Proceedings of the World Congress on Engineering 2008 Vol I
WCE 2008, July 2 - 4, 2008, London, U.K.

Hidden Markov Model Based Classification of Natural Objects in Aerial Pictures

Mohamed El Yazid Boudaren, Abdenour Labed, Adel Aziz Boulfekhar, Yacine Amara

Abstract—**In this paper, we propose a new probabilistic approach designed for supervised classification of natural objects (vegetal and mineral) in high resolution aerial pictures. It consists of a two layered hidden Markov models (HMM) based approach which takes into account the spatial correlations between natural objects. The performance of our approach has been tested on real world high resolution aerial pictures, and the obtained results demonstrated its effectiveness compared to those presented in the literature.**

Index Terms— **2D-HMM, Hidden Markov models HMM, Natural Objects Recognition.**

I. Introduction

This work is part of a more global one that consists in creating virtual environments from aerial pictures combined with altimetry data. In such environments, while getting too close to the ground, one has to solve the problem of limited textural resolution. So, these textures have to be amplified to get more realistic scenes and immerse the user in this virtual world. Amplification must take account of objects nature. For instance, grass and roads are not amplified in the same way. Hence, a classification of pixels in the picture must be performed in order to exploit efficiently these pictures in a virtual reality framework.

Classification of natural objects (mineral or vegetal) in aerial pictures can be seen as a missing data problem, since we need to assign each pixel to a missing (hidden) class of natural objects. In this paper, we consider a supervised classification and assume existence of spatial correlations between natural objects present in the area of interest.

II. Related Works

Premoze *et al*. [1] performed a classification of terrain in a mountainous region from grey level aerial pictures of a slightly lower resolution than those we worked on. They adopted a features vector of eight components to describe each pixel. First, they used data in the aerial pictures to estimate density distributions of the classes.

Manuscript received February 9, 2007.

Mohamed El Yazid Boudaren is with the Applied Mathematics Laboratory, EMP, Algiers, Algeria (phone: +213 551 147 274; email: boudarenyazid@hotmail.com).

Abdenour Labed is with the Applied Mathematics Laboratory, EMP, Algiers, Algeria (email: a-labed@lycos.com).

Adel Aziz Boulfekhar is with the Applied Mathematics Laboratory, EMP, Algiers, Algeria (email: boulfekhar.adel@hotmail.com).

Yacine Amara is with the Applied Mathematics Laboratory, EMP, Algiers, Algeria (email: amara.yacine@gmail.com).

Then, a Bayesian classifier is used for segmentation of pictures. But, their approach has some drawbacks, since it is based on a pixel piecewise classification. Indeed, spatial dependencies are considered only for immediate neighborhood, which is insufficient to recognize complex textures, so, broader neighborhoods are necessary. Besides, only dependencies within the same class are taken into account and natural objects are supposed to be spatially uncorrelated which is generally incorrect. This seriously affects the classification accuracy, especially when the segmentation concerns a unique region.

In this paper we propose a HMM based approach designed for supervised classification of natural objects in high resolution aerial images.

Hidden Markov Models (HMM) have long been used to efficiently model uni-dimensional data (sequences of symbols), in particular in speech recognition systems. In theory, HMMs can be also applied to multi-dimensional data. However, the complexity of the algorithms grows exponentially in higher dimensions, so that, even in dimension 2, the use of plain 2D-HMM becomes prohibitive in practice [2].

2D-HMM is defined in a similar way to 1D-HMM. The output observation is an array of symbols O_{xy} which are emitted in accordance with the current state q_{xy}. For instance, the pixels of an image scanned using a line by line ordering. In such model, the classical linear dependency is replaced by a double dependency which doesn't allow the factorization of computation as in 1D-HMM. This leads to an exponential increase in the amount of computation that is needed for the regular Baum-Welch and Viterbi algorithms. For this reason, the use of plain 2D-HMM is unaffordable in practice.

Many approaches have been proposed to overcome the complexity problem of 2D-HMMs [3]. One of the earliest versions of such approaches is described in [4] which uses a 1D-HMM to model horizontal bands of face images. A more elaborate idea consists in extracting 1D features out of the image or video, and model these features with one or more 1D models [5].

Another approach uses a two-level model, called Embedded HMM, where a first high level model contains super-states associated with a low level HMMs, which model the lines of the observed image [6]. The main disadvantage of these approaches is that they greatly reduce the vertical dependencies between states, as it is only achieved through a single super-state.

In this work, we propose an efficient model that avoids the exponential complexity of regular 2D-HMM while taking account of both horizontal and vertical dependencies within the aerial picture. Our model is two-layered: the higher layer comprises a unique HMM constituted of super states associated with one low level HMM each. This model differs from embedded HMM in the sense that it deals with pixel blocks instead of pixel lines as elementary symbols. Another difference is that our high level HMM whose states correspond to natural objects is ergodic. This allows us to model natural objects dependencies more accurately.

III. HMM MODELING

Given the fact that there are two kinds of interactions, correlations within the same object and correlations between objects, we considered a two layered probabilistic model. One lower layer constituted of as many HMMs as the number of object classes, to represent the local dependencies, and a higher one of a unique HMM, to represent the global ones.

The training of our model has been done in two steps: firstly, the low level HMMs are trained on unitextured pictures. Secondly the high level one is trained on multitextured pictures of the same region using the parameters of HMMs of the first step, according to Baum-Welch algorithm with slight modifications.

It is stated that there is no systematic recipe for adapting HMM to a specific application. Furthermore, to get satisfactory results, one has to operate accurately at each phase of the classification system designing: feature vector choice, modeling, learning and recognition.

A. Features Vector Choice

For a good modeling, one has to determine the texture and color features that are most relevant for the natural object analysis and learning. Computed at each pixel, these values are the only information the learning and classification algorithms have.

Although HMM constitute a very efficient tool of spatial correlation modeling, we included in the features vector a spatiality information.

After a set of tests, *HSV* is chosen for color space and the gradient norm of component *V* for spatiality information. Hence, our features vector is constituted of four components. That we denote k_1, k_2, k_3 and k_4.

HSV is a very appropriate choice for color space since its components *H*, *S* and *V* are uncorrelated, which is crucial as it will be explained in the next sections.

B. Lower Layer Modeling

To enable our classifier to recognize unitextured images (pixel blocks eventually), we considered as many low level HMMs as the number of texture classes. The total constitutes the lower layer of our global model.

In this section, the image is considered one dimensional. To extract the observation sequence, the image is scanned line by line, from left to right and right to left alternatively, to avoid providing the program with false dependencies between two successive pixels.

Let a low level HMM be $\lambda = (\pi, A, B)$. We denote the symbols set $Y=V^4$ with $V=\{0, 1\ldots255\}$ while the states set is denoted $S=\{s_1, s_2\ldots s_n\}$. At time *t* the system state is q_t and the observation is O_t.

Given the fact that each pixel is described by four components, which raises the number of possible symbols per state to 256^4, a problem arises if we achieve our modeling in a classical way. That is why we replaced *B* matrix by four matrixes B_1, B_2, B_3 and B_4. Each one is estimated separately, considering only one component each time. We assume that:

$$b_j(k_1,k_2,k_3,k_4)=b_1(j,k_1)\times b_2(j,k_2)\times b_3(j,k_3)\times b_4(j,k_4). \qquad (1)$$

This assumption does not considerably affect the approach validity since *H*, *S* and *V* components are supposed to be independent.

Each low level HMM is trained on a unitextured image (128 x 128 pixels). First of all, we used K-means algorithm to perform an unsupervised classification of the image pixels. This constitutes a pre-processing operation that aims to determine the appropriate states number of the corresponding low level HMM. Then, we achieved the learning process using Baum-Welch algorithm.

Identifying the texture class of a given unitextured image amounts to determining the low level HMM that maximizes the probability of generating the observation sequence of this image. Thus, to assign an image to a given class, we compute the probability of its generation from each low level HMM. The image is then assigned to the natural object class '*i*' for which the conditional probability $P\ (O/\lambda_i)$ is maximal. The computation of this probability involves *backward* and *forward* functions.

C. Higher Layer Modeling

This layer consists of a unique high level HMM which models dependencies between different natural objects (fig.1). To classify aerial pictures, this model cooperates with lower layer models.

First, let us enumerate the main elements of the high level HMM $\lambda = (\pi, A, B)$.

- Symbols are here blocks of (3x3), i.e. a symbol is a sequence of 9 pixels with four components each. At time *t* observation $O_t = (O_{t,1}, O_{t,2}\ldots O_{t,9}) \in Z = Y^9$ with $Y=V^4$.
- States: $S=\{S_1, S_2, S_3 \ldots S_N\}$, each state corresponds to a natural object. At instant *t*, the image state is denoted q_t. Pixels block are considered unitextured since each block is emitted by a state (textured object class).
- A $_{[N,\,N]}$ matrix: a_{ij} represents the probability that the system evolves from state S_i to state S_j. This corresponds also to the probability that the current pixels block belongs to object class 'i' given that the previous block belongs to class '*j*'.
- B matrix: $b_j(k)$ represent the probability of emitting symbol z_k (9 pixels block) given that the system current state is S_j. More explicitly, it is the probability of emitting the 9 pixels block from the low level HMM corresponding to the same natural object as state S_j. In our method, the lower layer of the model will intervene here to compute $b_j(k)$ instead of estimating matrix *B* as usually done.

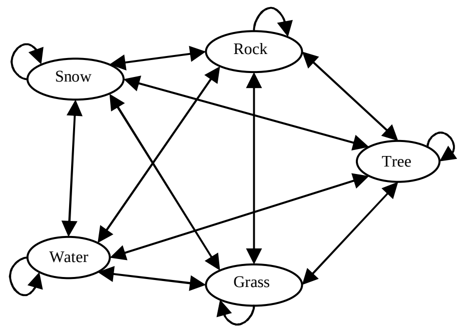

Fig. 1 – Sample of a five classes high level HMM structure

The observation sequence is extracted in a similar way to that of the lower layer. Instead of lonely pixels, each pixel is considered with its 8-neighborhood. Thus, to classify a pixel, we process it with its 8 direct neighbors, assuming each time that the block central pixel belongs to the same class as its neighbors. This will provide more information about texture, which can improve the recognition rate. The outcome of this assumption about neighborhood homogeneity is limited since we use overlapping blocks (fig. 2). A different treatment is devoted to the pixels of image boundary.

To estimate high level HMM parameters, we used Baum-Welch algorithm on which we performed slight modifications we performed in order to take into account the lower layer models.

In fact, B matrix is not estimated. To compute the value of $b_j(k)$, we resort to low level HMM corresponding to natural object 'j'.

Therefore,

$$b_j(k) = P(O_t = z_k / q_t = S_j)_{Higher} = P(O = z_k / \lambda_j)_{Lower} \qquad (2)$$

Unlike the lower layer HMM, where each symbol (pixel) was ascribed to a class separately using the evaluation of $P(O/\lambda_i)$ each instant, in the higher layer, the classification of an aerial picture pixels considers the whole picture. Indeed, the aerial picture is classified using *Viterbi* algorithm which determines the optimal states path that better explains the symbols (blocks) emission. Pixels are not attributed to classes till the whole image is processed. This enables our model to better perceive the picture. In fact, when analyzing local features in a small region taken lonely, it is sometimes difficult even for a human to tell what the image is about [7].

To better understand the classification process, let's consider fig. 2 where letters A, B…Y represent pixels. For the sake of simplicity, we only consider pixels: L, M and N.

First, the observation sequence $O = (O_1, O_2, O_3)$ is extracted. Note that O_1, O_2 and O_3 correspond to pixels L, M and N respectively (taken with their 8-neighbors each).

For each of O_1, O_2 and O_3, the observation sequence to be introduced to low level HMMs is obtained by scanning each local block column by column and not line by line. Therefore, we deal with vertical dependencies within the same block and horizontal ones between blocks.

Second, for each symbol O_l ($l=1, 2, 3$) and each state S_j we compute $b_j(O_l) = P(O_t=O_l/q_t=S_j)$ which is given by $P(O=O_l/\lambda_j)$ with λ_j being the HMM corresponding to natural object of state S_j (fig. 3).

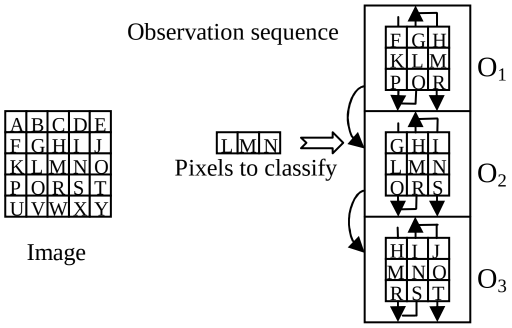

Fig. 2 – observation sequence extraction

Once these values calculated, the high level model has all the necessary parameters to find out the optimal states path. In fig. 2 case, decoding consists in determining $Q = (q_1, q_2, q_3)$ from S that maximizes the probability:

$P((O_{t1}, O_{t2}, O_{t3}) = (O_1, O_2, O_3) / (q_{t1}, q_{t2}, q_{t3}) = (q_1, q_2, q_3))$.

Finally, as each q_i corresponds to a natural object, finding out the optimal states path leads to identify pixels L, M and N.

Note that the computational complexity of this model remains low: we explore each block of pixels only once, for each block, we only have to consider all possible super states S_j (associated to λ_j) and compute $P(O_k/\lambda_j)$ for each. This only increases linearly (not exponentially) the model complexity.

D. Model Complexity

In this section, we provide an illustrative example to show that our algorithm exhibits only moderate computational complexity.

Let's consider an aerial image of length L and width W, let $T=L.W$ be the number of picture pixels. To classify these pixels, we start by extracting the observation sequence. It consists of T pixel blocks of 9 pixels each.

$O = O_1, O_2, \ldots O_k, \ldots O_T$.

With $O_k = (O_{k_1}, O_{k_2}, \ldots O_{k_9})$.

To assign each pixel of the picture to a class, we use the Viterbi algorithm applied to the observation sequence according to the high level HMM. Before we can apply Viterbi algorithm, we need to compute for each symbol (pixels block) O_k and each super state S_j (associated to low level HMM λ_j) the probability $P(O=O_k/\lambda_j)$. As this computation involves N low level HMMs of n states each applied to 9 pixels sequences, its complexity order is of: $(N.T.(9n^2))$. For our experimentation N=20 and n=3 for most low level HMMs.

The previous calculation provides us with the B matrix values necessary to the application of Viterbi algorithm. Now, we apply the decoding algorithm in a 1D-HMM context. This application has a complexity of $(N^2.T)$. This raises our model complexity to $[T(9Nn^2+N^2)]$. The complexity remains linear with the sequence length T.

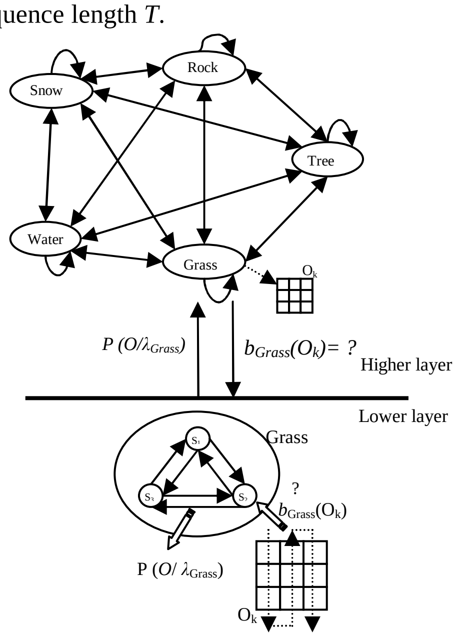

Fig. 3 – Cooperation between higher and lower layer

IV. Experimental Results

For our experiments, we used real world aerial pictures (like fig. 4) of a relatively large area, with a resolution of 50 centimeters.

For unitextured pictures, the correct classification rate obtained by low level HMMs on a set of 600 pictures (30 pictures of 32x32 pixels per object class) was of 93,17%.

To visually appreciate the multitextured pictures classification quality, we reconstituted some classified pictures according to class index of each pixel by associating a unique color with each class like in fig. 5.

Our results were then used to generate virtual interactive 3D-scene. This showed that our classifier was able to satisfactorily reproduce the original terrain.

However, the presence of shadow in several pictures limits the classification accuracy.

V. Conclusion

Designing our model with two layers to take account of dependencies between natural objects improves classification accuracy.

Although our model considers aerial picture one dimensional, it takes account of both pixels vertical dependencies through low level HMMs associated with natural object classes and natural objects horizontal dependencies through high level HMM. This constitutes a good tradeoff between classification accuracy and lower complexity of the model.

The use of overlapping blocks offers our classifier the opportunity to deal with sufficient amount of information without reducing the original aerial picture resolution.

To solve shadow problem, one can have recourse to firstly detect shadow pixels as a preliminary step, and treat them separately via a module devoted to this purpose.

Fig. 4 – Aerial picture sample (50cm per pixel)

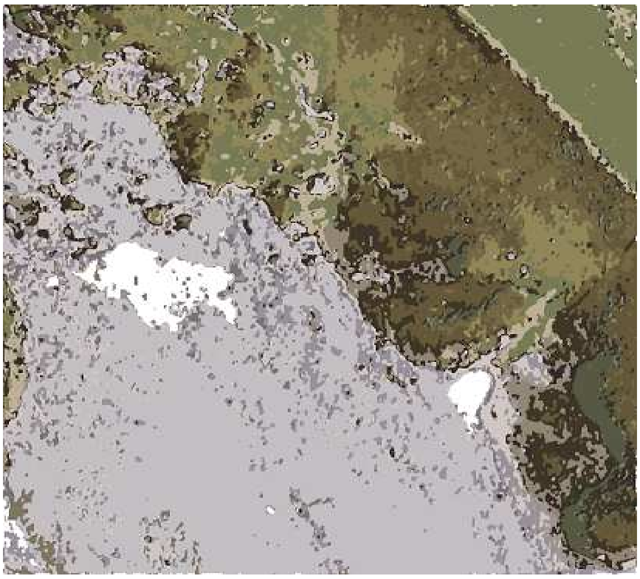

Fig. 5 – Reconstituted picture

References

[1] Simon Premoze, William B. Thompson and Peter Shirly: *Geospecific rendering of Alpine terrain*, Rendering Techniques'99, Proceedings of the Eurographics Workshop in Granada, 1999,Springer 1999, pp 107-118.

[2] Levin, E.; Pieraccini, R.; *Dynamic planar warping for optical character recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 3, 23-26 March 1992, pp.149 – 152.

[3] M. A. Mohamed and P. Gader, *Generalized hidden Markov models*-Part I: Theoretical Frameworks, IEEE Transaction on Fuzzy systems, February 2000, Vol.8, No.1, pp.67-81.

[4] Samaria, Ferdinando and Fallside, *Frank face identification and feature extraction using hidden Markov models*, Image Processing: Theory and Applications, Elsevier, 1993, pp 295-298.

[5] M. Brand, N. Oliver and A. Pentland, *Coupled hidden Markov models for complex action recognition*, In Proceedings, CVPR, pages 994--999. IEEE Press, 1997.

[6] S. Fine, Y. Singer, N. Tishby, *The hierarchical hidden Markov model: Analysis and applications*, Machine Learning 32(1998).

[7] Bernard Merialdo, Joakim Jiten, Eric Galmar, Benoit Huet, *A new approach to probabilistic image modeling with multidimensional hidden Markov models*, Adaptive multimedia retrieval2006, pp.95-107.