i, j) \in \mathbb{D}(k, l) | s^{(1)}_{k,l}\}.$$

For a fixed $(k, l) \in \mathbb{N}^{(1)}$ and a fixed state $s^{(1)}_{k,l}$, since $\mathbb{D}(k, l)$ is of size 2×2, the amount of computation needed for $\{\overline{s}^{(2)}_{i,j}\colon (i, j) \in \mathbb{D}(k, l)\}$ is of order $O(M_2^3)$, where M_2 is the number of states at Resolution 2. The total computation for the first step is then of order $M_2^3 \cdot (w/2)^2 \cdot M_1$, where M_1 is the number of states at Resolution 1. Since in the second step the Viterbi algorithm is applied to an image of size $(w/2) \times (w/2)$, the computation for the second step is of order $(w/2)M_1^{w-1}$. If w is sufficiently large and M_1 and M_2 are about the same as M, the total computation for the multiresolution model is of order $O(wM^{w-1})$. Therefore, the multiresolution model reduces the amount of computation by order M^w.

Since computational order increases exponentially with w, the cardinality of the side of an image, we usually divide the image into subimages with side size w_0 and classify the subimages separately. The computational order for the single-resolution HMM is reduced to $O((\frac{w}{w_0})^2 w_0 M^{2w_0-1})$, which is $O(w^2 M^{2w_0-1})$ if w_0 is fixed. For the multiresolution HMM, the computational order of the second step becomes $(\frac{w}{w_0})^2 \frac{w_0}{2} M^{w_0-1}$, which does not dominate the computation in the first step if $w_0 - 1 \leq 4$. Hence the total computational order is $O(w^2 M^{\max\{w_0-1, 4\}})$.

In practice, the path-constrained Viterbi algorithm [25], which preselects N nodes at each step for candidate paths, is applied to further reduce complexity. Since complexity varies significantly by changing parameters, including w_0 and N, computational time will be compared based on experiments in the next section.

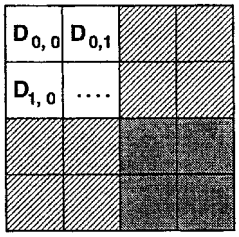


Fig. 8. DCT coefficients of a 4×4 image block

The comparison of complexity discussed above is for the testing process. In training, because more parameters need be estimated for the multiresolution model, a larger set of training data is required. As a result, the relative complexity of the multiresolution model in training is higher than in testing. In fact, if the number of states for each class is fixed across resolutions, with every increased resolution, the number of transition probabilities at the previous coarsest resolution increases by a fixed factor. Furthermore, Gaussian distributions and transition probabilities at the new coarsest resolution add parameters to be estimated. Therefore, the total number of parameters increases linearly with resolutions at a very high rate. Practically, however, in our applications the number of states at a resolution usually declines when the resolution becomes coarser since images tend to be more homogeneous at coarser resolutions. In addition, the intrascale dependence assumed in the 2-D MHMM allows adequate amount of context information to be used without driving the number of resolutions very high.

VII. Experiments

We applied our algorithm to the segmentation of man-made and natural regions of aerial images. The images are 512×512 gray-scale images with 8 bits per pixel (bpp). They are aerial images of the San Francisco Bay area provided by TRW (formerly ESL, Inc.) [30], [31]. An example of an image and its hand-labeled classified companion are shown in Fig. 9(a) and (b).

Feature vectors were extracted at three resolutions. Images at the two low resolutions were obtained by the Daubechies 4 [12] wavelet transform. The images at Resolution 1 and 2 are, respectively, the LL bands of the two-level and one-level wavelet transforms. At each resolution, the image was divided into 4×4 blocks, and DCT coefficients or averages over some of them were used as features. There are six such features. Denote the DCT coefficients for a 4×4 block by $\{D_{i,j}\colon i, j \in (0, 1, 2, 3)\}$, shown by Fig. 8. The six features are defined as

1) $f_1 = D_{0,0}$; $f_2 = |D_{1,0}|$; $f_3 = |D_{0,1}|$;
2) $f_4 = \sum_{i=2}^{3}\sum_{j=0}^{1}|D_{i,j}|/4$;
3) $f_5 = \sum_{i=0}^{1}\sum_{j=2}^{3}|D_{i,j}|/4$;
4) $f_6 = \sum_{i=2}^{3}\sum_{j=2}^{3}|D_{i,j}|/4$.

DCT coefficients at various frequencies reflect variation patterns in a block. They are more efficient than space-domain pixel intensities for distinguishing classes. Alternative features based on frequency properties include wavelet coefficients.

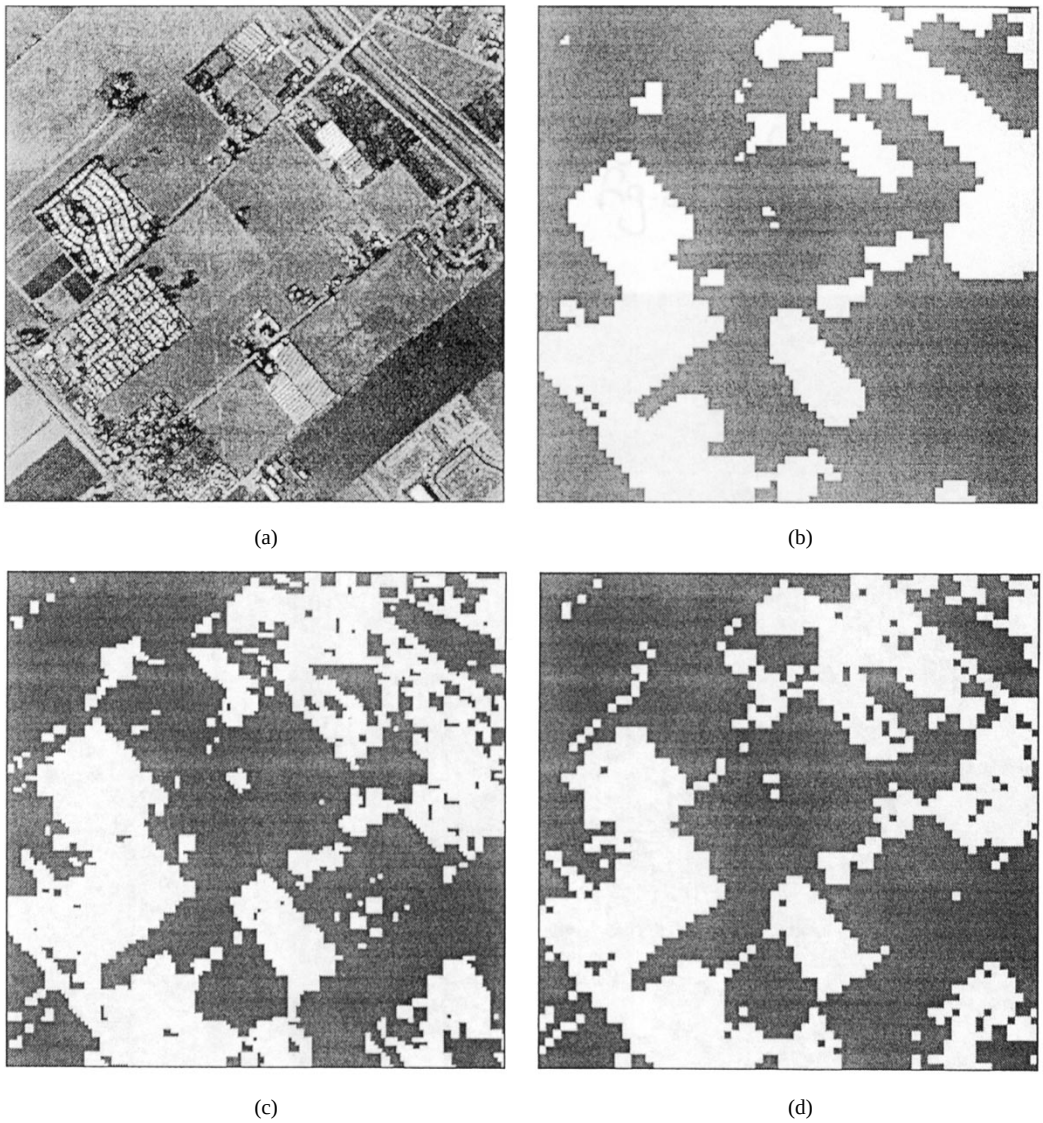


Fig. 9. A sample aerial image. (a) Original. (b) Hand-labeled classes. (c) Single-resolution HMM. (d) Three-level MHMM. White: man-made, Gray: natural.

In addition to the intrablock features computed from pixels within a block, the spatial derivatives of the average intensity values of blocks were used as interblock features. In particular, the spatial derivative refers to the difference between the average intensity of a block and that of the block's upper neighbor or left neighbor. The motivation for using interblock features is similar to that for delta and acceleration coefficients in speech recognition [39], [25].

The MHMM algorithm and its two fast versions were tested by sixfold cross-validation [36], [6]. For each iteration, one image was used as test data and the other five as training data. Performance is evaluated by averaging over all iterations. Under a given computational complexity constraint, the number of states in each class can be chosen according to the principle of Minimum Description Length [2]. The automatic selection of those parameters has not been explored deeply in our current algorithm. Experiments, which will be described, show that with a fairly small number of states, the MHMM algorithm outperforms the single-resolution HMM algorithm and other algorithms.

A 2-D MHMM tested is described by Table I (top), which lists the number of states assigned for each class at each resolution. With the MHMM, the average classification error rate computed at the finest resolution is 16.02%. To compare with well-known algorithms, we also used CART® [6], a decision tree algorithm, and LVQ1, version 1 of Kohonen's learning vector quantization algorithm [21], to segment the aerial images. Classification based on a single-resolution HMM with five states for the natural class and nine states for the man-made class was also performed. All these algorithms were applied to feature vectors formed at the finest resolution in the same way as those used for the 2-D MHMM. Both the average classification error rates and the error rates for each testing image in the sixfold cross-validation are listed in Table II. It is shown that the MHMM algorithm achieves lower error rates for all the testing images than the HMM algorithm, CART, and LVQ1. On average, CART and LVQ1 perform about equally well. In [34], the Bayes VQ algorithm was used to segment the aerial images. BVQ achieves an error rate of about 21.5%, nearly the same as that of CART.

TABLE I
THE NUMBER OF STATES FOR EACH CLASS AT EACH RESOLUTION

Class	res 1	res 2	res 3
natural	5	5	5
man-made	5	5	9
mixed	4	2	0

Class	res 1	res 2	res 3
natural	1	1	5
man-made	1	1	9
mixed	1	1	0

Class	res 1	res 2	res 3
natural	2	2	5
man-made	2	2	9
mixed	2	2	0

The HMM algorithm improves CART and LVQ1 by roughly 13%. The MHMM algorithm further improves the HMM by 15%.

The segmentation results for an example image are shown in Fig. 9(c) and (d). We see that the classified image based on the MHMM is both visually cleaner and closer to the hand-labeled classes in terms of classification error rates. The classification error rate achieved by the MHMM for this image is 11.57%, whereas the error rate for a single-resolution HMM is 13.39%.

As is mentioned in Section III, some multiresolution models consider only interscale statistical dependence. To test whether the intrascale dependence assumed in the 2-D MHMM is redundant given the interscale dependence, a 2-D MHMM discarding the intrascale dependence was evaluated by sixfold cross-validation. With this new model, given the state of a block, the states of its child blocks are independent. The number of states assigned to each class at each resolution in the new 2-D MHMM as well as all the other parameters controlling the computational complexity of the algorithm are the same as those used for the previous MHMM. The average error rate achieved by the new model is 17.26%, whereas the average error rate with the previous model is 16.02%. The experiment thus has demonstrated that the intrascale dependence makes improvement on classification in addition to the interscale dependence.

To compare with existing multiresolution models, consider the quadtree MSRF developed by Bouman and Shapiro [7]. The quadtree model assumes that, at the finest resolution, the probability density function of every class is a Gaussian mixture, which is equivalent to an HMM with several states in one class each corresponding to a component of the mixture [39]. At all the coarse resolutions, since features do not exist and only the prior probabilities of classes are considered, each class can be viewed as one state. Consequently, we examined a 2-D MHMM with parameters shown in Table I (bottom left). As the quadtree model ignores intrascale dependence, the 2-D MHMM was trained with the intrascale dependence dismissed. Such a 2-D MHMM has the same underlying state process as the quadtree model. Since the MSRF assumes features observed only at the finest resolution, when applying the 2-D MHMM to classification, we blocked the effect of features at the two coarse resolutions and only used the prior probabilities of classes for computing the joint *a posteriori* probability of states. The classification error rate obtained by cross-validation is 18.89%, higher than the error rate obtained with the HMM.

TABLE II
CLASSIFICATION ERROR RATES AT THE FINEST RESOLUTION BY DIFFERENT ALGORITHMS

Iteration	CART	LVQ1	HMM	MHMM	Fast 1	Fast 2
1	0.2263	0.2161	0.1904	0.1733	0.1886	0.1855
2	0.1803	0.1918	0.1765	0.1636	0.1566	0.1392
3	0.2899	0.2846	0.2034	0.1782	0.2914	0.2857
4	0.2529	0.2492	0.2405	0.2051	0.2430	0.2277
5	0.1425	0.1868	0.1834	0.1255	0.1906	0.1541
6	0.2029	0.1813	0.1339	0.1157	0.1816	0.1766
Ave.	0.2158	0.2183	0.1880	0.1602	0.2086	0.1948

For this 2-D MHMM, when features at the coarse resolutions are used, the error rate is reduced to 17.63%.

Although a more advanced MSRF, namely, the pyramid graph model, is also explored in [7] for segmentation, comparison is constrained to the quadtree model because equivalence cannot be established between the pyramid graph model and a 2-D MHMM. The former assumes that the state of a block depends on the states of blocks in a neighborhood at the next coarser scale, while the latter assumes dependence on the parent block and the sibling blocks at the same scale.

Experiments were performed on a Pentium Pro 230-MHz PC with a LINUX operating system. For both the single-resolution HMM and the MHMM, computational complexity depends on many parameters including the number of states in a model and parameters that control the extent of approximation taken by the path-constrained Viterbi algorithm. Instead of comparing computational time directly, we compare classification performance given roughly equal computational time. The average CPU time to classify a 512×512 aerial image with the HMM described previously is 200 s. With a 2-D MHMM described in Table I (bottom right) and somewhat arbitrarily chosen parameters required by the path-constrained Viterbi algorithm, the average user CPU time to classify one image is 192 s, slightly less than that with the HMM. The average classification error rate is 17.32%, 8% lower than the error rate achieved with the HMM. By using the more sophisticated model given by Table I (top) and more computation, the error rate can be improved further to 16.02%. With the HMM, however, applying more sophisticated models and more computation does not yield considerable improvement in performance.

The average user CPU time to classify one aerial image is 0.2 s for Fast Algorithm 1 and 7.3 s for Fast Algorithm 2,

much faster than the previous algorithms based on HMMs and MHMMs. The computation time of Fast Algorithm 1 is very close to that of CART, which is 0.16 s on average. In all cases, the classification time provided here does not include the common feature computation time, which is a few seconds. In the case of Fast Algorithm 1, the feature computation is the primary computational cost.

VIII. Testing Models

Although good results are achieved by algorithms based on the HMMs and MHMMs, which intuitively justify the models, in this section we examine the validity of the models for images more formally by testing their goodness of fit. The main reason for proposing the models is to balance their accuracy and computational complexity; that they are absolutely correct is not really an issue. The purpose of testing is thus more for gaining insight into how improvements can be made rather than for arguing the literal truthfulness of the models.

A. Test of Normality

A 2-D HMM assumes that given its state, a feature vector is Gaussian-distributed. The parameters of the Gaussian distribution depend on the state. In order to test the normality of feature vectors in a particular state, the states of the entire data set are searched according to the MAP (maximum *a posteriori*) rule using an estimated model. Feature vectors in each state are then collected as data to verify the assumption of normality. The test was performed on the aerial image data set described in Section VII. The model used is a single-resolution hidden Markov model with five states for the natural class and nine states for the man-made class.

The test of normality is based on the well-known fact that a multivariate normal distribution with covariance proportional to the identity is uniform in direction. No matter the covariance, its projection onto any direction has a normal distribution. For a general Gaussian distribution, a translation followed by a linear transform can generate a random vector with unit spherical normal distribution, perhaps in dimension lower than that of the original data. This process is usually referred to as decorrelation, or whitening.

For each state of the model, the normality of feature vectors was tested. The data were decorrelated and then projected onto a variety of directions. The normal probability plot [4] for every projection was drawn. If a random variable follows the normal distribution, the plot should be roughly a straight line through the origin with unit slope. The projection directions include individual components of the vector, the average of all the components, differences between all pairs of components, and seven random directions. Since the feature vectors are eight-dimensional, 44 directions were tested in all. Limitations of space preclude showing all the plots here. Therefore, details are shown for one state which is representative of the others.

Fig. 10(a) is the normal probability plots for each of the eight components. Counted row-wise, the seventh and eighth plots in Fig. 10(a) show typical "bad" fit to the normal distribution for projections onto individual components, whereas the first plot is a typical "good" fit. "Bad" plots are characterized by data that are truncated below and with heavy upper tails. Most plots resemble the fourth to sixth plots in Fig. 10(a), which are slightly curved. Fig. 10(b) shows plots for the average of all the components (the first plot) and projections onto random directions. We see that the average and the projections onto the random directions fit better with the normal distribution than do the individual components. These are due to a "central limit effect" and are shown consistently by the other states. Fig. 11 presents the normal probability plots for differences between some pairs of components. Differences between components also tend to fit normality better than the individual components for all the states. A typical "good" fit is shown by the third and the seventh plots, which are aligned with the ideal straight line over a broad range. A typical "bad" fit is shown by the second and sixth plots, which only deviate slightly from the straight line. But here, "bad" means heavy lower tails and truncated upper tails.

B. Test of the Markovian Assumption

For 2-D HMMs, it is assumed that given the states of the two neighboring blocks (right to the left and above), the state of a block is conditionally independent of the other blocks in the "past." In particular, "past" means all the blocks above and to the left. In this section, we test three cases of conditional independence given the states of block (m, n) and $(m-1, n+1)$: the independence of

$$\{(m-1, n), (m, n+1)\}$$
$$\{(m, n-1), (m, n+1)\}$$

and

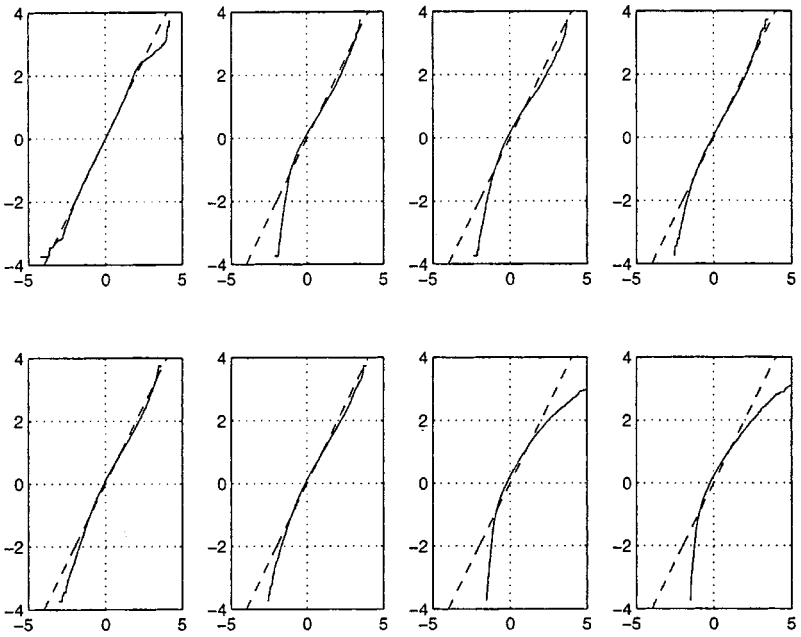
$$\{(m, n+1), (m-2, n+1)\}.$$

For notational simplicity, we refer to the three cases as Cases 1, 2, and 3, which are shown in Fig. 12.

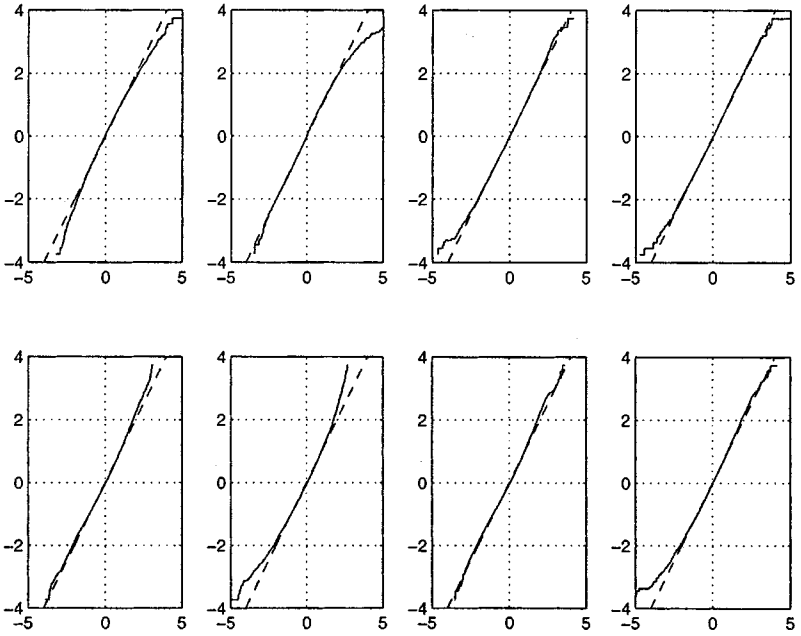
For 2-D MHMMs, two assumptions are tested. One is the Markovian assumption that given the state of a block at resolution r, the state of its parent block at resolution $r-1$ and the states of its child blocks at resolution $r+1$ are independent. A special case investigated is the conditional independence of block $(i_0, j_0) \in \mathbb{N}^{(r-1)}$ and one of its grandchild blocks (i_2, j_2), $i_2 = 4i_0$, $j_2 = 4j_0$, $(i_2, j_2) \in \mathbb{N}^{(r+1)}$. Fig. 13(a) shows the conditioned and tested blocks. The other assumption tested is that given the states of parent blocks at resolution r, the states of nonsibling blocks at resolution $r+1$ are independent. In particular, a worst possible case is discussed, that is, given the states of two adjacent blocks at resolution r, the states of their two adjacent child blocks are independent. The spatial relation of those blocks is shown in Fig. 13(b).

As with the test of normality in the previous section, the test of independence was performed on the aerial image data set. States were searched by the MAP rule. Then, the test was performed on those states. In the case of the HMM, the same model for the test of normality was used. For the MHMM, the three-resolution model described by Table I (bottom right) was used.

To test the conditional independence, for each fixed pair of conditional states, a permutation χ^2 test [23], [8] was applied. The idea of a permutation test dates back to Fisher's exact test



(a)



(b)

Fig. 10. Normal probability plots for one state. (a) Each component. (b) The average of all the components and projections onto random directions.

[15] of independence in a 2×2 contingency table [4]. In principle, Fisher's exact test can be generalized to testing independence of an $r \times c$ contingency table. The difficulty is with computational complexity, which seriously constrains the use of exact tests. Mehta and Patel [27] have taken a network approach to achieve computational feasibility. Boyett [8] proposed subsampling random permutations to reduce computation. The random permutation approach is taken here for simplicity.

Suppose the test for independence is for block $(m-1, n)$ and $(m, n+1)$ (Case 1 of the HMM). The entry $\alpha_{i,j}$ in the contingency table is the number of occurrences for $(m-1, n)$ being in state i and $(m, n+1)$ being in state j. Denote the marginal counts by

$$r_i = \sum_{j=1}^{M} \alpha_{i,j}, \quad c_j = \sum_{i=1}^{M} \alpha_{i,j}, \quad \text{and} \quad n = \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_{i,j}$$

where M is the total number of states. For each $\alpha_{i,j}$, generate $\alpha_{i,j}$ indices $\binom{i}{j}$. A list is generated by assembling indices $\binom{i}{j}$

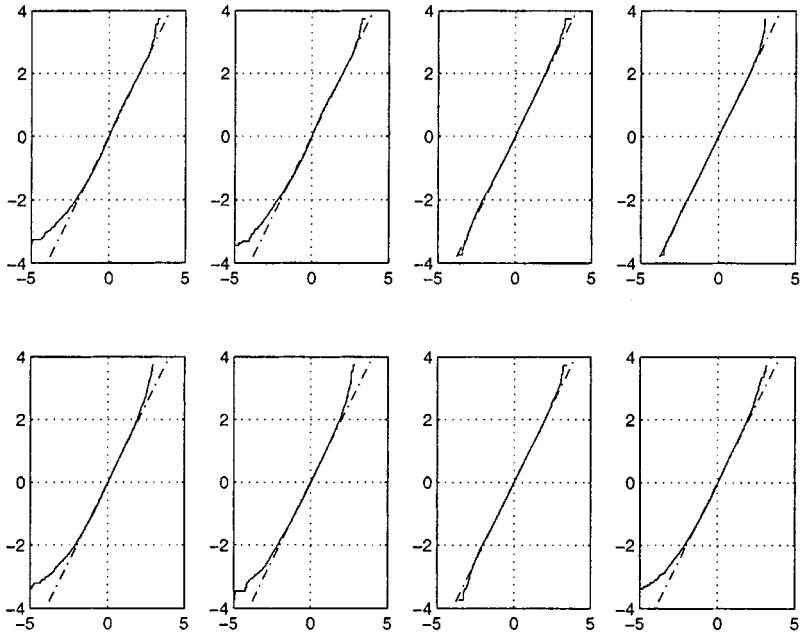


Fig. 11. Normal probability plots for one state: differences between pairs of components.

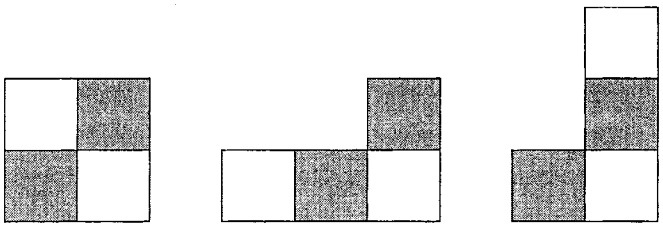
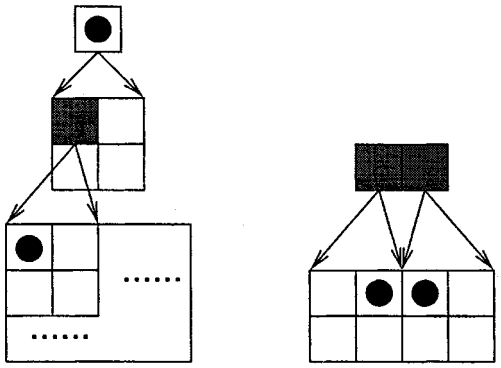


Fig. 12. Tests of conditional independence for the HMM. The states of gray blocks are conditional states; the states of white blocks are states upon which tests for independence were performed.



(a) (b)

Fig. 13. Tests of the MHMM assumptions. (a) Markovian property across resolutions. (b) Conditional independence of child blocks descended from different parent blocks. The states of gray blocks are conditional states; the states of dotted blocks are states upon which tests for independence were performed.

in a certain order. A permutation is obtained by randomly permuting the second number j while fixing the order of the first number i. For an example 2×2 contingency table

$$\begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$$

a list as follows is generated:

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 1 & 1 & 1 & 2 & 2 \end{pmatrix}.$$

Fixing the first row and permuting the second row might yield a list

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 1 & 2 & 1 & 1 & 1 \end{pmatrix}.$$

The permutation yields new counts for the number of $\binom{i}{j}$ in the list, denoted as $\hat{\alpha}_{i,j}$. Note that the marginal counts remain unaltered; that is,

$$r_i = \sum_{j=1}^{M} \hat{\alpha}_{i,j} \qquad c_j = \sum_{i=1}^{M} \hat{a}_{i,j}.$$

For the particular case of the above list, the new contingency table is

$$\begin{pmatrix} 1 & 2 \\ 4 & 1 \end{pmatrix}.$$

For both the original contingency table and those generated by random permutations, we compute Pearson's χ^2 statistic [4]

$$\chi^2 = n \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{(\alpha_{i,j} - r_i c_j/n)^2}{r_i c_j}. \tag{12}$$

The quantity $\alpha_{i,j}$ is replaced by $\hat{\alpha}_{i,j}$ for tables generated by permutations. Denote the χ^2 statistic of the original contingency table as χ^2_{obs}. The p-value for the original contingency table is

$$p = \frac{\text{number of contingency tables for which } \chi^2 \geq \chi^2_{\text{obs}}}{\text{number of permutations} + 1}.$$

The number of permutations used was 1000.

Since conditional independence is of concern, p-values were computed for each condition. The HMM in discussion has a total of 14 states, which yield 14×14 conditions, each corresponding to a pair of states for neighboring blocks above and to the left. We thus have 196 p-values for each case of the independence tests shown in Fig. 12. For Cases 1, 2, and 3, the medians of the p-values are 0.055, 0.462, and 0.443, respectively. The percentage of p-values above 0.05 for Cases 1, 2, and 3 is around 50%, 95%, and 90%, correspondingly. Results show that Cases 2 and 3 fit the conditional independence assumption about equally well, and much better than does Case 1. This coincides with our intuition. We expect that the conditional independence assumption is less true for Case 1 since the two blocks under examination touch at a corner.

To test the Markovian property across resolutions for the 2-D MHMM, p-values were computed for each of the six conditional states at Resolution 2. Among the six p-values, one is 0.89, another is 0.17, and all the other four are below 0.05, indicating strong dependence between Resolution 1 and 3. However, the Markovian property across resolutions is usually assumed to maintain computational tractability. For the testing of conditional independence of nonsibling blocks, there are $6 \times 6 = 36$ state pairs of parent blocks, each of which is a condition. The median of the 36 p-values was 0.44. About 70% of them were above 0.05. Therefore, for most conditions, there is no strong evidence for dependence between blocks descended from different parents.

IX. Conclusions

In this paper, a multiresolution 2-D hidden Markov model is proposed for image classification, which represents images by feature vectors at several resolutions. At any particular resolution, the feature vectors are statistically dependent through an underlying Markov mesh state process, similar to the assumptions of a 2-D HMM. The feature vectors are also statistically dependent across resolutions according to a hierarchical structure. The application to aerial images showed results superior to those of the algorithm based on single-resolution HMMs. As the hierarchical structure of the multiresolution model is naturally suited to progressive classification if we relax the MAP rule, suboptimal fast algorithms were developed by searching for states in a layered fashion instead of the joint optimization.

As classification performance depends on the extent to which 2-D multiresolution hidden Markov models apply, the model assumptions were tested. First, we tested, at least informally, the assumption that feature vectors are Gaussian-distributed given states. Normal probability plots show that the Gaussian assumption is quite accurate. Second, we tested the Markovian properties of states across resolutions and within a resolution. A permutation χ^2 test was used to test the conditional independence of states. The results do not strongly support the Markovian property across resolutions, but this assumption is usually needed to maintain computational tractability. At a fixed resolution, the bias of the Markovian property assumed by the HMM is primarily due to assuming the conditional independence of a state and its neighboring state at the left upper corner given the left and above neighboring states. Therefore, to improve a 2-D HMM, future work should include the left upper neighbor in the conditioned states of transition probabilities.

Acknowledgment

The authors acknowledge the helpful suggestions of the reviewers.

References

[1] K. Abend, T. J. Harley, and L. N. Kanal, "Classification of binary random patterns," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 538–544, Oct. 1965.

[2] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2743–2760, Oct. 1998.

[3] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 766–784, Mar. 1992.

[4] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[5] T. Binford, T. Levitt, and W. Mann, "Bayesian inference in model-based machine vision," *Uncertainty in Artificial Intelligence*, 1988.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. London, U.K.: Chapman & Hall, 1984.

[7] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Processing*, vol. 3, pp. 162–177, Mar. 1994.

[8] J. M. Boyett, "Random $R \times C$ tables with given row and column totals," *Appl. Statist.*, vol. 28, pp. 329–332, 1979.

[9] H. Choi and R. G. Baraniuk, "Image segmentation using wavelet-domain classification," in *Proc. SPIE Tech. Conf. Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, July 1999, pp. 306–320.

[10] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artif. Intell.*, vol. 42, pp. 393–405, 1990.

[11] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.

[12] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1–21, 1977.

[14] P. A. Devijver, "Probabilistic labeling in a hidden second order Markov mesh," in *Pattern Recognition in Practice II*. Amsterdam, The Netherlands: Elsevier, 1985, pp. 113–23.

[15] R. A. Fisher, *The Design of Experiments*. Edinburgh, Scotland: Oliver and Boyd, 1953.

[16] C. H. Fosgate, H. Krim, W. W. Irving, W. C. Karl, and A. S. Willsky, "Multiscale segmentation and anomaly enhancement of SAR imagery," *IEEE Trans. Image Processing*, vol. 6, pp. 7–20, Jan. 1997.

[17] W. T. Freeman and E. C. Pasztor, "Learning low-level vision," in *Proc. 7th Int. Conf. Computer Vision*, Corfu, Greece, Sept. 1999.

[18] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[19] D. Heckerman, "A tutorial on learning with Bayesian Networks," Microsoft Res., Tech. Rep. MSR-TR-95-06, Nov. 1996.

[20] D. Knill and W. Richards, Eds., *Perception as Bayesian Inference*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[21] T. Kohonen, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1989.

[22] M. S. Landy and J. A. Movshon, Eds., *Computational Models ofVisual Processing*. Cambridge, MA: MIT Press, 1991.

[23] L. C. Lazzeroni and K. Lange, "Markov chains for Monte Carlo tests of genetic equilibrium in multidimensional contingency tables," *Ann. Statist.*, vol. 25, pp. 138–168, 1997.

[24] E. Levin and R. Pieraccini, "Dynamic planar warping for optical character recognition," in *Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, San Francisco, CA, Mar. 1992, pp. 149–152.

[25] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two dimensional hidden Markov model," *IEEE Trans. Signal Processing*, vol. 48, pp. 517–33, Feb. 2000.

[26] G. Loum, P. Provent, J. Lemoine, and E. Petit, "A new method for texture classification based on wavelet transform," in *Proc. 3rd Int. Symp. Time-Frequency and Time-Scale Analysis*, June 1996, pp. 29–32.

[27] C. R. Mehta and N. R. Patel, "A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables," *J. Amer. Statist. Assoc.*, vol. 78, pp. 427–434, June 1983.

[28] Y. Meyer, *Wavelets Algorithms and Applications*. Philadelphia, PA: SIAM, 1993.

[29] R. D. Nowak, "Multiscale hidden Markov models for Bayesian image analysis," in *Bayesian Inference in Wavelet Based Models*. New York: Springer-Verlag, 1999, pp. 243–266.

[30] K. L. Oehler, "Image compression and classification using vector quantization," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1993.

[31] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 461–73, May 1995.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[33] P. Pérez and F. Heitz, "Restriction of a Markov random field on a graph and multiresolution statistical image modeling," *IEEE Trans. Inform. Theory*, vol. 42, pp. 180–90, Jan. 1996.

[34] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification," *IEEE Trans. Image Processing*, vol. 5, pp. 347–60, Feb. 1996.

[35] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948.

[36] M. Stone, "Cross-validation: A review," *Math. Operationforsch. Statist. Ser. Statist.*, pp. 127–39, 1978.

[37] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, pp. 1549–60, Nov. 1995.

[38] A. J. Viterbi and J. K. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 325–332, May 1974.

[39] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *HTK—Hidden Markov Model Toolkit*. Cambridge, U.K.: Cambridge Univ. Press, 1995.