Advances in Environmental Research 5 (2001) 435–444

Advances in Environmental Research

www.elsevier.com/locate/aer

The automated extraction of environmentally relevant features from digital imagery using Bayesian multi-resolution analysis

Chris Pal[a,b,*], Dave Swayne[b], Brendan Frey[a]

[a] *Department of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1*
[b] *Computing Research Laboratory for the Environment, University of Guelph, Guelph, ON, Canada N1G 2W1*

Accepted 2 January 2001

Abstract

In this paper, we discuss the use of hierarchical tree-structured Bayesian networks for integrating knowledge concerning contextual relationships between environmentally relevant features extracted from digital imagery at multiple resolution scales. In our model, conditional probability distributions over continuous valued observations are parameterized using a mixture of multivariate Gaussian distributions. Separate classifiers for pixels and groups of pixels are used as sub-components of the overall model. The Bayesian formalism allows models to be composed in a systematic and statistically sound manner. We illustrate how this approach can be used to resolve ambiguity leading to classification errors and thus improve techniques for the classification of land use from aerial imagery. We present an example relevant to ecosystem analysis, the monitoring of urban growth and the automatic generation of input parameters for hydrologic models.

Keywords: Image feature extraction; Image analysis; Image segmentation; Knowledge integration; Bayesian networks; Computer vision; Ecosystem analysis; Monitoring urban growth; Hydrologic modeling

1. Introduction

For applications involving large areas of land, the automated pixel level classification of digital imagery is desirable if not essential. Additionally, in many cases, it is desirable to represent homogeneous areas in vector format. However, ambiguities exist when only pixel level information is used for the classification of a given pixel and this often leads to misclassification errors that humans never make and a ‘salt and pepper’ effect in areas that should be homogeneously classified. Additionally, if a sub-goal for the underlying application involves constructing a vectorized form of the map, ‘salt and pepper’ classification errors hinder this process. To overcome this problem, it is possible to integrate prior knowledge concerning features extracted at different scales of resolution to resolve ambiguities. In the following sections, we describe how Bayesian networks provide a statistically principled methodology for integrating knowledge concerning features extracted over multiple levels of resolution.

In this paper, we present the extraction of features for subsequent ecosystem analysis involving the moni-

* Corresponding author. Tel.: +1-519-888-4567; fax: +1-519-885-1208.
E-mail address: cjpal@uwaterloo.ca (C. Pal).

1093-0191/01/$ - see front matter
PII: S1093-0191(01)00095-8

toring of urban growth and hydrologic modeling as sub-tasks. When analyzing remotely sensed imagery of the land to extract environmentally relevant features, often one wishes to accurately classify and segment the image into homogeneous areas, suitable for storage in vector format. For ecosystem analysis, important features or ecostructures (American Forests, 1998) include the relative spatial arrangement of different types of trees, grasslands, roads, and house. Such data can be obtained through field measurements; however, the interpretation of high-resolution aerial photography provides an alternative to such surveys. As such, specialized software tools are needed to extract features from such imagery.

The task of high-resolution ecosystem analysis can therefore involve the following steps:

1. Selecting sample sites;
2. Determining and extracting ecostructures;
3. Performing analysis determining economic benefits; and
4. Extrapolating, if necessary to estimate city or regional conditions.

Importantly, step 3 may involve the estimation of the following factors and associated economic impacts (American Forests, 1998):

- Carbon storage and sequestration;
- Pollution removal of ozone, sulfur dioxide, nitrogen dioxide, and carbon monoxide;
- Storm water runoff reduction;
- Summer energy conservation;
- Urban wildlife habitat; and
- Urban heat islands (impacting residential energy savings)

Urban ecosystem analysis thus involves the monitoring of urban growth and hydrologic modeling as potential sub-tasks within the overall analysis. For the sub-task of monitoring urban growth, the identification of dwellings and roads in a given area over a period of time is of key interest. For hydrologic modeling, features such as grasslands, trees, houses and roads are important for determining parameters that describe the influence of land use on infiltration and runoff (e.g. SWMM: Curtis and Huber, 1993; Huber and Dickinson, 1988, a Storm Water Management Model). These parameters describe water pervious and impervious regions.

2. Multi-resolution analysis

There are many techniques for analyzing images that can be characterized as multi-resolution analysis. For example, with respect to mathematical analysis, relatively simple wavelet decompositions (Press et al., 1992) are often thought of as multi-resolution analysis techniques. In contrast, for the task of 'hand' classifying imagery, people commonly analyze an image at different scales of resolution as a result of looking at the details of an image or 'stepping back' to examine more of the image. Additionally, depending on the resolution of the image and the specific application, different features may be of interest (e.g. deciduous forest vs. oak tree or highway vs. drainage ditch). In the context of ecosystem analysis, the ecostructures defined at differing resolutions are often different. Depending on the goals of the analysis, resolution levels may be analyzed separately or information across scales may be integrated for assessing ecosystem health (Johnson and Ganapati, 1988).

Here, we focus on the task of automatically extracting environmentally relevant features into a classification image, through analyzing imagery at different spatial scales. Fig. 1 illustrates an example of some aerial imagery of an urban area and a classification of the image for coarser scale classes. This imagery has an original resolution of 1 m per pixel. Fig. 2 illustrates an example of a pixel level classification for an area near the lower right of Fig. 1. Tables 1 and 2 indicate the conceptual classes used at these two different respective levels of resolution or scale.

Most importantly, there is often uncertainty involved with the detection of features at each resolution level. Many techniques have been proposed to manage uncertainty within a relatively large software system such as fuzzy logic (Mendel, 1995) and the Dempster–Shafer theory (Shafer, 1976). However, Bayesian networks provide an intuitive, statistically principled and sound formalism (Heckerman, 1995).

An image model that has been used in the past in the remote sensing community is the multi-resolution

Table 1
Conceptual classes used in the coarse scale classification of Fig. 1

Color	Coarse resolution classes
	Sidewalk area
	Industrial area
	Dense trees
	Road
	Sparse trees
	Parking lot
	Open grass field
	Residential yard
	House roof

Table 2
Conceptual classes used in the fine scale classification in Fig. 2

Fine resolution classes	
Unknown deep shadow	Deciduous tree, grass and shadow
Shadowed grass	Deciduous tree and shadow
Burnt grass	Deciduous tree and grass
Green grass	Deciduous tree and pavement
Sidewalk and grass	Deciduous tree
Sidewalk, grass and tree	Car boundary
Sidewalk concrete	Car materials
Shadowed pavement	Industrial roof boundary
Pavement	Industrial roofing
Grass, pavement and concrete	Residential roof boundary
Gravel shoulder	Residential roofing
Pavement and yellow line	Coniferous tree and shadow
Pavement red tree grass	Coniferous tree
Deciduous tree, sidewalk and shadow	Unknown
Deciduous tree and sidewalk	

tree (Chou et al., 1994). There are many variations of this model found in the literature. With this model, the image is broken up into resolution levels. At each level of resolution is associated a random variable characterizing the content of the image at that resolution level. Starting with a single variable for the whole image, successive levels of resolution in the image are analyzed as a grid of slightly higher resolution variables. The states of variables at successive resolution levels are related to the variable representing the portion of the grid at the coarser resolution level containing the finer scale variable. At the coarsest level, variables represent classes describing the overall image, while at the lowest level, variables represent pixels.

Such a model can be represented succinctly using a graphical formalism for illustrating Bayesian networks in which variables (representing discrete classification states or vector valued observations) are illustrated as nodes in a graph and the relationships between variables are illustrated with arrows between the nodes. For nodes with no arrows pointing to the node, an unconditional probability table is associated with that node. For nodes with arrows coming in, a conditional probability table of the current node given its parents is associated. Fig. 3 illustrates a multi-resolution tree model as a Bayesian network, using this notation.

In the model illustrated in Fig. 3, the white nodes are ‘hidden’ unobserved variables corresponding to ‘higher level’ conceptual classes, while the gray nodes correspond to observed values (i.e. pixel values). The conditional probability distributions over the observed variables are often specified using parameterized functions such as the Gaussian distribution. Such tree-structured graphs are particularly attractive from a computational point of view in that fast algorithms based on probabilistic message passing or probability propagation (Pearl, 1988) between variables in the graph can be used to update the probabilities of unobserved variables based on values given for the observed variables. Once conditional and unconditional variables in a graph have been determined, an initial propagation of information contained in the unconditional and conditional probability tables among the variables in the graph is performed to find the initial probabilities for all the variables in the graph. The same message-passing or probability propagation algorithm can be used to update the states of the variables in the graph when new information becomes available (e.g. to classify image features given a new image).

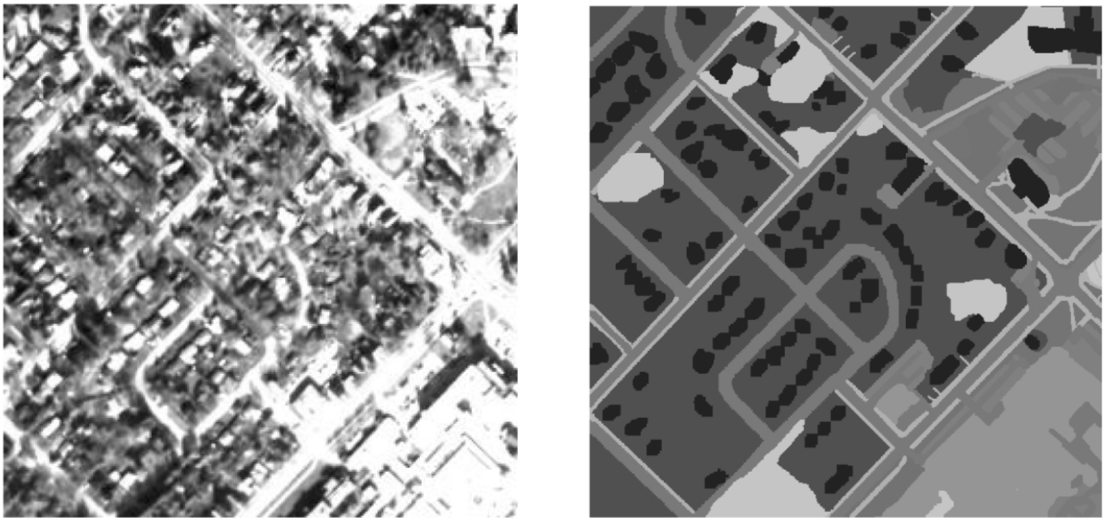

Fig. 1. (Left) Aerial imagery of an urban area. (Right) A potential segmentation.

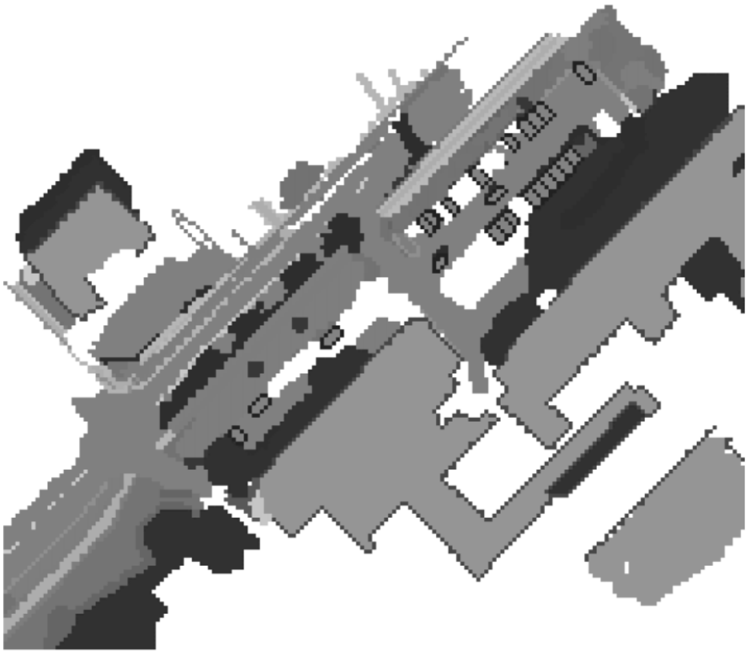

Fig. 2. A pixel level classification of a small portion of Fig. 1.

Using this formalism of a large tree-structured Bayesian network, the task of 'learning' the probability distributions over the states of these 'hidden' variables can be stated in terms of a statistical estimation problem. The algorithm for belief propagation discussed earlier is described by (Pearl, 1988) in terms of inference in probabilistic expert systems. It is only relatively recently that it has become fairly well known that this algorithm is equivalent to many other stochastic computations described in the literature (e.g. the forwards–backwards algorithm; Frey, 1998). Importantly, viewing such statistical computations in terms of a message-passing procedure within graphs facilitates the development of models with alternate structures.

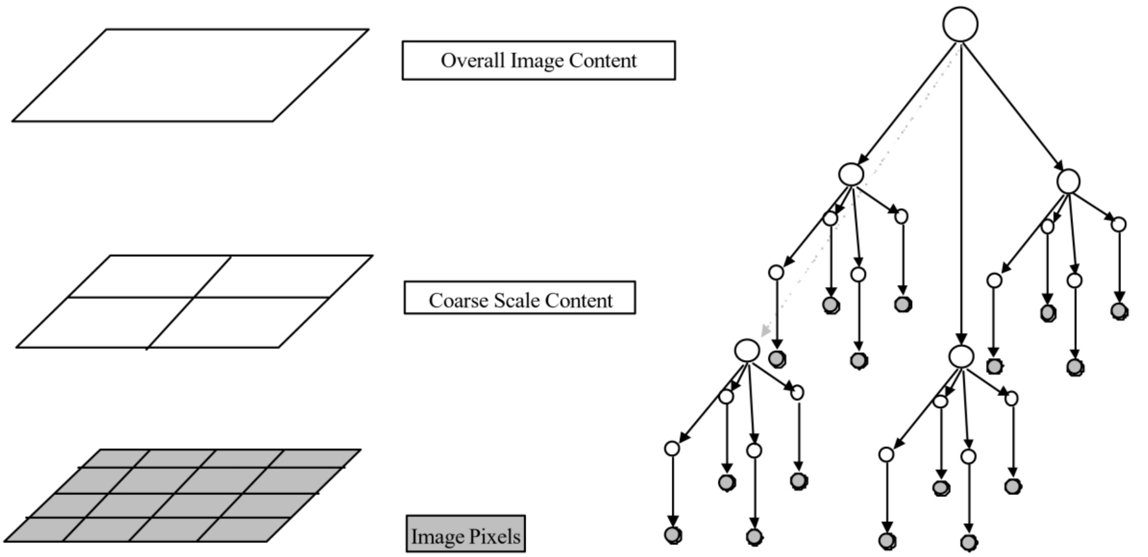

Fig. 3. An illustration of a commonly used multiscale tree model as a Bayesian network. Observed variables are indicated in gray and unobserved variables are indicated in white.

3. Methodology

3.1. Overview

In our approach, we construct a tree-structured probability model. However, in contrast to the model described previously, we introduce observed variables at each resolution level. These observations are obtained by gathering additional labeled data for classes associated with each of the resolution levels. Furthermore, we construct probability distributions over these observed variables by fitting Gaussian mixtures over the labeled feature classes. Introducing hidden sub-class variables into the graphical model then allows the mixtures to be combined. The task of building up this model from sub-components is described in further detail in the following sub-sections. The complete model is illustrated in Fig. 4 using the graphical formalism defined earlier.

3.2. Mixture models

One way to represent an unconditional probability distribution for continuous data is in terms of a mixture model (Bishop, 1995). In this context, a model is constructed and parameterized to estimate a probability density. Thus, when the mixture model takes on an appropriate functional form, certain parameters of the model can be interpreted as higher-level class variables in a probability diagram. Consider that we could construct a density function as a linear combination of basis functions where the number m, of basis functions is much less than the number of data points. Data points $\vec{\mathbf{x}}$ could consist of pixel level spectral measurements or grayscale values from a patch of pixels in an image. One can write a model for the density of a given class as a linear combination of a set of m independent and parameterized, component or 'class' densities $(\vec{\mathbf{x}}\,|k)$ such that $\vec{\mathbf{x}} \in R^n$, where n is the number of dimensions. Such a model takes the following form of Eq. (1).

$$P(\vec{\mathbf{x}}) = \sum_{k=1}^{m} P(\vec{\mathbf{x}}|k)P(k) \tag{1}$$

The probabilities $P(k)$ can be called prior probabilities of a data point having been generated from class k of the mixture; they are often called the mixing parameters and in the literature they are commonly denoted by π_k. The corresponding Bayesian network can be written such that there is a single categorical variable K taking on states $k \in [1,2,\ldots,m]$, where m is the number of classes. This variable causes a distribution over the continuous variable $\vec{\mathbf{x}}$. Data can be generated from such a model by selecting a point at random with probability $P(k)$ and then generating a data point from the corresponding class density $P(\vec{\mathbf{x}}\,|k)$. If we wish to find the probability of a class, then we can use Bayes' theorem:

$$P(k|\vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{x}}|k)P(k)}{P(\vec{\mathbf{x}})} = \frac{P(\vec{\mathbf{x}}|k)P(k)}{\sum_{k=1}^{m} P(\vec{\mathbf{x}}|k)P(k)} \tag{2}$$

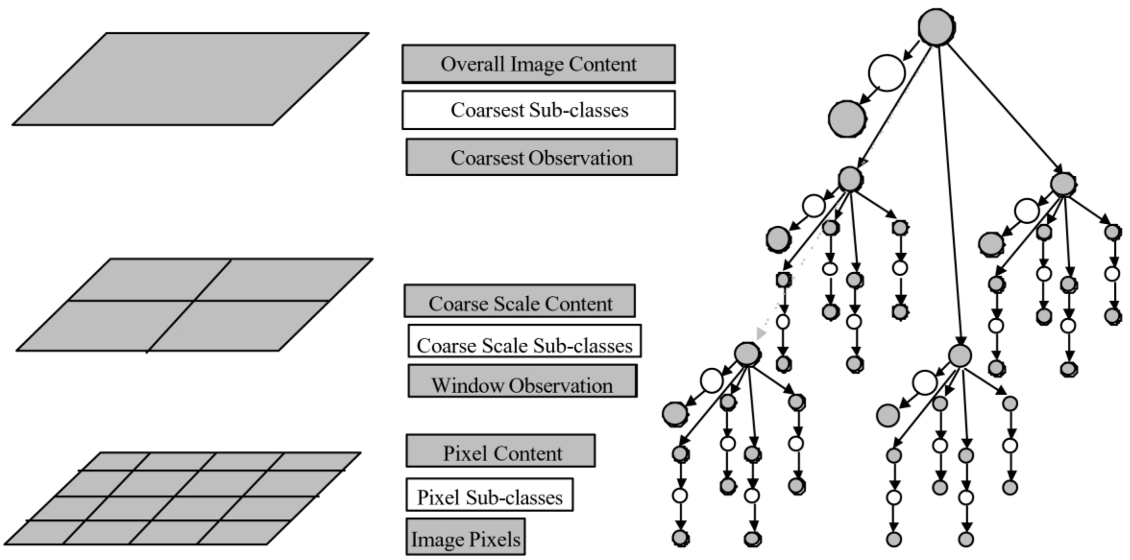

Fig. 4. The Bayesian network structure used in this investigation. Observed variables are illustrated in gray, while unobserved variables are illustrated in white.

This represents the posterior probability that a particular class k was responsible for generating the data point $\vec{\mathbf{x}}$.

3.3. A mixture model with classes and subclasses

The simple mixture model developed so far can be expanded slightly to represent an even more useful model. Consider a scheme in which $P(\vec{\mathbf{x}})$ is modeled by using a 'higher-level' class variable C taking on states $c \in [1,2,\ldots,n]$, where n is the number of states or 'classes'. In this model, with each class is associated a set of 'sub-classes' encoded as a set of states K_c of the larger sub-class variable K. In this model, the union of all the states of each of the sets K_c produces the states of the variable K (i.e. $\bigcup_c K_c = K$). K_c thus takes on m_c states $k_c \in [1,2,\ldots,m_c]$ for each class. The total number of states in K is thus $m = \sum_{c \in C} m_c$ states. The probability $P(k\,|\,c)$ for a given state $k \in K$ is thus zero if $k \notin K_c$ and all the states for which $k \in K_c$ are normalized for each among themselves for each c. Thus, $P(K|C) = \pi^c_{k'}$ can be written as a sparse matrix, where k' represents the subclass indices in the original mixtures. The sparse matrix takes the form of Eq. (3), with zero elements of the matrix left as blank:

$$P(K|C) = \begin{bmatrix} \pi^1_1 \;\; \pi^1_2 \ldots \pi^1_{m_1} & & & \\ & \pi^2_1 \;\; \pi^2_2 \ldots \pi^2_{m_2} & & \\ & & \ddots & \\ & & & \pi^n_1 \;\; \pi^n_2 \ldots \pi^n_{mn} \end{bmatrix} \tag{3}$$

Here, the rows of the matrix represent $P(K|C)$ for each class c of C. The columns represent each subclass state k of K. Thus, the unconditional probability, $P(\vec{\mathbf{x}})$ can be calculated by exploiting the structure of the sparse representation in the following way:

$$P(\vec{\mathbf{x}}) = \sum_{c \in C} \sum_{k_c \in K_c} P(\vec{\mathbf{x}}|k_c)P(k_c|c)P(c) \tag{4}$$

If $P(C)$ is constructed appropriately (i.e. normalized), it is easy to see that the unconditional probability is still normalized in this expanded model, i.e.

$$\int P(\vec{\mathbf{x}})d\vec{\mathbf{x}} = \sum_{c \in C} P(c) \sum_{k_c \in K_c} P(k_c|c) \int P(\vec{\mathbf{x}}|k_c)d\vec{\mathbf{x}} = 1 \tag{5}$$

One now has a probabilistic model in which there is a class variable that causes a distribution over a subclass variable that, in turn, causes a (parameterized) distribution over the continuous variable $\vec{\mathbf{x}}$. This probability model is illustrated in Fig. 5 as a Bayesian network and a factor graph (Frey, 1998) in which the associated probability tables and random variables are made explicit as rectangular nodes in the graph.

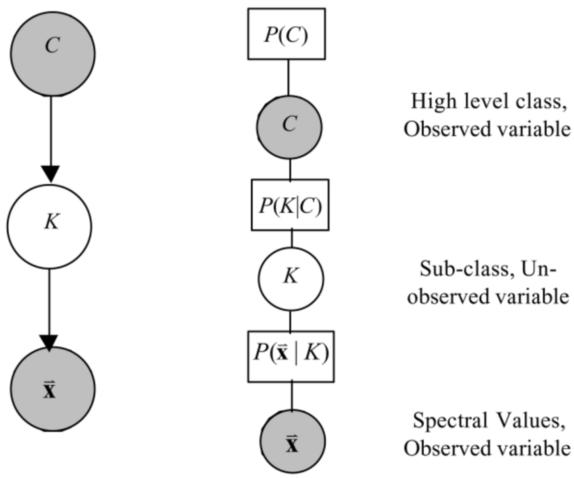

Fig. 5. Left: A class–subclass mixture model for class C causing a set of Gaussian sub-classes, indexed by K over vector space $\vec{\mathbf{x}}$. Right: The equivalent factor graph.

Consider now a calculation for the posterior probability of a particular state of the class variable given an observation of $\vec{\mathbf{x}}$, $P(c^* \in C|\vec{\mathbf{x}} = \vec{\mathbf{x}}_{obs})$. Using Bayes' rule, the calculation is simple:

$$P(c^*|\vec{\mathbf{x}}_{obs}) = \frac{P(\vec{\mathbf{x}}_{obs}|c^*)P(c^*)}{P(\vec{\mathbf{x}}_{obs})}$$

$$P(c^*|\vec{\mathbf{x}}_{obs}) = \frac{\sum_{k_{c^*} \in K_{c^*}} P(\vec{\mathbf{x}}_{obs}|k_{c^*})P(k_{c^*}|c^*)P(c^*)}{\sum_{c \in C} \sum_{k_c \in K_c} P(\vec{\mathbf{x}}_{obs}|k_c)P(k_c|c)P(c)} \tag{6}$$

The probability model of Fig. 5 is a simple, but a powerful building block for constructing complicated probability distributions over continuous variables with discrete parents using a relatively few number of parameters. Furthermore, models with this structure can be used as components to construct the leaves of the tree illustrated in Fig. 4.

3.4. 'Learning' or fitting mixture models from data

Mixture models can be 'learned' using the common statistical procedure of maximizing the likelihood of a data set consisting of values for the $\vec{\mathbf{x}}$ vector and the associated class labels. Practically every labeled data set can be 'clustered' or modeled separately and then combined as described above. If the subclass variables are given, then the parameters for Gaussian probability distributions $P(\vec{\mathbf{x}}\,|\,k)$ are found simply by calculating the mean and variance (or for higher dimensional vectors, the variance–covariance matrix) for each subclass.

When the subclass variables are not given but the number of states m_c for each K_c are given, then the Expectation Maximization algorithm or EM algorithm (Dempster et al., 1977) can be used. Using this procedure, one essentially 'fills in' the missing values with their expected value and iteratively fits the model with these 'hidden' variables. The algorithm will find a local maximum in the Maximum Likelihood solution space (Bishop, 1995).

When the number of states for each K_c is not given, one way to determine the appropriate number of states is to introduce a penalty term for the complexity of the model into the maximum likelihood calculation. This term is often constructed by measuring the number of bits used to encode the floating-point coefficients of the associated Gaussian probability distribution for a single subclass variable. Finding a model is then accomplished by searching through the penalized likelihood function by either adding additional Gaussians or starting with a large number of Gaussians and successively combining classes. This second approach is taken in (Bouman, 1995) and is also used for Gaussian mixture clustering for our investigation.

4. Results

4.1. Pixel-level clustering

The following example illustrates the application of EM clustering to a set of pixel data. The data was gathered from a small image of the side of a road, approximately 10×30 m in size. The image was at a resolution of 1 m per pixel. The points are represented in the CIE *xy* chromaticity plane (Publication CIE No. 15.2, 1986). The CIE coordinate system has been chosen for simplicity. The EM algorithm was applied to this data set along with CIE Y and the output of an edge filter (Wilson and Bhalerao, 1992). This filter essentially measures gradient magnitude information for each of the data points. For the data in Fig. 6 the algorithm finds three dominant classes for the probability density model illustrated by the contour plot in Fig. 7. The ellipses of constant conditional probability of four less dominant classes are also illustrated. By inspecting the means of each of the Gaussian clusters found using the clustering procedure, one can assign physically meaningful labels to these clusters. Furthermore, if this procedure is applied separately to appropriately selected sets of data, then the mixture

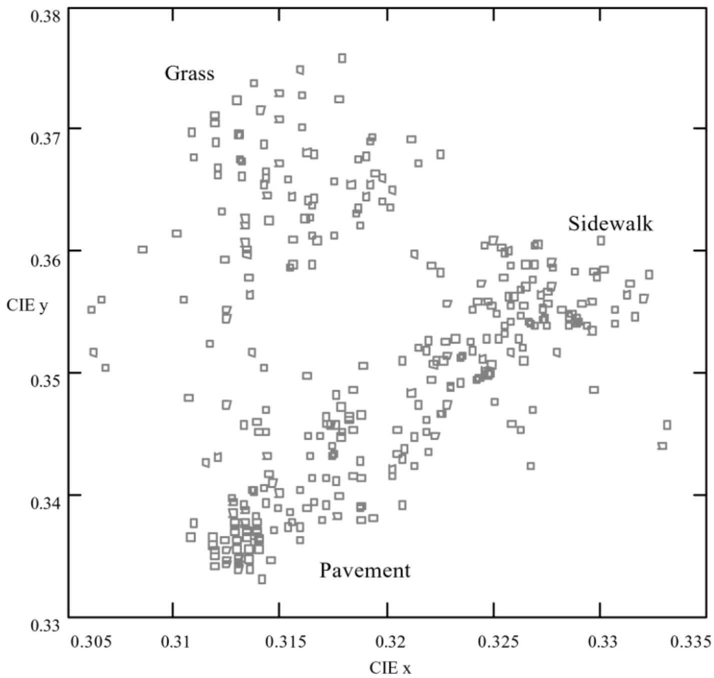

Fig. 6. Chromaticity coordinates for pixels from the side of a road.

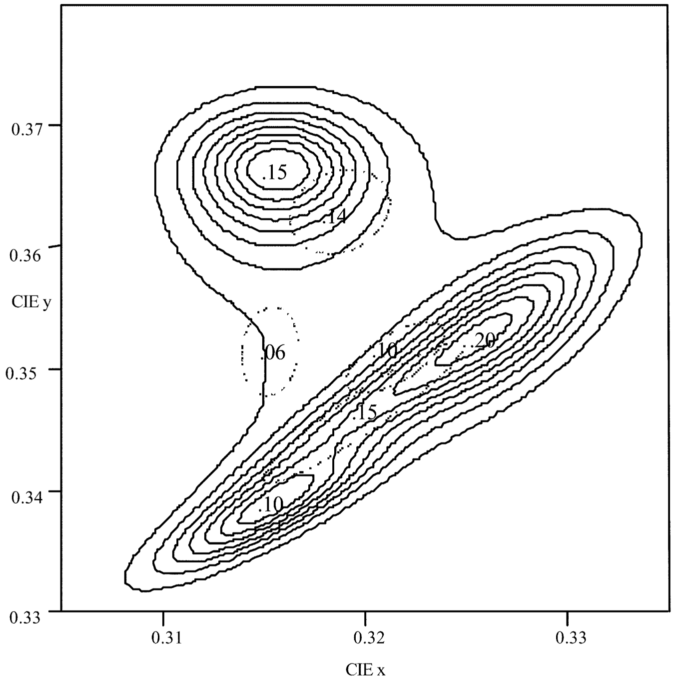

Fig. 7. A contour plot for the density model of $P(\vec{\mathbf{x}})$ with four additional ellipses illustrating the location of the less dominant classes. The priors for each Gaussian are indicated numerically in the center of the corresponding ellipse.

models can be combined into class–subclass models, as described previously, for the probability distribution over image features.

By using the Bayesian tree illustrated in Fig. 4, intuitively one can see how it is possible to alter the priors associated with each of the discrete class variables C from Fig. 5. This, in turn, influences the probabilities of the subclass variable K. In a more practical sense, consider the identification of trees within a residential neighborhood. A pixel level classi-

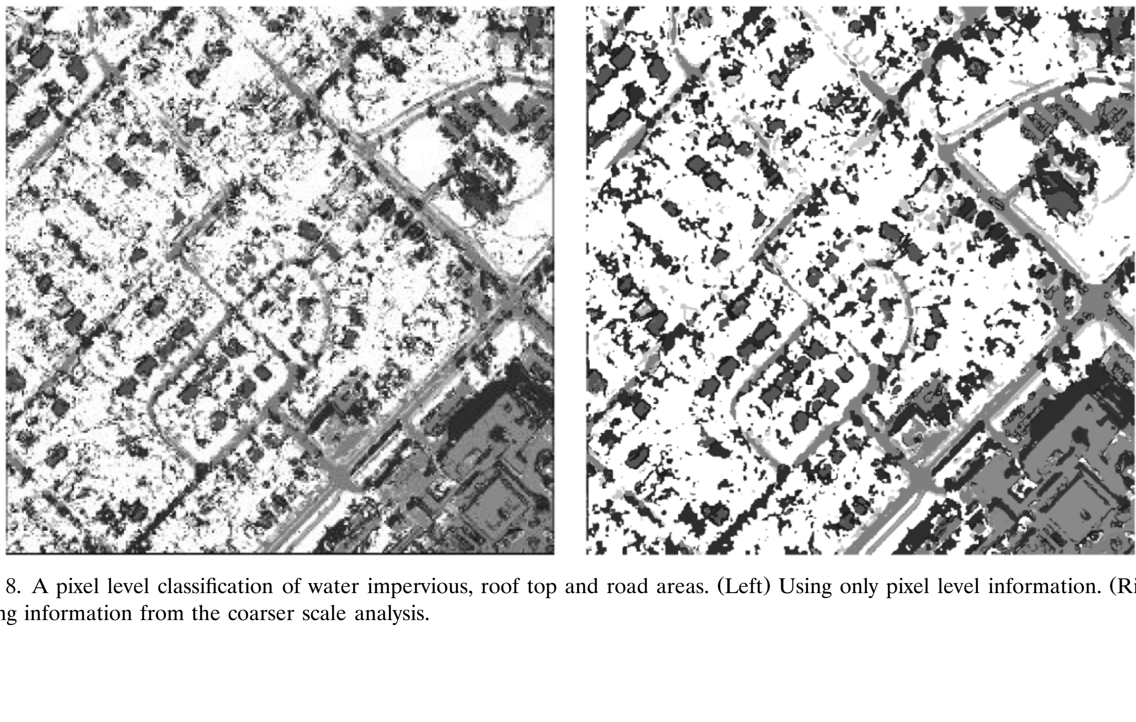

Fig. 8. A pixel level classification of water impervious, roof top and road areas. (Left) Using only pixel level information. (Right) Using information from the coarser scale analysis.

fier can easily be confused between green cars and green trees and green grass. The addition of contextual information derived from grayscale analysis of a window of pixels provides information taking the form of prior knowledge in the model. The tree model allows such information be passed in a statistically sound manner, as a probabilistic 'message' in the tree.

4.2. Using the model

In our investigations, we applied this clustering procedure to pixel level data and to the grayscale values for 16×16 pixel windows transformed into the fourth order Daubechies anisotropic wavelet domain (Press et al., 1992). A labeled 'training data set' was constructed from a subset of the sliding windows within the image. As such, the means of the Gaussian clusters found with the EM algorithm for these windows roughly correspond to image templates for different texture classes. As we are given differing labels for the data set for different resolution levels, the computation of the conditional probability distributions for fine scale or pixel level classes given coarse scale or window level classes is a straightforward calculation taken directly from frequency counts. Dirichlet priors (Heckerman, 1995) can be used to avoid zero elements in these probability tables when small amounts of training data are available. Observation information from these two analysis levels was then combined using the probabilistic message passing procedure in the tree-structured network as described earlier. For both the pixel level classifier and tree model, a pixel level Markov random field (Geman and Geman, 1984) was used to help clean up some of the misclassifications. A comparison of a pixel level classification not using the contextual information derived from the tree and a pixel level classification using the contextual information derived from the tree is given in Fig. 8 and Fig. 9.

5. Discussion and conclusions

We have presented a probabilistic methodology for the management and integration of prior knowledge in an image analysis system. We have described a relatively specific model. However, we emphasize that the formalism of the graphical probability model facilitates the intuitive development of differing models using alternate features, classes, resolution scales, and different numbers of resolution scales. Importantly, the formalism provides a statistically sound technique for integrating and managing uncertain knowledge. Similar techniques have been used in the past; however, the graphical formalism of the probability model provides a clear way to illustrate the manner in which the knowledge is integrated.

Our approach produces classification images that suffer from fewer errors, clearly observed as a reduction of the 'salt and pepper' appearance characteristic of classifiers that only use pixel level measurements. Such classification images are thus more suitable for subsequent conversion to vector format. However, our methodology is limited for extremely cluttered scenes. For the extraction of some features such as the com-

Fig. 9. A pixel level classification of pervious, green space areas. (Left) Using only pixel level information. (Right) Using information from the coarser scale analysis.

plete, connected road network within highly cluttered scenes, more specialized classifiers (e.g. Steger et al., 1997) must be used due to the high number of occlusions. Often, such specialized algorithms rely on generating good 'initial guesses' for the location of features. Our technique can thus provide such information. However, in some cases, the degree of occlusion in highly cluttered imagery may be so great that the only accurate way to verify the existence of a feature may be to make field measurements.

References

American Forests, 1998. Regional Ecosystem Analysis: Puget Sound Metropolitan Area. American Forests series of Urban Ecosystem Analyses. American Forests Publications.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Bouman, C.A., 1995. CLUSTER: An unsupervised algorithm for modeling Gaussian mixtures. Technical Report. Purdue University. http://www.ece.purdue.edu/~bouman/software/cluster/manual.pdf

Chou, K., Willsky, A., Benveniste, A., 1994. Multiscale recursive estimation, data fusion and regularization. IEEE Trans. Automat. Contr. 39 (3), 464–478.

Curtis, T.G., Huber, W.C., 1993. SWMM AML-An ARC/INFO Processor for the Storm Water Management Model (SWMM). Proc. 1993 Runoff Quantity and Quality Modeling Conference, Reno, NV, (NTIS, in press). US EPA, Athens, GA 30605.

Heckerman, D., 1995. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March, (revised: November 1995), 1996.

Dempster, A.P., Laird, M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Proc. R. Stat. Soc. B-39, 1–38.

Frey, B.J., 1998. Graphical Models for Machine Learning and Digital Communication. MIT Press, Cambridge, MA.

Geman, S., Geman, D., November 1984.

Huber, W.C., Dickinson, R.E., 1988. Storm Water Management Model, Version 4, User's Manual. EPA/600/3-88/001a (NTIS PB88-236641/AS). US EPA, Athens, GA, p. 30605.

Johnson, G.D., Ganapati, P.P., September 1988.

Mendel, J.M., 1995. Fuzzy logic systems for engineering: a tutorial. Proc. IEEE 83 (3), 345–377.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, CA.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in C. Second edition. Cambridge University Press, Cambridge, p. 591. Reprint with corrections, 1995.

Publication CIE No. 15.2, 1986. Colorimetry, Second Edition. Central Bureau of the Commission Internationale de L'Eclairage, Vienna, Austria.

Shafer, G.A., 1976. A Mathematical Theory of Evidence. Princeton University Press.

Steger, C., Mayer, H., Radig, B., 1997. The role of grouping for road extraction. Automatic Extraction of Man-Made Objects in Aerial and Space Images. Birkhauser Verlag, Basel, Switzerland, pp. 265–274.

Wilson, R., Bhalerao, A.H., 1992. Kernel designs for efficient multiresolution edge detection and orientation estimation. IEEE Trans. Pattern Anal. Machine Intelligence 14 (3), 384–389.