Chapter 40
A New Scheme for Land Cover Classification in Aerial Images: Combining Extended Dependency Tree-HMM and Unsupervised Segmentation

Mohamed El Yazid Boudaren and Abdel Belaïd

Abstract An important challenge to any image pixels classification system is to correctly assign each pixel to its proper class without blurring edges delimiting neighboring regions.

In this paper, we present an aerial image mapping approach that advantageously combines unsupervised segmentation with a supervised Markov model based recognition. The originality of the proposed system carries on three concepts: the introduction of an auto-adaptive circular-like window size while applying our stochastic classification to preserve region edges, the extension of the Dependency Tree–HMM to permit the computation of likelihood probability on windows of different shapes and sizes and a mechanism that checks the coherence of the indexing by integrating both segmentations results: from unsupervised over segmentation, regions are assigned to the predominating class with a focus on inner region pixels. To validate our approach, we achieved experiments on real world high resolution aerial images. The obtained results outperform those obtained by supervised classification alone.

Keywords Land cover classification · hidden Markov model · aerial images

40.1 Introduction

Land Cover Classification (LCC) in high resolution aerial images is an important application of remote sensed data. It consists of identifying the natural objects present in a high resolution aerial image given a set of known patterns. In the most general case of aerial images, when the image contains several regions of different patterns,

M.E.Y. Boudaren (✉)
Applied Maths Lab., Military Polytechnic School, P.O. Box 17, Algiers, 16111, Algeria
e-mail: boudaren@gmail.com

A. Belaïd
LORIA Lab., Read Team, P.O. Box 239, Vandoeuvre-lès-Nancy, 54506, France
e-mail: abelaid@loria.fr

S.-I. Ao and L. Gelman (eds.), *Electronic Engineering and Computing Technology*,
Lecture Notes in Electrical Engineering 60, DOI 10.1007/978-90-481-8776-8_40,

the aim is to label each pixel with the corresponding texture. Evidently, the labeling process subsumes image segmentation but besides segmenting the image to different regions, it assigns each region to one of the natural objects patterns.

Achieving the classification at pixel level is a big issue in LCC problem. In fact, it is easier to identify an image of a relatively big size than identifying a lonely pixel. In fact, pixel-wise approaches for image classification are not usually suitable to solve problems often found in remote sensing application [1, 2]. They result in a disgusting salt and pepper effect.

Recent researches clearly show the advantages of integrating spatial dimension to spectral features by using segmentation based classification methods and, hence, focusing into image regions instead of pixels [3, 4].

More elaborated approaches use a family of Markov models to model the contextual interactions between labels. However, genuine 2D-Markov modeling of the contextual information is a time consuming iterative process [5].

On the other hand, reasonable complexity approaches identify each pixel by taking into account its neighboring pixels, usually by computing a similarity measure (likelihood probability for instance) on square windows centered at concerned pixels [6]. The drawbacks of such approaches are the following:

- They adopt square windows which may introduce a bias toward rectangular regions. Moreover, corner pixels are more distant than other pixels. Adopting non-square windows is usually unaffordable due to the used model or measure nature.
- The bigger is the window, the more likely the identification is correct. However, adopting a too big window may penalize small regions. A tradeoff is generally made.
- Since a static window size is adopted, the window size is then too small to perform efficient classification for all image pixels and too high to preserve edges since the classification of frontier pixels are biased through introduction of neighborhood pixels.

In this paper, we propose a system that overcomes the previous difficulties by introducing the following:

- Segmentation is achieved through unsupervised segmentation which preserve region edges even if it provides an over-segmented image.
- Each region is identified through stochastic supervised classification.
- Likelihood probability may be computed on windows of different sizes and shapes centered at considered pixels.
- To determine window size and shape, an auto-adaptive distance is computed based on the considered pixel position towards region edges.
- To permit likelihood probability computation on non-rectangular windows, we extended the Dependency Tree-hidden Markov model (DT-HMM) by allowing four directional dependencies instead of two, and adopting the central pixel as root instead of upper-left pixel when dealing with rectangular windows.

The reminder of the paper is organized as follows: in Section 40.2, we introduce our Extended Dependency Tree-HMM (EDT-HMM) that extends DT-HMM. Section 40.3 describes our indexing scheme. Section 40.4 shows experimental results. Conclusion and future works are given in Section 40.5.

40.2 Extended Dependency Tree-Hidden Markov Models

Markov models (Markov Random Fields, Hidden Markov Fields, Hidden Markov Models, Hidden Markov Trees ...) were extensively and successfully used for texture modeling and segmentation [7]. This is majorly due to their ability to model contextual dependencies and noise absorption [8]. However, their performance depends widely on the model architecture: genuine 2D-models yield better results but exhibits much higher computational complexity [8]. In general, the more complex is the model, the better are the performances.

Nevertheless, for computational complexity reasons, several approaches consider linear models like HMM even if such a model is not suited for two-dimensional data [10]. More elaborated approaches resort to 2D-models with simplifying assumption. One simplifying assumption that provides good results with a linear complexity is that assumed in DT-HMM [11, 12]: one site (pixel) may depend on either the horizontal or vertical predecessor, but not on both the same time.

The extension of DT-HMM in this work is motivated by two reasons: the need to compute likelihood probability on non-rectangular shaped windows of different sizes and the need to adopt central pixel (to be labeled) as the dependency tree root since the root shows more interactions with neighbors than other pixels do.

40.2.1 EDT-HMM Overview

Before describing our model principles, let us define the applicability conditions of the EDT-HMM model on a window w with respect to root r.

The window w must fit the following condition:

- For each site s of w, s must have at least one neighbor $v \in N_s$ that belongs to w and fulfills: $\|v, r\| < \|s, r\|$ where N is the 4-neighborhood and $\| \ \|$ is the Euclidean distance.

Let w be a window verifying the condition above, and let r be the center of the window and $Y_r = \{y_s / s(i, j) \in w\}$ be the set of features vectors (RGB for instance) of pixels inside $w \cdot Y_r$ is then the observable process. Let X be the hidden process. The likelihood probability is given by:

$$P(Y_r/\lambda) = \sum_X P(Y_r/X, \lambda) P(X/\lambda) \tag{40.1}$$

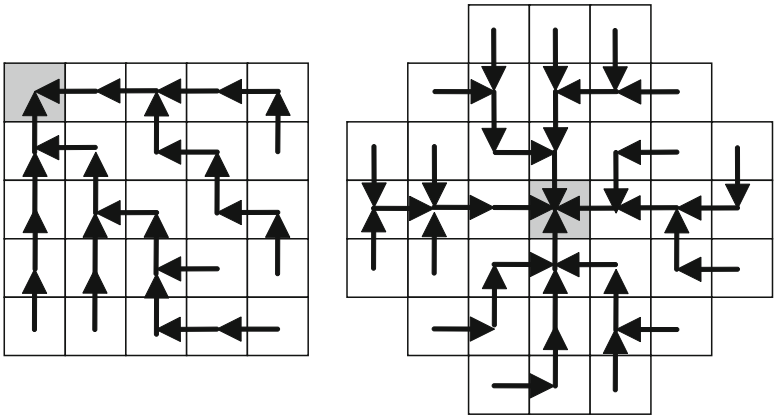

Fig. 40.1 Examples of random dependency trees according to DT-HMM (*left*) and EDT-HMM (*right*) formalisms

Unlike DT-HMM, where each pixel may have a predecessor chosen between two directions, in the EDT-HMM modeling, a pixel s may have a predecessor v chosen randomly from the 4-Neighborhood (up, down, right or left) and verifying the Euclidean distance property. Note that, the neighborhood directions of all pixels of w define a tree structure T like depicted in Fig. 40.1. We note $T(s) = v$.

The likelihood probability to observe Y_r given the parameters of the DT-HMM $\lambda(\pi, A, B)$ can be approximated as follows:

$$\begin{aligned} P(Y_r/\lambda) &\approx \sum_T P(Y_r/T,\lambda) \\ &\approx \sum_T \sum_X P(Y_r/X,\lambda) P(X/T,\lambda) \\ &\approx \sum_T \sum_X \left\{ \prod_{s\in w} P(y_s/x_s,\lambda) P(x_s/T,\lambda) \right\} \end{aligned} \tag{40.2}$$

In this paper, we propose to evaluate the likelihood on a set of random dependency trees τ. The previous equation becomes:

$$\begin{aligned} P(Y_r/\lambda) &\approx \sum_{T\in\tau} P(Y_r/T,\lambda) \\ &\approx \sum_{T\in\tau} \sum_X P(Y_r/X,\lambda) P(X/T,\lambda) \\ &\approx \sum_{T\in\tau} \sum_X \left\{ \prod_{s\in w} P(y_s/x_s,\lambda) P(x_s/T,\lambda) \right\} \end{aligned} \tag{40.3}$$

Thereafter, we remind the definition of the Model parameters π, A and B.

$$b_i(O) = P(y_s = O/x_s = i) \tag{40.4}$$

$$P(x_s = j/x_v = i, \lambda) = \begin{cases} \pi_j & if\ s = r \\ a_{ij} & otherwise \end{cases} \tag{40.5}$$

where $i, j = 1, ..N$ represent hidden states.

Note that a_{ij} only depends on i and j and not on the direction (horizontal or vertical).

To compute the likelihood probability of Eq. (40.3), we define the backward function $\beta_i\ (s)$ representing the probability of observing the data contained in the sub-tree of T with s as a root starting from the hidden state i.

$$\beta_i\ (s) = \begin{cases} b_i\ (y_s) & \text{if s is a leaf} \\ b_i\ (y_s) & \prod\limits_{T(v)=s} a_{ij}\beta_j\ (v)\ \text{otherwise} \end{cases} \tag{40.6}$$

Note that the likelihood probability of Eq. (40.3) can be evaluated as follows for each dependency tree T:

$$P(Y_r/T, \lambda) = \sum_{i=1}^{N} \pi_i \beta_i(r) \tag{40.7}$$

This computation exhibits a reasonable complexity (linear with window size).

The extension of the DT-HMM only concerns likelihood probability computation and Viterbi decoding whereas learning is performed the same way as in DT-HMM context.

The Viterbi decoding process can be achieved in a similar way to the likelihood probability computation. On the other hand, learning is performed via an iterative way the same as for the DT-HMM model, since the parameters are the same:

- Initialize model parameters
- Choose a random dependency tree T as described above (respecting the Euclidean distance constraint)
- Perform learning as in a linear framework (like in 1D-HMM)

40.3 Classification Scheme

To produce a class map of a given aerial image, we follow the scheme depicted in Fig. 40.2. In the following paragraphs, we describe each step.

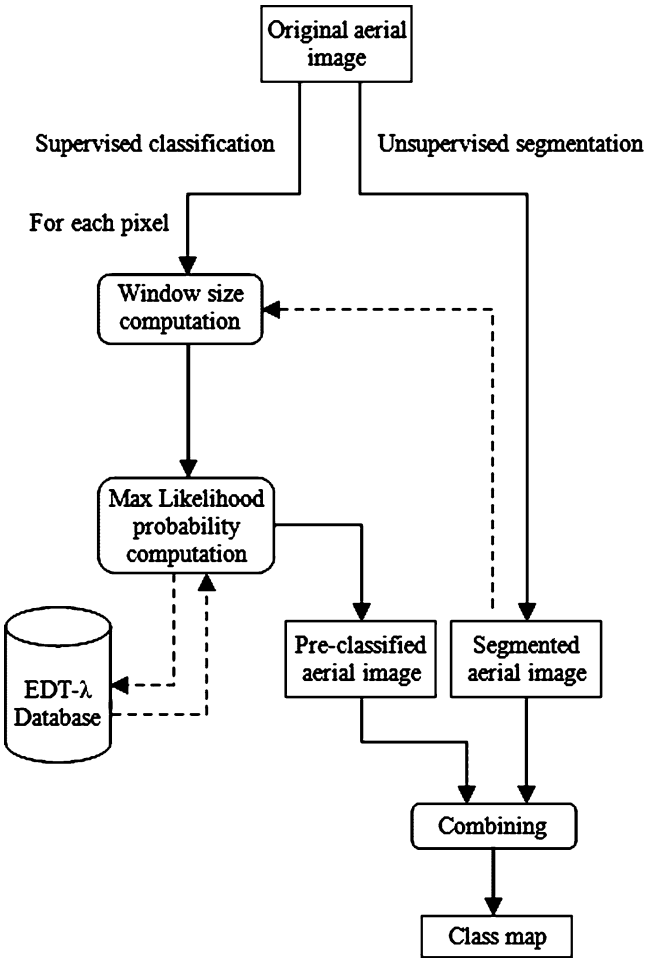

Fig. 40.2 Classification scheme

40.3.1 Image Unsupervised Segmentation

Before classifying the image pixels, we need to perform an image unsupervised segmentation that fits the following conditions:

- Image edges are preserved
- Pixels of the same region belong necessarily to the same natural object class, i.e. we may have an over-segmentation but not under-segmentation

This step serves as a pre-processing one, and will guide the rest of steps of the classification process.

Fig. 40.3 A sample of high resolution aerial image (RGD 73–74, 2008) (*left*) and its corresponding unsupervised segmentation (*right*)

One unsupervised segmentation that has been shown to provide good results is the one produced by the EDISON system [13, 14] which we use in this work. A sample of a high resolution aerial image (50 cm per pixel) and the corresponding unsupervised segmentation via the EDISON system are provided in Fig. 40.3.

40.3.2 Window Size Computation

Since texture is not a local phenomenon, in order to classify a pixel s, we consider it with its neighborhood. More explicitly, we will compute the likelihood of the data inside a window centered at the pixel under consideration.

Let us denote w_s such a window and Y_s the data associated to that window. The class λ_s of the central pixel s is the class that maximizes the likelihood probability:

$$\lambda_s{}^* = \arg\max_{\lambda \in \Lambda} \; P\left(Y_s/\lambda\right) \tag{40.8}$$

Most approaches adopt a square window of a fixed size for all pixels. A trade-off is usually made so that the window is enough big to correctly classify the central pixel and enough small to preserve the region edges.

In this work, we propose to dynamically compute the window size to allow our system to deal with a maximum amount of information without distorting region edges. The more the pixel to classify is far from the region boundary, the larger is the window whereas edge pixels are classified without considering their neighborhood.

Hence, window size is chosen so that pixels within the window belong to the same region according to an unsupervised over-segmentation of the image.

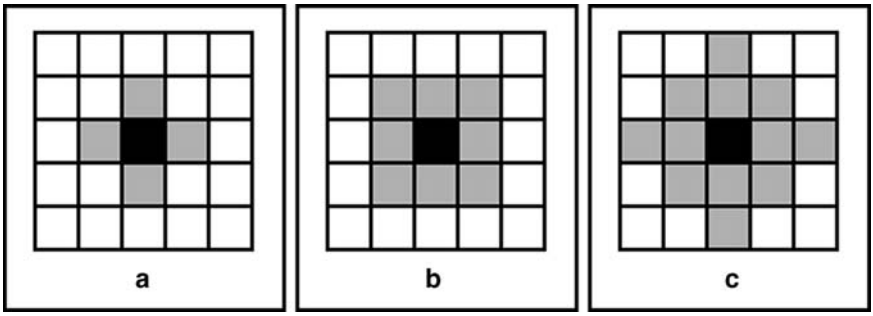

Fig. 40.4 Window shape and size for different values of Ray_s. a-$Ray_s{}^2 = 1$, b-$Ray_s{}^2 = 2$, c-$Ray_s{}^2 = 4$

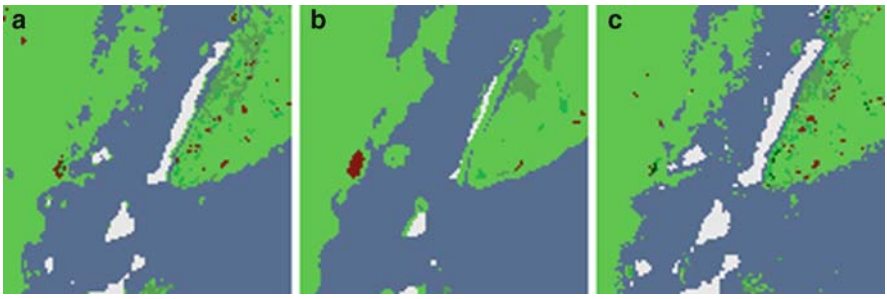

Fig. 40.5 Impact of window size on the pre-classification accuracy. a-$Ray_s{}^2 = 1$, b-$Ray_s{}^2 = 10$, c-auto-adaptative Ray_s

Window shape and size depend on a unique parameter Ray_s that represents the maximum Euclidean distance between neighbors and central pixel s. Figure 40.4 shows samples of windows of different shapes and sizes.

This parameter is obtained from the pre-segmented image. Its value is the maximum value possible so that pixels within the window belong to the same region. A comparative analysis of the window size impact on the accuracy of classification of the aerial image of Fig. 40.3 is shown in Fig. 40.5.

40.3.3 Image Pre-classification

To assign a pixel to a class, we compute the likelihood probability of observing the window centered on that pixel according to the EDT-HMM of each natural object class. The pixel is then allocated to the class that maximizes this probability (Fig. 40.5).

The parameters of the EDT-HMM corresponding to each natural object class are obtained after a learning process achieved on mono-class aerial images. To represent each pixel, we used the classical RGB color space. To estimate the parameters of the DT-HMM of each class, we achieve K-Means clustering on pixels of mono-class image of the corresponding class to divide the image pixels on N sub-classes. Subsequently, we obtain the parameters of N Gaussian functions. These parameters serve as an initialization of our EDT-HMM. The final parameters of the model are then obtained after an iterative process as described in the previous section.

40.3.4 Classification Correction

After the previous steps, the resulted class map suffers from the so called salt and pepper phenomenon. This is majorly to the difficulty to distinguish between several similar textures, especially for pixels near boundaries. To overcome this involvedness we propose to merge pixels of the same region (according to the unsupervised segmentation) into the same natural object class with a focus on inner pixels of the region, since those pixels were classified considering larger windows (Fig. 40.6). Explicitly, each region R is assigned the natural class that fits the following rule:

$$X_R{}^* = \arg\max_{X \in \Lambda} \sum_{\lambda_s = X, s \in R} size\ (w_s)$$

Fig. 40.6 Image classification correction Original aerial image (*left*), class map (*right*)

40.4 Experimentation

40.4.1 Data Overview

For our experimentation, we consider real world aerial images with a resolution of 50 cm per pixel (Fig. 40.6). The images were provided by La Régie de Gestion des Données des pays de Savoie, France ([9]).

The pictures were taken in relatively good light conditions; however, some images suffer from presence of shadow in some parts.

40.4.2 Learning Database

Learning was performed on mono-class images. These images were carefully extracted from the aerial images of the same area of study (Fig. 40.7).

40.4.3 Mono-class Images Generation

To demonstrate the capacity of the DT-HMM to represent natural object textures, we generate mono-class images using the corresponding DT-HMM and compare them to images generated by 1D-HMM and GMM (Fig. 40.8).

40.4.4 Experimental Results

To evaluate the robustness of our aerial images pixels' classification system, we considered three types of test images:

- Mono-class images, for which the classifier is expected to assign all pixels to the corresponding class (Fig. 40.9)

Fig. 40.7 Samples of learning images of classes: tree (*left*), snow (*middle*) and water (*right*)

Fig. 40.8 Mono-class image generation of class Tree using: 1D-HMM (*middle*) and DT-HMM (*right*)

Fig. 40.9 Mono-class image classification. Original aerial image (*left*), ground truth map (*middle*) and class map (*right*)

Fig. 40.10 Classification result on mosaic image. Original aerial image (*left*), ground truth map (*middle*) and class map (*right*)

- Mosaic images, assembled by combining different classes into regular boxes so that we can easily produce a corresponding ground truth map (Fig. 40.10)
- Natural aerial images, for which we don't have a precise ground truth map. Thus, only a visual evaluation can be achieved in this case (Fig. 40.6)

To produce the unsupervised segmentation of areal images, we acknowledge the use of EDISON system software [13, 14]. The obtained classification of the mosaic

image is very similar to the corresponding ground truth. Notice that the shadow class is not included in the ground truth map since we know to which class the shadowy pixels belong.

40.5 Conclusion

In this paper, we proposed an approach that advantageously combines supervised EDT-HMM modeling and unsupervised segmentation to classify land cover pixels. Instead of achieving our classification using a static window size, we resorted to an auto-adaptive window size depending on the position of pixel under consideration towards region boundaries.

Overall, the experimental results show that our system produces satisfactory class maps in a reasonable time given the linear complexity of the modeling. Note that several textures are so similar to each other that it is sometimes very difficult even for a human to distinguish between them.

As future work, we propose to apply EDT-HMM modeling to other kinds of problems.

References

1. Mesev, V.: Remotely Sensed Cities. Taylor & Francis, London (2003)
2. Thomas, N.C., Congalton, R.: A comparison of urban mapping methods using high-resolution digital imagery. Photogrammet. Eng. Remote Sens. **69**(9), 963–972 (2003)
3. De Jong, S.M., Freek, D.M.: Remote Sensing ImageAnalysis: Including the Spatial Domain. Springer, Berlin (2006)
4. Jensen, J.: Introductory Digital Image Processing. Prentice-Hall, New York (2006)
5. Levin, E., Pieraccini, R.: Dynamic planar warping for optical character recognition. IEEE International Conference on Acoustics, Speech and Signal Processing **3**, 149–152 (1992)
6. Permuter, H., Francos, J., Jermyn, I.H.: Gaussian mixture models for texture and colour for image database retrieval. IEEE International Conference on Acoustics, Speech and Signal Processing **3**, 569–572 (2003)
7. Noda, H., Mahdad Shirazi, N., Kawaguchi, E.: MRF based texture segmentation using wavelet decomposed images. Pattern Recog. **35**, 771–782 (2002)
8. Pieczynski, W.: Markov models in image processing. Traitement de Signal **20**(3), 255–277 (2003)
9. RGD73-74, 2008, Régie de Gestion des Données des Deux Savoies. http://www.rgd73-74.fr
10. Boudaren, M.Y., Labed, A., Boulfekhar, A.A., Amara, Y.: Hidden Markov model based classification of natural objects in aerial pictures. IAENG International Conference on Signal and Image Engineering, pp. 718–721, London, 2–4 July 2008
11. Merialdo, B.: Dependency Tree Hidden Markov Models. Research Report RR-05-128, Institut Eurecom (2005)
12. Merialdo, B., Jiten, J., Galmar, E., Huet, B.: A new approach to probabilistic image modeling with multidimensional hidden Markov models. Adap. Multimed. Retriev. 95–107 (2006)
13. Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
14. Meer, P., Georgescu, B.: Edge detection with embedded confidence. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1351–1365 (2001)