Modeling and Detection of Geospatial Objects Using Texture Motifs

Sitaram Bhagavathy, *Member, IEEE,* and B. S. Manjunath, *Fellow, IEEE*

***Abstract*— We propose the use of *texture motifs*, or characteristic spatially recurrent patterns for modeling and detecting geospatial objects. A method is proposed for learning a texture motif model from object examples and detecting objects based on the learned model. The model is learned in a two-layered framework—the first learns the constituent "texture elements" of the motif and the second, the spatial distribution of the elements. In the experimental session, we first demonstrate the model training and selection methodology for different objects given a limited dataset of each. We then emphasize the utility of such models for detecting the presence or absence of geospatial objects in large aerial image datasets comprising tens of thousands of image tiles.**

***Index Terms*— geospatial object, object detection, object model**

I. Introduction

Aerial and satellite images of the earth (or *geospatial* images) are critical sources of information in diverse fields such as geography, cartography, meteorology, surveillance, city planning. These images contain visual information about various natural and man-made features on or above the surface of the earth. Manual annotation of geospatial images covering even a relatively small area of the earth is a tedious task. This has necessitated research into automated annotation of geospatial images. An important component of this research comprises *object detection* methods, which are model-driven methods that seek to identify probable locations of specified features of interest or *objects* in geospatial images. For example, detection of buildings and roads is a useful step in cartography. Detection of objects such as harbors, airports, golf courses, housing colonies, vineyards, and parking lots is useful for updating geographical databases such as the Alexandria Digital Library (ADL) Gazetteer [1] which index the locations of several object types. Automated object detection is an important step toward an object-based representation of geospatial images.

The detection of geospatial objects with simple geometric or shape models such as buildings [2], [3], [4], [5], [6], roads [7], [8], [9], and other small objects [10], [11] has been explored adequately in the literature. This is not the case for *compound* objects, such as harbors and golf courses, characterized by several "parts" and their spatial layout. For example, harbors contain boats and golf courses contain trees and grass, both with a distinct spatial arrangement (Fig. 1).

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. Sitaram Bhagavathy is currently with Thomson Corporate Research, Princeton, NJ 08540. Email: {sitaram, manj}@ece.ucsb.edu. This research was supported by the following grants: NSF-DLI #IIS-49817432 and NSF IIS #0329267.

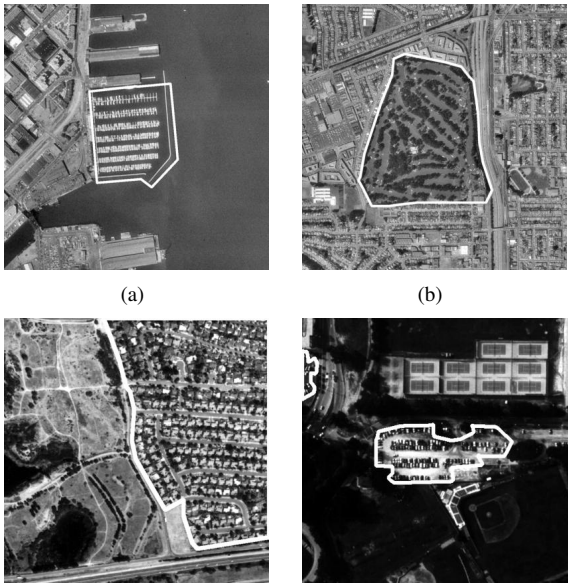


Fig. 1. Examples of geospatial objects: (a) a *harbor*, (b) a *golf course*, (c) a *housing colony*, and (d) a *parking lot*. The white borders show the extent of the object in the image.

There are two domains in which visual structure in images can be analyzed, namely the spatial domain (pixel intensities), and the frequency domain (fourier spectrum). The former has been the preferred domain for describing the structure of compound geospatial objects. Spatial analysis methods have been proposed for describing the constituents and layout of such objects. These methods usually divide an image into spatial units (closed regions, lines, etc.) through image segmentation or edge detection/linking. Spatial relations between units are analyzed using relational models such as production systems [12], semantic networks [13], [14], human-specified constraints or *rules* [15], [16], and evidential reasoning [17].

There are several obstacles to using strictly spatial analysis for the modeling and detection of compound objects. 1) Compound geospatial objects often contain a large number of parts, e.g. a harbor may contain hundreds of boats. 2) The structural relations among parts are often loose and vary from one object instance to another. In order to robustly recognize an object, this variation has to be accounted for. 3) Geospatial images are highly detailed, usually on the order of thousands of pixels in each dimension. These factors reduce the appeal of strictly spatial domain analysis methods for detecting compound objects.

In this paper, we propose solutions that combine informa-

tion from the spatial and frequency domains. These utilize joint space-frequency analysis techniques developed in the framework of texture analysis. The importance of texture as a visual cue for object detection has long been acknowledged in computer vision. Mahmood [18] showed that texture-based attentional selection could reduce the combinatorial search that occurs during object detection. Braithwaite and Bhanu [19] use tuned Gabor filters for detecting objects in infrared images with strongly oriented and periodic features. They demonstrate the detection of a tank by using a filter manually tuned to the frequency corresponding to the periodic pattern of the rows of wheels. Jain et al. [20] use Gabor filters to derive features which are then utilized for segmenting objects, such as tanks, cars, and fingerprints, in complex backgrounds. Although they demonstrate success in segmenting objects in a scene, the problem of actually detecting a specified object is not addressed. Schmid [21] proposed a method to construct models for objects with texture-like visual structure given positive and negative example images. Schmid uses rotation-insensitive features which could be a disadvantage if orientational relationships between spatial patterns is important.

The role of image texture in geospatial image analysis has mostly focussed on the classification of certain types of land-cover such as terrain types, crops, and urban settlements (for example, [22], [23], [24], [25], [26], [27], [28]). In this paper, we extend the use of texture analysis to model-driven detection of compound geospatial objects such as harbors, golf courses, and so on. In a nutshell, this paper proposes methods that apply frequency-domain texture analysis to address the problems of 1) detecting compound objects in geospatial images, and 2) learning appearance models for such objects from examples.

The organization of this paper is as follows. Sec. II introduces the concept of texture motifs and its application to object modeling and detection in geospatial images. Sec. III lays down the fundamentals of texture analysis using Gabor filters. Sections IV, V, and VI describe the method used for learning a model for a texture motif given object examples. Sec. VII provides the experimental results which include model training using examples and application of the learned models to the detection of geospatial objects in large aerial image datasets. Sec. VIII concludes with a discussion of future research directions.

II. Texture Motifs for Object Detection

Texture analysis provides a framework for the efficient analysis of recurrent and possibly regular arrangements of image primitives. At a lower-level, such primitives may be a set of local intensity patterns including edges, bars, and smooth regions. At a higher level, they may correspond to physical features such as cars, boats, trees, water, and so on, by whose repetitive spatial arrangements, several geospatial objects are formed. Consider the *harbor* object which contains the recurrent pattern formed by the arrangement of boats and water. Harbors may be detected by detecting boats via their model shapes after segmentation, and finding those that occur in certain regular arrangements which are modeled a priori. This is how spatial analysis methods discussed earlier would proceed. However, the description and detection complexity could be significantly reduced by using frequency-domain texture analysis. There are many advantages in using frequency-domain texture analysis to describe spatially recurrent patterns. 1) Frequency-domain texture analysis is generally less computationally expensive than image segmentation and edge detection/linking, especially for large and highly detailed geospatial images. 2) Texture analysis using a Gabor filter bank provides a compact description of visual structure present in a neighborhood. 3) Texture can gracefully handle variation in object appearance. Texture analysis can capture regularity in a pattern as well as tolerate a degree of randomness. For example, the boats in a harbor are moored with approximately the same distance to each other but with some variance.

Several geospatial objects contain recurrent spatial patterns with distinct visual appearance. For example, observe the patterns in a *harbor* (Fig. 1) formed by the arrangement of boats and water, and that formed by the arrangement of trees and grass in a *golf course*. These patterns enable most humans to easily recognize the corresponding object, provided that they have seen it before (even if only briefly). Such spatially recurrent patterns that are characteristic of an object are termed the *texture motifs* of the object. Thus, the pattern formed by boats and water is a texture motif of a harbor, and the arrangement of trees and grass is a texture motif of a golf course. The problem of detecting objects can now be translated to that of detecting their texture motifs.

The concept of texture motifs leads to texture-based computational models for many objects, which can be applied to object detection. This approach offers a powerful alternative to shape-based and edge-based models, which are prohibitively expensive to compute, due to the level of complexity and detail often found in geospatial objects. Of course, not all geospatial objects contain texture motifs. We restrict our treatment to those that do. Examples of objects with texture motifs include golf courses, harbors, trailer parks, vineyards, and airports.

III. Gabor Filters and Visual Structure

Spatially recurrent patterns have the property of being distinctive in both their spatial appearance and in their frequency distribution. Thus the spatial appearance of such patterns can be studied via their frequency domain characteristics. By performing texture analysis using Gabor filters at different scales and orientations, these patterns can be efficiently 1) described in the frequency domain, and 2) localized in the spatial domain.

Texture analysis is performed by applying a bank of scale and orientation selective Gabor filters to an image. These filters are constructed as follows [29]. A two-dimensional Gabor function $g(x, y)$ can be written as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \quad (1)$$

A class of self-similar functions referred to as Gabor wavelets is now considered. Let $g(x, y)$ be the mother wavelet. A *Gabor filter bank* can be obtained by appropriate dilations and

translations of $g(x, y)$ through the generating function:

$$\begin{aligned} g_{s,k}(x,y) &= a^{-s} g(x', y'), \ a > 1 \\ x' &= a^{-s}(x\cos\theta + y\sin\theta) \quad \text{and} \\ y' &= a^{-s}(-x\sin\theta + y\cos\theta) \end{aligned} \tag{2}$$

where $s \in 0, \ldots, S-1$, $k \in 0, \ldots K-1$, and $\theta = k\pi/K$ is the orientation of the filter w.r.t. the vertical. The indices k and s indicate the orientation and scale of the filter respectively. K is the total number of orientations and S is the total number of scales in the filter bank. The filter bank parameters $\{\sigma_x, \sigma_y, a, W\}$ are computed by the method described in [29], given the input specifications S, K, and the upper and lower center frequencies, U_h and U_l.

The texture in the neighborhood of a pixel is represented by an SK-dimensional feature vector obtained by convolving the image with a Gabor filter bank at S scales and K orientations. Let $\mathbf{c}(\mathbf{x})$ denote the feature vector extracted from the neighborhood of pixel $\mathbf{x} = [x \ y]^T$. This feature vector is given by

$$\mathbf{c}(\mathbf{x}) = [F_{0,0}(\mathbf{x}) \ F_{0,1}(\mathbf{x}) \ \ldots \ F_{S-1,K-2}(\mathbf{x}) \ F_{S-1,K-1}(\mathbf{x})]^T, \tag{3}$$

where $F_{s,k}(\mathbf{x})$ is the filter output at pixel $\mathbf{x}$, obtained by convolving the image $I(\mathbf{x})$ with the filter $g_{s,k}(\mathbf{x})$. In other words, $F_{s,k}(\mathbf{x}) = |g_{s,k}(\mathbf{x}) * I(\mathbf{x})|$.

Texture descriptors derived from Gabor filter banks have been widely used for browsing and similarity retrieval in image databases [30], [31], [32], [33]. Gabor filter-based texture analysis has the ability to describe higher-order structure in objects. We exploit this ability to address the problem of visually detecting geospatial objects containing texture motifs. Consider, for example, the harbor object. By choosing Gabor filters at appropriate scales and orientations, it is possible to localize the pattern of boats parked side by side and the pattern corresponding to the rows of boats. It is thus possible to localize these patterns in the spatial domain without having to perform image segmentation or edge detection/linking. This implies a decrease in the complexity of object description and search.

IV. Learning Object Models from Examples

The problem of learning models for objects is posed as a problem of learning a representation for the texture motifs of the objects from low-level texture features extracted from examples.[1] Building on the work done in [34], this section presents a probabilistic framework for this learning problem. We ask ourselves the following question.

How do we represent the visual appearance of a texture motif, say, the arrangement of boats and water in a harbor?

There are different aspects that constitute this appearance. Firstly, there are the local intensity variations that form textural elements such as flat areas, bars, edges, and so on. These can be interpreted as the low-level building blocks of the motif. For example, they may correspond to water, boats, and edges between them. It has been shown in the previous section that these local intensity variations can be effectively captured and described by low-level texture features based on Gabor filters at multiple scales and orientations. Assuming that the texture features generated by different elements populate different volumes of the texture feature space, it is possible to statistically learn the elements of a pattern. In this work, a semi-supervised statistical approach is adopted for this task. This forms the first layer of the overall representation of the texture motif.

The second layer of the representation is the spatial distribution of low-level texture elements in the motif, since this influences its distinct visual appearance. A Gaussian mixture model (GMM) for this is learned from examples using features derived from histograms of texture elements in spatial neighborhoods. Confidence measures generated using this model are then used for detecting object presence.

V. Learning the Texture Elements of a Motif

Suppose we are given M examples of an object that contains one or more texture motifs. Let us further assume that all the motifs are formed by a spatial combination of N_t texture elements. Then the N_t elements are learned from low-level texture features extracted from the examples, in order to arrive at the first layer of representation. Let us uniformly sample a number of texture features (usually proportional to the size of the example) from each of the M object examples. If the object consists of multiple texture elements, the sampled vectors form several clusters in the texture feature space. Let each cluster be considered to represent a distinct texture element.

It can be argued that as M becomes large, the N_t largest clusters formed by the sampled vectors correspond to the texture elements in the object. The argument is justified thus. The more examples a texture element appears in, the more the evidence in favor of it being an important texture element of the object. If an element occurs in very few examples, it is less likely to be critical to the description of the object. With increasing M, clusters formed by features occurring in a majority of examples are expected to become dominant. On the other hand, clusters formed by features that occur in just a few examples become relatively smaller.

In this work, Gaussian mixture models (GMM) are applied to solve the clustering problem in a semi-supervised approach. Mixtures of Gaussians have been used to model image feature distributions for a variety of research objectives. In [35], texture-based image segmentation is performed by clustering texture feature vectors using mixtures of Gaussians. In the Blobworld system [36], mixtures of Gaussians are used to derive image descriptors for content-based retrieval. The Expectation-Maximization (EM) algorithm is used to discover the feature vector groupings that correspond to the visual blobs in an image. There are several factors that motivated us to use a GMM to cluster texture features instead of the simpler K-means algorithm. Firstly, a GMM accounts for the density of each cluster. This is important because the feature vectors from different textures are observed to have different densities of distribution in the feature space. Secondly, GMM has a

[1] An object "example" here refers to an image containing an instance of the object, along with a binary mask that isolates the object region from the background (see Fig. 3).

parametric representation that allows easy model comparison. Finally, the EM framework can be extended to elegantly handle rotations of textural patterns (Sec. V-D).

We model texture features that occur in an object as a GMM with N_t Gaussian components. Each component in the GMM corresponds to one texture element. In other words, the features corresponding to each texture element is assumed to be follow a Gaussian distribution. It is also possible to train the GMM with a $N_t' > N_t$ components and choose the N_t most probable ones as corresponding to the texture elements. In this work, the choice of N_t is made by the user based on experimental evidence. As will be described in Sec. VII-A, the modeling parameters including N_t are chosen to obtain the "best" object detection performance in terms of *precision* and *recall*. The best parameters are chosen from a set of candidate parameters determined by the user based on visual inspection of the object examples.

A. The GMM Framework

Assuming that there are N_t texture elements in an object, the probability density function of $\mathbf{c}(\mathbf{x})$ (or simply $\mathbf{c}$) can thus be expressed as a mixture distribution,

$$p_t(\mathbf{c}) = \sum_{j=1}^{N_t} P_t(j) p_t(\mathbf{c}|j), \tag{4}$$

where $p_t(\mathbf{c}|j)$ is the conditional pdf of the feature $\mathbf{c}$ generated by the j^{th} texture element and $P_t(j)$ is the prior probability of the j^{th} element. The subscript t is used to clarify that we are learning the *texture elements*. This subscript is applied to all parameters and probabilities in the first layer of texture motif representation.

The conditional pdf $p_t(\mathbf{c}|j)$ is Gaussian and is given by

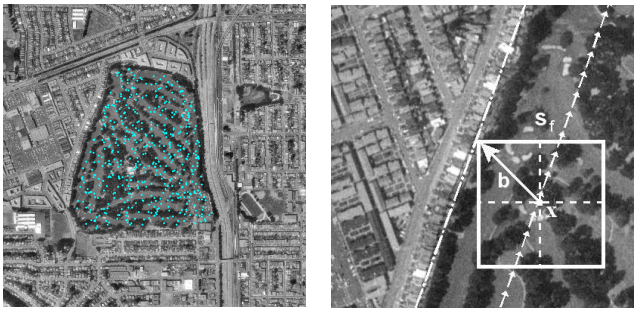
$$p_t(\mathbf{c}|j) = \frac{\exp\left[-\frac{1}{2}(\mathbf{c}-\boldsymbol{\mu}_{tj})^T \Sigma_{tj}^{-1}(\mathbf{c}-\boldsymbol{\mu}_{tj})\right]}{(2\pi)^{d/2}|\Sigma_{tj}|^{1/2}}, \tag{5}$$

where d is the dimensionality of $\mathbf{c}$. The number of elements N_t along with the distribution means and covariance matrices are the parameters that specify the object model Θ_t. In other words,

$$\Theta_t = \{(P_t(j), \boldsymbol{\mu}_{tj}, \Sigma_{tj}); j = 1 \ldots N_t\}. \tag{6}$$

The EM algorithm [37] is used to estimate the GMM parameters from training data, which are obtained from object examples as described in the following section.

Note that the texture element model Θ_t is learnt separately for each object and not over all objects. A high number of texture elements (N_t) is needed to describe texture motifs across all objects. As will be seen later, N_t determines the dimensionality of the second layer of texture motif representation. A small N_t is desirable with regard to the complexity and reliability of the next learning stage. Therefore, we learn texture elements in an object-specific manner.



(a) (b)

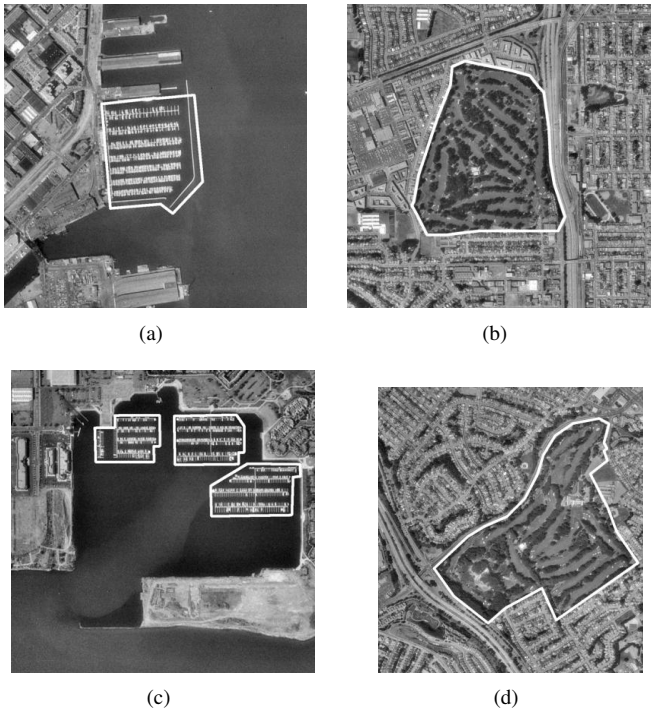
Fig. 2. **Sampling methodology:** (a) Uniform random sampling of pixels from the golf course object. The pixels around which the texture features are sampled are marked as dots. (b) Illustration of the valid sampling region. To prevent the square Gabor kernel from exceeding the object border (dot-dash line), its center (pixel $\mathbf{x}$) should stay within the short-arrows line.

B. Feature Sampling for GMM Learning

The training set for an object consists of a set of examples or instances, such as those shown in Fig. 3. Each instance is provided as an image and an associated mask delineating the object region. The texture samples for training the GMM are drawn from pixels strictly inside the object regions, as depicted in Fig. 2(a). Of course, texture is a neighborhood property, not a pixel property. The texture features are generated by convolving the image with square Gabor filter kernels. Let s_f be the *kernel size*, i.e. the length of its side in pixels. We need to make sure that the sampled texture features are not "corrupted" by intensity variations outside the object region. This implies that if the kernel is centered at an object pixel, no part of it should project outside the object (Fig. 2(b)). This results in the exclusion of a band of pixels at the borders of the object region. The width of this band is $s_f/\sqrt{2}$ pixels in the worst case when the border is parallel to a diagonal of the kernel, and $s_f/2$ pixels in the best case when it is parallel to a side of the kernel. The object region minus this band is termed the *valid sampling region*. In practice, the valid sampling region is obtained by morphological erosion of the binary mask image with a square structuring element of side s_f pixels.

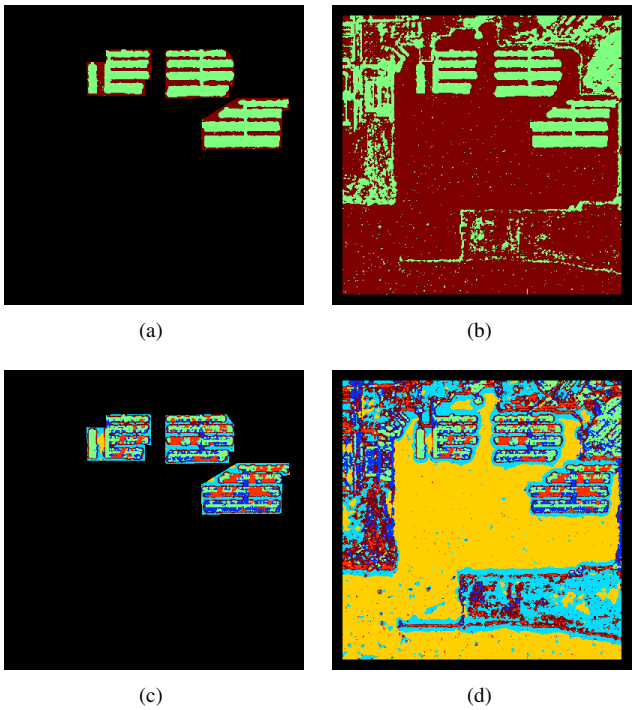
Let the training set for an object be denoted by $\mathcal{O} = \{R_{v,1}, R_{v,2}, \ldots, R_{v,N_o}\}$ where $R_{v,i}$ is the valid sampling region of the i^{th} example and N_o is the number of training examples of the object. From each example i, $n_i = \beta|R_{v,i}|$ features are sampled uniformly, where $|R_{v,i}|$ is the number of pixels in $R_{v,i}$ and β is chosen according to the acceptable complexity (depending on available CPU speed, memory, etc.) of the GMM learning task. The training data for learning the GMM is obtained from the union of the sampled features from each example in the training set. Thus the GMM is learned from a total of $\sum_i n_i$ sampled features.

Having obtained this training data, the EM algorithm [37] is used to estimate the parameters of the GMM, which are given by (6). A K-means clustering process is applied to bootstrap the EM algorithm. After the learning process, each Gaussian component in the mixture represents one texture element in the object. The prior probability of each component gives information about the relative contribution of that texture



(a) (b) (c) (d)

Fig. 3. Examples from a training set for the "harbor" object (left column) and the "golf course" object (right column). The borders of the object regions (masks) are indicated in white.



(a) (b) (c) (d)

Fig. 4. The texture element labelings for the harbor image in Fig. 3(e), with $N_t = 2$ (top row), and $N_t = 6$ (bottom row). Each color corresponds to a label. The top row shows the labels inside the object and the bottom row shows the overall labelings.

element in forming the object.

C. Texture Element Labeling

After a GMM has been learned for an object, a maximum a posteriori (MAP) classifier is used to label any pixel $\mathbf{x}$ to its generating texture element $i^*(\mathbf{x})$, as follows.

$$i^*(\mathbf{x}) = \underset{1 \le i \le N_t}{\arg\max} P_t\left(i|\mathbf{c}(\mathbf{x})\right), \tag{7}$$

where $P_t(i|\mathbf{c}(\mathbf{x}))$ is the probability that the feature vector $\mathbf{c}(\mathbf{x})$ came from the i^{th} Gaussian component of the GMM. The posterior probabilities $P_t(i|\mathbf{c}(\mathbf{x}))$ are obtained using Bayes' rule as follows,

$$P_t(i|\mathbf{c}) = \frac{p_t(\mathbf{c}|i)P_t(i)}{\sum_i p_t(\mathbf{c}|i)P_t(i)}. \tag{8}$$

Fig. 4 shows the texture element labels assigned to a harbor training image, using GMMs learned from the harbor examples in Fig. 3. Different labelings are shown for the same image, obtained by learning GMMs with different N_t (number of components). The function $p_t(\mathbf{c}(\mathbf{x}))$ gives the density in the feature space at the point corresponding to $\mathbf{c}(\mathbf{x})$. However, the magnitude of $p_t(\mathbf{c}(\mathbf{x}))$ does not directly convey the confidence of a pixel belonging to the object. A reason for this, in addition to the curse of dimensionality, is that the texture elements of harbor occur in other regions as well. This is clear by observing the top and bottom rows in Fig. 4. Therefore, it is the spatial arrangement of these elements that distinguish harbors from other objects.

D. A Note on Rotation Invariance

A major obstacle in learning the texture elements with the above GMM formulation is that the texture features $\mathbf{c}(\mathbf{x})$ (given by (3)) are derived from orientation-selective Gabor filters and are therefore sensitive to the orientation of the texture element (and therefore the motif/object). Texture elements recurring in several examples can be learned consistently only when the objects in the examples have similar orientations. This is the case with the harbor examples in Fig. 3, but often in practice the training examples have arbitrary orientations.

Suppose the texture features are derived from Gabor filters oriented at 30° intervals, i.e. 0°, 30°, 60°, and so on. Then a 30° rotation of the texture is equivalent to a circular shifting of the feature vector components at each scale. Hence, the features sampled from a textural pattern of varying orientation form multiple clusters in the feature space. In order to handle objects with varying orientations, the number of Gaussian components in the GMM has to be adjusted to take into account the additional clusters formed by variation in orientation. Then the following question arises. Which clusters are associated with the same texture element, i.e. caused by a rotation of the same element? This is a difficult question to answer. Furthermore, to model motif appearance at different orientations, it is necessary to augment the training set by considering all orientations of the training instances. This increases the complexity of the learning process.

Alternatively, Newsam [38] has proposed a variation to the EM algorithm that takes the orientation of a texture into account while training a GMM. By treating the (discretized) orientation of a pattern as a missing variable in the EM

framework, the equivalence between rotated patterns is learned automatically. In the resulting GMM, each Gaussian component corresponds to a cluster of "orientation-normalized" features. This variation to the EM algorithm is described below.

The Orientation-Normalized GMM [38]

The conditional probability of a feature vector $\mathbf{c}$, given that it is generated from component j and its orientation index is k, is written as

$$p_t\left(\mathbf{c}|j,k\right)=\frac{\exp\left[-\frac{1}{2}\left(\mathbf{c}_k-\boldsymbol{\mu}_{tj}\right)^T\Sigma_{tj}^{-1}\left(\mathbf{c}_k-\boldsymbol{\mu}_{tj}\right)\right]}{\left(2\pi\right)^{d/2}\left|\Sigma_{tj}\right|^{1/2}}. \quad (9)$$

The term $\mathbf{c}_k$ is the vector $\mathbf{c}$ circularly shifted by k orientations where $k\in\{1,..,K\}$. Note that the orientation k is with respect to the normalized orientation of the mixture component. The pdf of the feature distribution in an object class is modeled as a N_t-component GMM,

$$p_t\left(\mathbf{c}\right)=\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{N_t}p_t\left(\mathbf{c}|j,k\right)P_t(j), \quad (10)$$

where we have assumed that the orientation k is independent of j and equiprobable (in the absence of a priori information).

Each component represents a single texture element in a manner oblivious to its orientation. This model is completely specified by the parameters $\Theta_t=\{(P_t\left(j\right),\boldsymbol{\mu}_{tj},\Sigma_{tj})\,;j=1\ldots N_t\}$. A modified version of the EM algorithm is used to estimate the parameters of the GMM. Rotation is taken into account by modifying the EM algorithm to include the orientation k of the feature vector as additional missing data.

The procedure in (7) for labeling each pixel $\mathbf{x}$ to its texture element $i^*(\mathbf{x})$ has to modified as well. It becomes

$$i^*(\mathbf{x})=\underset{1\le i\le N_t}{\arg\max}\left[\max_k P_t\left(i|\mathbf{c}_k(\mathbf{x})\right)\right], \quad (11)$$

where

$$P_t(i|\mathbf{c}_k)=\frac{p_t(\mathbf{c}|i,k)P_t(i)}{\sum_i p_t(\mathbf{c}|i,k)P_t(i)}, \quad (12)$$

assuming that the orientations k are equiprobable.

VI. Spatial Distribution of Texture Elements

It can be observed from Fig. 4 that the spatial configuration of the labels inside the "boats and water" texture motif of the harbor region is quite different from that outside. Then, the task of the second layer is to describe this spatial configuration of labels, and model its variation within the motif. A simple method of describing the spatial distribution of labels is by the use of a *spatial histogram*, as shown in Fig. 5. The descriptor at a pixel $\mathbf{x}$, denoted as $\mathbf{h}(\mathbf{x})$, is the vector of normalized frequencies of texture element labels in a square window centered at $\mathbf{x}$. In other words, $\mathbf{h}(\mathbf{x})=[h_1(\mathbf{x})\ h_2(\mathbf{x})\ \ldots\ h_{N_t}(\mathbf{x})]^T$ where $h_l(\mathbf{x})$ is the normalized frequency of label l in the window. Obviously, the dimensionality of the above descriptor is N_t, the total number of texture element labels.

The texture element label $i^*(\mathbf{x})$ at a pixel $\mathbf{x}$ is given by (11). If we temporarily write $i^*(\mathbf{x})$ as $i^*(x,y)$ (since $\mathbf{x}=[x\ y]^T$), then $i^*(x+x_o,y+y_o)$ is the label at an offset of (x_o,y_o) from $\mathbf{x}$. Let $I_l(z)$ be an indicator function that is 1 if $z=l$ and 0 otherwise. Then $h_l(\mathbf{x})$ can be computed as

$$h_l(\mathbf{x})=h_l(x,y)=\frac{1}{s_h^2}\sum_{x_o=-\frac{s_h-1}{2}}^{\frac{s_h-1}{2}}\sum_{y_o=-\frac{s_h-1}{2}}^{\frac{s_h-1}{2}}I_l(i^*(x+x_o,y+y_o)), \quad (13)$$

where s_h (usually an odd number) is the length of the side of the square window around pixel $\mathbf{x}$ (Fig. 5). The scale of the description depends on the value of s_h. Note that the spatial histogram feature $\mathbf{h}(\mathbf{x})$ is non-directional and coarse because it only gives an idea of the relative presence of the labels in a 2-D neighborhood of a pixel.

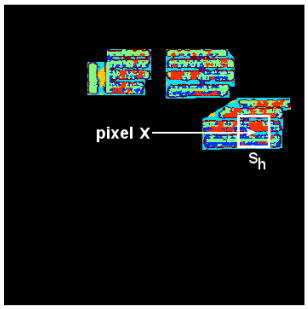


Fig. 5. The spatial histogram of texture element labels is built by taking a square window of size s_h around pixel $\mathbf{x}$. The normalized frequencies of the $N_t=6$ labels inside the window forms a 6-dimensional vector $\mathbf{h}(\mathbf{x})$.

A. Learning the Second Layer

Once again, the GMM is employed to model the variation of $\mathbf{h}(\mathbf{x})$ in the object region. Let the variation in the spatial configuration be modeled by a GMM with N_s components as follows,

$$p_s(\mathbf{h})=\sum_{j=1}^{N_s}P_s(j)p_s(\mathbf{h}|j), \quad (14)$$

where the conditional pdfs $p_s(\mathbf{h}|j)$ are Gaussian, given by

$$p_s\left(\mathbf{h}|j\right)=\frac{\exp\left[-\frac{1}{2}\left(\mathbf{h}-\boldsymbol{\mu}_{sj}\right)^T\Sigma_{sj}^{-1}\left(\mathbf{h}-\boldsymbol{\mu}_{sj}\right)\right]}{\left(2\pi\right)^{N_t/2}\left|\Sigma_{sj}\right|^{1/2}}. \quad (15)$$

The model for the second layer of representation is then specified by

$$\Theta_s=\{(P_s\left(j\right),\boldsymbol{\mu}_{sj},\Sigma_{sj})\,;j=1\ldots N_s\}. \quad (16)$$

The subscript s here is used to clarify that we are learning the *spatial distribution* of the texture elements in the motif.

The training data is obtained by sampling spatial histograms $\mathbf{h}(\mathbf{x})$ around several pixels $\mathbf{x}$ inside the object region. The procedure for sampling and creating the training data for the GMM is similar to that in Sec. V-B. The valid sampling region in this case is obtained by morphological erosion of the binary mask image using a square structuring element with a side length of $\max(s_f,s_h)$ pixels, i.e. the larger between the filter kernel size and the spatial neighborhood size. This ensures that neither the texture features nor the histograms at the sampled pixels are influenced by non-object pixels. The

sampling parameter β is again chosen appropriately, and need not be the same as the value chosen for the first layer. After creating the training data, the GMM is learned via the EM algorithm. It should be clear that, since the features are non-directional, the conventional GMM formulation is used and not the orientation-normalized version.

Having learned Θ_s, the density function $p_s(\mathbf{h}(\mathbf{x}))$ can be interpreted as the confidence of finding at a pixel $\mathbf{x}$, the spatial configuration corresponding to the learned texture motif. Experimental results indicate that p_s is a good measure for the confidence of a pixel belonging to a texture motif (or the object containing the motif). The importance of the spatial arrangement of texture elements for describing a motif is evident from this.

VII. Experimental Results

This section comprises two parts. The first one concerns the evaluation of the trained models, using a limited dataset of object examples, it order to select the best model for a real application. The second part discusses a real application of the selected model, which is to drastically reduce the manual labor involved in ascertaining the presence and location of specified objects in large aerial image datasets.

A. Results on Training and Model Selection

The primary dataset chosen for our study consists of aerial images drawn from the Digital Orthophoto Quarter-Quadrangle (DOQQ) coverage of California, which is available through the Alexandria Digital Library (ADL). The ADL Gazetteer [1] is a resource that provides georeferencing information for several objects (synonymous with *feature types* in [1]). Several instances of objects such as harbors, golf courses, and airports, can be located through the Gazetteer. The corresponding aerial images are then extracted from the ADL DOQQ coverage. Each object instance used for training is provided in two pieces: a) a rectangular image region containing the object, and b) a manually created binary mask defining the object region.

Model Evaluation Methodology:

Suppose, for an object of study, we have a training set and a test set of example images, with their corresponding masks. From the training set, the GMMs Θ_t and Θ_s are learned as described in Sections V–VI. The specifications of the Gabor filter bank used for extracting texture features, $\mathbf{c}(\mathbf{x})$ in (3), are $S = 5$, $K = 6$, $U_l = 0.05$ and $U_h = 0.4$ (see Sec. III). The filter kernel size s_f (see Fig. 2) is set to 75 pixels, so as to support the filter with the largest spatial extent, in the filter bank. In the testing stage, the learned models are applied to each instance of the test set in three steps as follows.

1) Application of Θ_t to obtain the texture element labels $i^*(\mathbf{x})$ as described in Sec. V-C. Note that in practice, the orientation-normalized GMM is used and the labels are obtained using (11).
2) Computation of the spatial histogram features $\mathbf{h}(\mathbf{x})$ from the label field $i^*(\mathbf{x})$, as described in Sec. VI.
3) Application of Θ_s to obtain the confidence measure $p_s(\mathbf{h}(\mathbf{x}))$, as described in Sec. VI-A.

The main tool used for evaluating the performance of the proposed approach is the *precision-recall graph*. These are obtained by computing precision and recall while varying the threshold t_o on the confidence measure, $p_s(\mathbf{h}(\mathbf{x}))$. Let $I_o(\mathbf{x})$ be an indicator function, which has a value 1 if pixel $\mathbf{x}$ lies inside the object region (defined by the user-provided masks) and 0 if it does not. Let us define another indicator function $I_{t_o}(\mathbf{x})$ thus,

$$I_{t_o}(\mathbf{x}) = \begin{cases} 1, \text{if } p_s(\mathbf{h}(\mathbf{x})) > t_o \\ 0, \text{else.} \end{cases} \tag{17}$$

Now, for a given t_o, precision $\mathcal{P}(t_o)$ and recall $\mathcal{R}(t_o)$, are defined as,

$$\mathcal{P}(t_o) = \frac{\sum_i I_o(\mathbf{x_i}) I_{t_o}(\mathbf{x_i})}{\sum_i I_{t_o}(\mathbf{x_i})} \text{ and } \mathcal{R}(t_o) = \frac{\sum_i I_o(\mathbf{x_i}) I_{t_o}(\mathbf{x_i})}{\sum_i I_o(\mathbf{x_i})}, \tag{18}$$

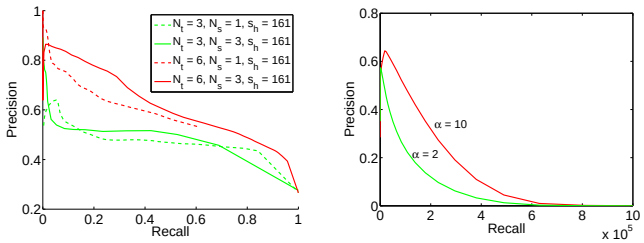
where $\mathbf{x}_i$ are indexed over all the pixels in the test images, both inside and outside the object region. Therefore, *precision* tells us how many pixels are correctly identified as belonging to the object. The *recall* tells us how many pixels belonging to the object are correctly identified as such. The precision-recall graph plots $\mathcal{P}(t_o)$ against $\mathcal{R}(t_o)$ while varying t_o. It displays the tradeoff between precision and recall at different thresholds.

Results:

Two geospatial objects are selected for comprehensive testing of the proposed modeling approach. These are *golf courses* and *harbors*. The dataset for golf courses contains nine instances, and that for harbors contains six. Since the datasets are small, the experimental methodology employs cross-validation techniques. Cross-validation implies that each instance is used in turn for testing, while being excluded from the training set. This technique enables more comprehensive testing on all the instances, which is not possible by rigidly partitioning the dataset into one training set and one test set. The cross-validation strategy is applied as follows. The nine instances in the golf course dataset are randomly partitioned into three sets of three instances. Each set is used in turn as the test set, while the training set comprises the union of the remaining sets. Similarly, cross-validation for harbors is done by dividing the dataset into two sets of three instances. In the end, we shall have tested and obtained $p_s(\mathbf{h}(\mathbf{x}))$ for all instances in the dataset. The precision-recall graph is then plotted by applying (18) to the aggregated test results, for different t_o.

Fig. 6(a) shows the precision-recall graph for the golf course dataset, for different modeling parameters (N_t, N_s, and s_h). A plot that lies entirely above another is better, since it implies a higher precision and recall for all thresholds. Therefore, the aim is to attain "higher" plots by choosing modeling parameters wisely. In practice, the plots may intersect one another. When this happens, the model is chosen according to the relative merits of the intersecting plots, e.g. the one that gives higher precision at the required recall rate. It can be observed from Fig. 6(a) that the best overall model (among the ones considered) has parameters $N_t = 6$, $N_s = 3$, and $s_h = 161$.

For object detection, a proper threshold t_o has to be chosen that results in a high confidence of detecting the object (high



(a) (b)

Fig. 6. (a) Precision-recall curves with different modeling parameters for the golf course dataset, and (b) F-measure $\mathcal{F}_\alpha(t_o)$ for different α computed using the golf course dataset with modeling parameters $N_t = 6$, $N_s = 3$, and $s_h = 161$.

recall) and a low false-alarm rate (high precision). All pixels $\mathbf{x}$ that have $p_s(\mathbf{h}(\mathbf{x})) > t_o$ shall then be denoted as object pixels. Often, the choice if t_o is based on a trade-off between precision and recall. This process is simplified by means of the *F-measure* [39] which combines precision and recall into one measure that depends on t_o. The F-measure is the harmonic mean of precision and recall, and is defined as,

$$\mathcal{F}_\alpha(t_o) = \frac{1}{\frac{1}{1+\alpha}\left(\frac{\alpha}{\mathcal{P}(t_o)} + \frac{1}{\mathcal{R}(t_o)}\right)} = \frac{(\alpha+1)\mathcal{P}(t_o)\mathcal{R}(t_o)}{\mathcal{R}(t_o) + \alpha\mathcal{P}(t_o)}, \quad (19)$$

where $\alpha \in [0, +\infty)$ is the relative weight placed on precision over recall. Fig. 6(b) plots $\mathcal{F}_\alpha(t_o)$ against t_o for different α values, choosing $N_t = 6$, $N_s = 3$, and $s_h = 161$. The threshold value t_o^* corresponding to the peak of the plot (with desired α) is chosen for object detection purposes. Fig. 7 shows the detected golf course regions using t_o^*, for $\alpha = 10$ and $\alpha = 2$. The correctly detected regions are the ones inside the object regions specified by the black borders. Note that with a lower α, recall is higher at the expense of precision resulting in both a higher detection rate and false-alarm rate. Note also that the many of the falsely detected regions correspond to a trees-and-grass texture motif quite similar to that found in golf courses.

Fig. 8(a) shows the precision-recall graph for the harbor dataset, for different modeling parameters (N_t, N_s, and s_h). The best model parameters (among the ones considered) in this case are $N_t = 3$, $N_s = 1$, and $s_h = 51$. For this model, Fig. 8(b) plots $\mathcal{F}_\alpha(t_o)$ against t_o for different α values. The threshold value t_o^* corresponding to the peak of the plot (with desired α) is chosen for object detection purposes. Fig. 9 shows the detected harbor regions using t_o^*, for $\alpha = 10$ and $\alpha = 2$. Note a lower α leads to a higher detection rate at the expense of increasing the false-alarm rate.

Fig. 10 and Fig. 11 demonstrates object detection in larger geospatial images containing several object instances. Fig. 10(a) shows a large image containing several golf courses. The object regions are delineated with white borders. Fig. 10(b) shows the detected golf course regions following the application of the two-layered texture motif model for golf courses. Similarly, Fig. 11(a) shows a large image containing several harbors. Fig. 11(b) shows the detected harbor regions using the texture motif model for harbors. It can be observed in both cases that most of the object regions are reliably isolated.

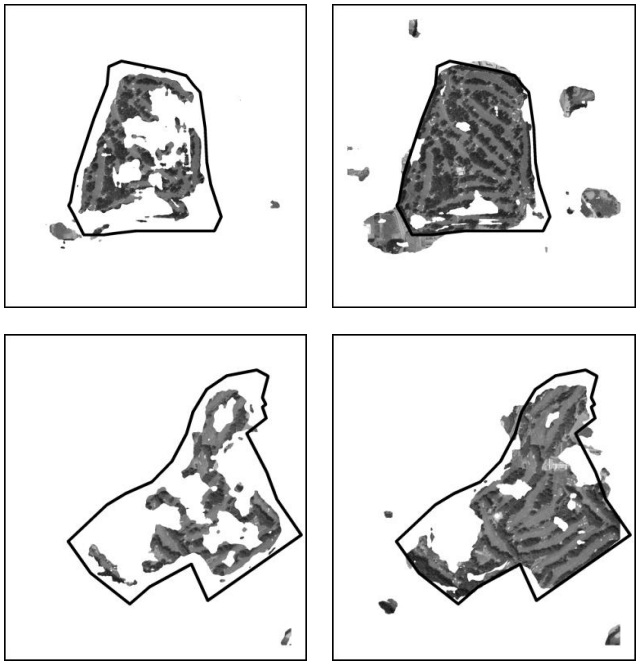
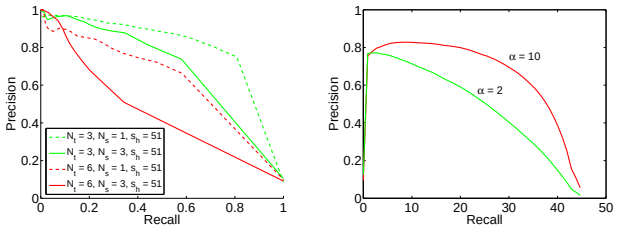


Fig. 7. The left column shows the detected golf course regions using the threshold t_o^* chosen from Fig. 6(b) for $\alpha = 10$ (with $N_t = 6$, $N_s = 3$, and $s_h = 161$). The right column shows the detected golf course regions for $\alpha = 2$ (with $N_t = 6$, $N_s = 3$, and $s_h = 161$). The borders of the desired object regions are marked in black.



(a) (b)

Fig. 8. (a) Precision-recall curves with different modeling parameters for the harbor dataset, and (b) F-measure $\mathcal{F}_\alpha(t_o)$ for different α computed using the harbor dataset with modeling parameters $N_t = 3$, $N_s = 1$, and $s_h = 51$.

B. Results on Pruning Large Datasets

Geographic databases such as the Alexandria Digital Library (ADL) Gazetteer [1] index the locations of several object types, including harbors, golf courses, and airports. However, instances of these objects are currently manually located and indexed. The manual labor involved in this process could be greatly reduced by applying model-driven approaches for automatically identifying probable locations of objects. This results in the elimination of many areas that, with high probability, do not contain the object. The resulting *pruned* dataset is much smaller than the original, making it much easier for manual verification of object presence.

The object modeling and detection approach presented in this paper is very useful in this regard. In the following, we shall demonstrate that our approach is capable of significantly pruning a dataset of images while looking for an object. For this purpose, we choose four objects: harbors, golf courses, housing colonies, and parking lots. Examples of these objects

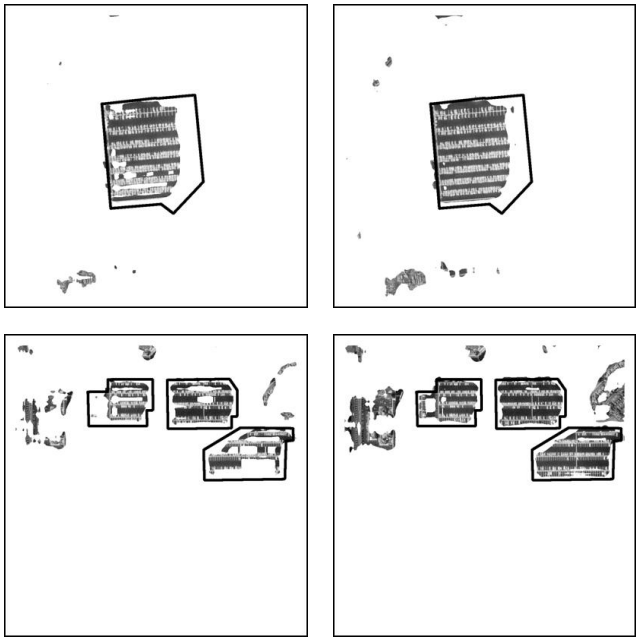
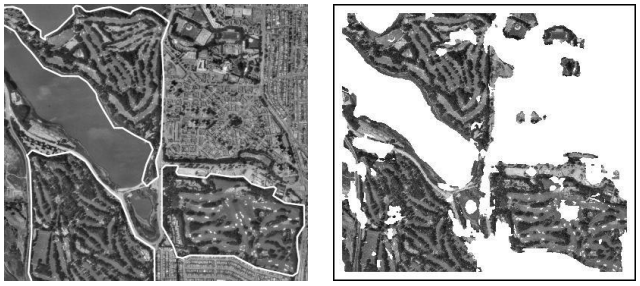


Fig. 9. The left column shows the detected harbor regions for $\alpha = 10$ (with $N_t = 3$, $N_s = 1$, and $s_h = 51$). The right column shows the detected harbor regions for $\alpha = 2$ (with $N_t = 3$, $N_s = 1$, and $s_h = 51$). The borders of the desired object regions are marked in black.



(a) (b)

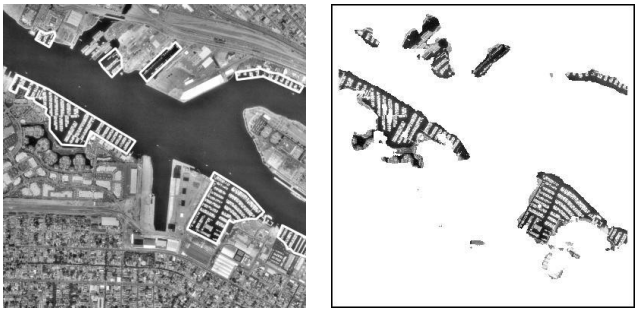
Fig. 10. (a) A large geospatial image containing several golf course regions (denoted with white borders); (b) The detected golf course regions with the application of the two-layered texture motif model.

are shown in Fig. 1. In the case of the last two objects, only the parameters of the selected model are mentioned and the results of detailed evaluation of different models are not presented.

Methodology:

The images are divided into tiles of size $N_{\rm ts} \times N_{\rm ts}$ pixels, with an overlap of $N_{\rm ol}$ pixels between adjacent tiles. For a given object, groundtruth information is created by labeling each of these tiles as 1 or 0 depending on whether the object is present in the tile or not. Let T_i be the $i^{\rm th}$ tile in the dataset, with $G_i \in \{0, 1\}$ denoting its groundtruth label. Suppose we apply an object detection algorithm on tile T_i, and get the "decision" $D_i \in \{0, 1\}$. In other words, $D_i = 1$ if an object is detected by the algorithm in tile T_i, and $D_i = 0$ if not.

We demonstrate the performance of the detection method by plotting the fraction of *false alarm* tiles (false alarm rate) with that of *missed* tiles (miss rate) from the dataset. T_i is a *false alarm* tile if $G_i = 0$ and $D_i = 1$, i.e. an object is detected when in fact it is not present in the tile. T_i is a *missed* tile if



(a) (b)

Fig. 11. (a) A large geospatial image containing several harbor regions (denoted with white borders); (b) The detected harbor regions with the application of the two-layered texture motif model.

$G_i = 1$ and $D_i = 0$, i.e. an object is not detected but it does exist in the tile. Both false alarms and misses are undesirable. Pruning a dataset is often a tradeoff between the two.

Let $\mathit{fmiss}(t_o)$ and $\mathit{ffa}(t_o)$ denote the fraction of missed and false alarm tiles respectively for a threshold t_o on the confidence measure, $p_s(\mathbf{h}(\mathbf{x}))$ in (14). These are computed as

$$\begin{aligned} \mathit{fmiss}(t_o) &= \frac{1}{N}\sum_i \left(1 - D_i(t_o)\right) G_i, \text{ and} \\ \mathit{ffa}(t_o) &= \frac{1}{N}\sum_i D_i(t_o)(1 - G_i), \end{aligned} \tag{20}$$

where N is the total number of tiles in the dataset. $D_i(t_o) = 1$ if an object is detected at tile T_i given the threshold t_o. In our experiments, we set $D_i(t_o) = 1$ if at least $D_{\rm min}$ pixels in T_i have a confidence measure greater than t_o. $D_{\rm min}$ is set at 200 pixels for both golf courses and harbors. In practice, if a tile T_i has $D_i(t_o) = 1$, then we set the detection labels of the neighboring tiles to also be 1. This is done in order to avoid tiles being missed on account of their overlapping a small portion of the object. If a neighboring tile contains a larger portion of the object, its confidence is inherited by the current one. The "false alarm vs. missed" plot for an object is obtained by varying t_o and recording the corresponding values of $\mathit{ffa}(t_o)$ and $\mathit{fmiss}(t_o)$.

Results:

For harbors and golf courses, the dataset consists of large aerial images (of resolution 1 m/pixel) from the ADL DOQQ collection, each with a typical size close to 7500×6600 pixels. $N_{\rm ts}$ is set to 1024 pixels and $N_{\rm ol} = 512$ pixels. For housing colonies, the dataset consists of large aerial images (of 1-m/pixel resolution) of the Santa Barbara region, taken from airplanes. These images have dimensions close to 5000 pixels. A similar dataset is used in the case of parking lots, except that the images are of 0.5 m/pixel resolution. For the latter two objects, $N_{\rm ts} = 128$ and $N_{\rm ol} = 0$.

Fig. 12(a) shows the false alarm vs. missed plot in the case of the golf courses. A total of 157 DOQQs were considered, of which 20 contained one or more golf courses. Among the 22530 resulting tiles, 350 are given a groundtruth label of 1 since they overlap a golf course. The diagonal line connecting the 1's denotes the expected plot for a random detection decision, i.e. the worst possible plot. A plot that dips close

to the origin is considered good since, at certain thresholds, a low rate is obtained for both false alarms and misses. From Fig. 12(a), it can be seen that no golf course tiles are missed at a false alarm rate of 43.22%. In other words, 56.78% of the tiles are eliminated without missing any golf course tiles. However, if we relax the acceptable miss rate to 14.57%, then the false alarm rate drops to 25.51%.

Fig. 12(b) shows the false alarm vs. missed plot in the case of the harbors. A total of 214 DOQQs were considered, of which 24 contained one or more harbors. Among the 30765 resulting tiles, 313 are given a groundtruth label of 1 since they overlap a harbor. From Fig. 12(b), it can be seen that no harbor tiles are missed at a false alarm rate of 56.17%. In other words, 43.83% of the tiles are eliminated without missing any harbor tiles. However, if we relax the acceptable miss rate to 9.46%, then the false alarm rate drops to 16.29%.

Fig. 12(c) shows the false alarm vs. missed plot in the case of the housing colonies. The parameters of the selected *housing colony* model used in this experiment, are $N_t = 6$, $N_s = 1$, and $s_h = 51$. A total of 44 aerial images were considered and all barring four contained housing colonies. Among the 54501 resulting tiles, 6442 are given a groundtruth label of 1. The missrate is very close to zero (0.42%) at a false alarm rate of 60.26%, which means that about 40% of the tiles are eliminated with a negligible number of missed tiles. However, if we relax the acceptable miss rate to 10.76%, then the false alarm rate drops to 17.81%.

Fig. 12(d) shows the false alarm vs. missed plot in the case of the *parking lot* object. The parameters of the selected *parking lot* model used in this experiment, are $N_t = 8$, $N_s = 3$, and $s_h = 61$. A total of 4 aerial images were considered, all of which contain at least one parking lot. Among the 4900 resulting tiles, 243 are given a groundtruth label of 1. It can be seen that no parking lot tiles are missed at a false alarm rate of 38.33%. In other words, 61.67% of the tiles are eliminated without missing any parking lot tiles. However, if we relax the acceptable miss rate to 15.64%, then the false alarm rate drops to 21.69%.

Thus an effective pruning of large datasets is achieved, reducing the manual labor involved in ascertaining the presence and location of geospatial objects.

VIII. Conclusion and Future Work

This paper introduces the concept of texture motifs enabling model-driven detection of geospatial objects. Texture motifs of an object are spatially recurrent patterns that are characteristic to the object. Such spatial patterns can be observed in geospatial objects such as golf courses, harbors, and airports. Detection of an object then reduces to detecting one or more of its texture motifs. This is done by learning an appearance model for texture motifs from object examples. Object models based on texture motifs provide a powerful alternative to shape-based and edge-based models, which are prohibitively expensive to compute, due to the level of complexity and detail often found in geospatial objects.

The second contribution of this paper is a semi-supervised framework for learning a two-layered model for texture motifs of an object from examples. The first layer learns the local intensity variations in the motif that form textural elements such as flat areas, bars, edges, and so on. These can be interpreted as the low-level building blocks of the motif. This layer is learned by clustering Gabor filter outputs sampled from the object examples in a rotation-invariant manner. The second layer of the representation learns the spatial distribution of low-level texture elements in the motif, since this influences its distinct visual appearance. A Gaussian mixture model (GMM) for this is learned from examples using features derived from histograms of texture elements in spatial neighborhoods. Confidence measures generated using this model are then used for detecting object presence.

The quality of the models are evaluated on the basis of their application to object detection. Experimental results demonstrate that such a modeling approach is quite effective in detecting complex geospatial objects. We illustrate the usefulness of our approach in reducing the manual labor involved in identifying object locations in large aerial image datasets.

Finally, it should be observed that though texture is an important feature in object detection, it is by no means the only one. The combination of texture with other features, such as color and shape, should increase the robustness of object detection. Knowledge-guided segmentation schemes [40] could be explored as a means of combining different features and models, with the goal of improving both the reliability and precision of object detection.

References

[1] L. Hill, J. Frew, and Q. Zheng, "Geographic names: The implementation of a gazetteer in a georeferenced digital library," *D-Lib Magazine*, January 1999.

[2] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 131–152, February 1988.

[3] R. B. Irvin and D. M. McKeown, "Methods for exploiting the relationship between buildings and their shadows in aerial imagery," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1564–1575, November 1989.

[4] J. Shufelt and D. M. McKeown, "Fusion of monocular cues to detect man-made structures in aerial imagery," *Computer Vision, Graphics and Image Processing*, vol. 57, no. 3, pp. 307–330, May 1993.

[5] S. Noronha and R. Nevatia, "Detection and modeling of buildings from multiple aerial images," *Pattern Analysis and Machine Intelligence*, pp. 501–518, May 2001.

[6] A. L. Reno and D. M. Booth, "Using models to recognise man-made objects," in *IEEE Workshop on Visual Surveillance*, 1999.

[7] A. Gruen, O. Kubler, and P. Agouris, Eds., *Automatic extraction of man-made objects from aerial and space images (I)*. Birkhauser, Basel, 1995.

[8] A. Gruen, E. P. Baltsavias, and O. Henricsson, Eds., *Automatic extraction of man-made objects from aerial and space images (II)*. Birkhauser Verlag, 1997.

[9] E. P. Baltsavias, A. Gruen, and L. V. Gool, Eds., *Automatic extraction of man-made objects from aerial and space images (III)*. A. A. Balkema, 2001.

[10] M. Mueller and K. Segl, "Object recognition based on high spatial resolution panchromatic satellite imagery," in *Joint workshop of ISPRS on Sensors and Mapping from Space*, 1999.

[11] M. Mueller, K. Segl, and H. Kaufmann, "Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery," *Pattern Recognition*, vol. 37, pp. 1619–1628, August 2004.

[12] M. Nagao and T. Matsuyama, *A structural analysis of complex aerial photographs*. Advanced Applications in Pattern Recognition, Plenum Publishing, 1980.

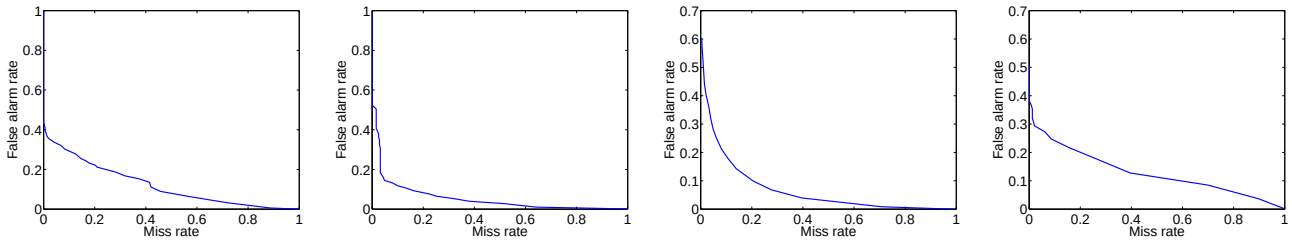


Fig. 12. The false alarm vs. missed plots for detecting (a) golf courses, (b) harbors, (c) housing colonies, and (d) parking lots, in large aerial image datasets.

[13] B. Nicolin and R. Gabler, "A knowledge-based system for the analysis of aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 25, no. 3, pp. 317–329, 1987.

[14] F. Quint, *Recognition of structured objects in monocular aerial images using context*. Mapping buildings, roads and other man-made structures from images, Ed. F. Leberl. Mnchen, 1997, pp. 213–228.

[15] J. David M. McKeown, J. Wilson A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 5, pp. 570–585, September 1985.

[16] J. David M. McKeown, W. A. Harvey, and L. E. Wixson, "Automating knowledge acquisition for aerial image interpretation," *Computer Vision, Graphics and Image Processing*, vol. 46, pp. 37–81, 1989.

[17] T. Matsuyama and V. Hwang, *SIGMA: A knowledge-based aerial image understanding system*. Advances in computer vision and machine intelligence, Plenum Press, 1990.

[18] S. T. F. Mahmood, "Attentional selection in object recognition," Ph.D. dissertation, MIT, Cambridge, 1993.

[19] R. N. Braithwaite and B. Bhanu, "Hierarchical Gabor filters for object detection in infrared images," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1994, pp. 628–631.

[20] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognition*, vol. 30, no. 2, pp. 295–309, February 1997.

[21] C. Schmid, "Constructing models for content-based image retrieval," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001, pp. 39–45.

[22] J. S. Weska, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 4, pp. 269–285, 1976.

[23] J. Carr, "Spectral and textural classification of single and multiple band digital images," *Computers & Geosciences*, vol. 22, pp. 849–865, 1996.

[24] C. Zhu and X. Yang, "Study of remote sensing image texture analysis and classification using wavelet," *International Journal of Remote Sensing*, vol. 19, no. 16, pp. 3197–3203, 1998.

[25] S. Berberoglu, C. Lloyd, P. Atkinson, and P. Curran, "The integration of spectral and textural information using neural networks for land cover mapping in the Mediterranean," *Computers & Geosciences*, vol. 26, pp. 385–396, 2000.

[26] T. Wassenaar, J. Robbez-Masson, P. Andrieux, and F. Baret, "Vineyard identification and description of spatial crop structure by per-field frequency analysis," *International Journal of Remote Sensing*, vol. 23, no. 17, pp. 3311–3325, 2002.

[27] T. Ranchin, B. Naert, M. Albuisson, G. Boyer, and P. Astrand, "An automatic method for vine detection in airborne imagery using wavelet transform and multiresolution analysis," *Photogrammetric Engineering & Remote Sensing*, vol. 67, no. 1, pp. 91–98, 2001.

[28] V. Karathanassi, C. Iossifidis, and D. Rokos, "A texture-based classification method for classifying built areas according to their density," *International Journal of Remote Sensing*, vol. 21, no. 9, pp. 1807–1823, 2000.

[29] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[30] S. Newsam, L. Wang, S. Bhagavathy, and B. S. Manjunath, "Using texture to analyze and manage large collections of remote sensed image and video data," *Journal of Applied Optics: Information Processing*, vol. 43, no. 2, pp. 210–217, January 2004.

[31] T. Quack, U. Monich, L. Thiele, and B. Manjunath, "Cortina: A system for large-scale, content-based web image retrieval," in *ACM Multimedia*, October 2004.

[32] W. Y. Ma and B. S. Manjunath, "Netra: a toolbox for navigating large image databases," *Multimedia Systems*, vol. 7, no. 3, pp. 184–198, May 1999.

[33] W.-Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs," *Journal of the American Society for Information Science*, vol. 49, no. 7, pp. 633–48, May 1998.

[34] S. Bhagavathy, S. Newsam, and B. S. Manjunath, "Modeling object classes in aerial images using texture motifs," in *Proceedings of the International Conference on Pattern Recognition*, August 2002.

[35] R. Manduchi, "A cluster grouping technique for texture segmentation," in *Proceedings of the International Conference on Pattern Recognition*, 2000, pp. 1060–1063.

[36] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proceedings of the International Conference on Visual Information Systems*, 1999, pp. 509–516.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. Series B, 39, no. 1, pp. 1–38, 1977.

[38] S. Newsam and B. S. Manjunath, "Normalized texture motifs and their application to statistical object modeling," in *CVPR Workshop on Perceptual Organization in Computer Vision (POCV)*, June 2004.

[39] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, 1979.

[40] B. Sumengen, S. Bhagavathy, and B. S. Manjunath, "Graph partitioning active contours for knowledge-based geospatial segmentation," in *Proceedings of the IEEE CVPR Workshop on Perceptual Organization in Computer Vision*, June 2004, pp. 54–54.