3D Texture Classification Using the Belief Net of a Segmentation Tree

Sinisa Todorovic and Narendra Ahuja
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, U.S.A.
{sintod, ahuja}@vision.ai.uiuc.edu

Abstract

This paper presents a statistical approach to 3D texture classification from a single image obtained under unknown viewpoint and illumination. Unlike in prior work, in which texture primitives (textons) are defined in a filter-response space, and texture classes modeled by frequency histograms of these textons, we seek to extract and model geometric and photometric properties of image regions defining the texture. To this end, texture images are first segmented by a multiscale segmentation algorithm, and a universal set of texture primitives is specified over all texture classes in the domain of region geometric and photometric properties. Then, for each class, a tree-structured belief network (TSBN) is learned, where nodes represent the corresponding image regions, and edges, their statistical dependecies. A given unknown texture is classified with respect to the maximum posterior distribution of the TSBN. Experimental results on the benchmark CUReT database demonstrate that our approach outperforms the state-of-the-art methods.

1. Introduction

Textured surfaces in natural scenes are usually characterized by variations in local height, color and reflectance, and hence referred to as 3D texture. Analysis of images of 3D texture is a challenging task, since different lighting and viewing conditions give rise to significant changes in texture appearance, due to, for example, shadowing, foreshortening, and occlusion, as illustrated in Fig. 1.

Several recent studies on texture have addressed the dependence of texture appearances on imaging conditions [2–5, 9]. In [3, 4], parametric models based on surface roughness and correlation lengths have been developed for classification of textures in the Columbia-Utrecht (CUReT) database, which contains texture images over a wide range of systematic changes in illumination and viewpoint. Further, in [5], a universal set of textons (texture primitives) and their frequency histogram have been proposed to address

Figure 1. Sample 4 from CUReT database [3]: under various imaging conditions the images seem to represent different surfaces.

3D effects. Textons have been defined as cluster centers of filter responses over a stack of images with representative viewpoints and lighting. However, in their approach, a set of registered images with known imaging parameters of the same unknown texture must be presented for classification. Two similar approaches have been proposed in [2, 9], where 2D textons are extracted as cluster centers in filter-response space, while their frequency histogram is expressed as a vector function of imaging parameters. Thereby, they have accomplished a computationally simpler texture representation, capable of classifying single images without any *a priori* information, unlike in [5].

Our approach draws from prior work the ideas to build a universal set of primitives, and to learn their joint distribution. Also, in this paper, we build a series of models for each texture class over a set of images parameterized by varying illumination and viewpoints. Here, to reduce the number of models per class, we employ the standard K-Medoid algorithm, following the approach in [9]. In the classification stage, a given unknown texture, obtained under unknown viewing and lighting directions, is recognized with respect to the maximum posterior distribution of the learned texture models.

The twofold novelty of our approach stems from the domain in which we define texture primitives, and from the specification of their joint distribution. Unlike in prior work, where texture features are extracted by a bank of preselected filters, we seek to capture geometric and photometric properties defining the texture, in unsupervised manner. To this end, we perform a multiscale segmentation of an

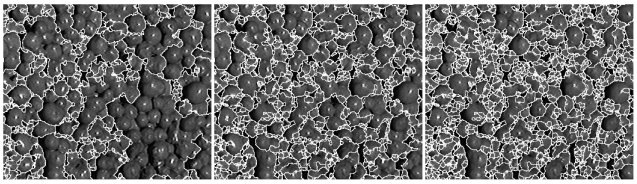

Figure 2. Sample 35 from CUReT: marked regions are nodes of the segmentation tree at levels 8, 9 and 10, respectively.

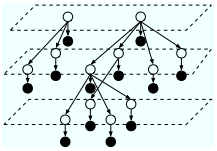

Figure 3. TSBN has irregular structure of the segmentation tree; black nodes represent observables, and white, hidden variables.

image by using an algorithm discussed in [1, 7], which outputs a segmentation tree. The segmentation tree contains all segmentations that can be identified in the image, corresponding to all different degrees of saliency, e.g., defined as color homogeneity. Nodes at upper levels correspond to more salient regions, whereas any cutset of the tree provides a 2D layout of the segmented regions, as illustrated in Fig. 2. Each node of the segmentation tree is characterized by a feature vector that includes geometric and photometric properties of the corresponding region–namely, region area, boundary shape, and color mean and variance. The number of tree levels and the homogeneity values associated with them, as well as the number of children of each node are a priori unknown, and are dynamically determined by the image at hand.

The segmentation tree serves as a rich description of the image for deriving texture models, which in this paper comprises two stages. First, the segmented regions in the training images of all texture classes are clustered in the aforementioned feature space of geometric and photometric properties. Then, a texture primitive is specified as a vector containing the mean and variance of the feature vectors of regions in a cluster. These texture primitives form a finite universal dictionary of texture "words" characterizing all texture classes. Note that supervision in prior work, with respect to pre-selecting an optimal filter bank, is eliminated in this paper by the segmentation algorithm, which dynamically determines the optimal domain of texture primitives.

In the second modeling stage, for each texture class, we build a tree-structured belief network (TSBN), depicted in Fig. 3. TSBNs are very popular statistical models in image processing and computer vision [6, 8]. The TSBN of an image consists of hidden and observable random variables organized in the same structure as that of the corresponding segmentation tree of the image. Observables are the feature vectors of geometric and photometric properties of the corresponding regions in the segmentation tree, and are mutually independent given their corresponding hidden variables. Hidden variables are labels of the texture primitives, specified in the first modeling stage, while connections between them represent parent-child statistical dependencies. Note that unlike histograms in prior work the TSBN captures spatial dependencies among texture regions. Furthermore, the ascendant-descendant (Markovian) connections in the TSBN encode the statistical properties of pixel neighborhoods of varying size. All this makes TSBNs more expressive models than frequency histograms used in prior work.

The joint distribution of hidden and observable variables fully characterizes the TSBN model of a given texture class, and allows for texture classification within the Bayesian framework, i.e., with respect to the maximum posterior distribution, computed here by the standard belief propagation algorithm [6]. Experiments of texture classification are presented on 20 samples from the CUReT database [4] . The results demonstrate that our approach offers a viable solution to 3D texture classification.

2. Observables and Texture Primitives

In this paper, images are represented by segmentation trees [1, 7], where each region (node) i is associated with a feature vector, $\boldsymbol{y}_i$, comprising the intrinsic geometric and photometric properties of region i. Let $\boldsymbol{\mu}_i$ and Σ_i^2 denote the mean and covariance of region i color values. Also, let A_i denote the region area. To describe the boundary shape of i, we parse the image into L pie slices, each of which begins at the centroid of i, and subtends the the same angle $2\pi/L$. Next, we compute the normalized histogram $\boldsymbol{h}_i=\{h_i(l)\}_{l=1}^{L}$, of the number of pixels of region i that fall in pie slice l. Clearly, the region feature vector, specified as $\boldsymbol{y}_i = [\boldsymbol{\mu}_i, \Sigma_i, A_i, \boldsymbol{h}_i]$, can be easily extended, as dictated by the requirements of a particular application. These feature vectors represent observable random variables in the TSBN.

In the first stage of learning, segmented regions of the training images of all texture classes are clustered by the standard K-Means algorithm in the feature space determined by $\boldsymbol{y}_i$ values. The K-Means produces K clusters, $\{C_k\}_{k=1}^{K}$, each of which defines the associated texture primitive. A texture primitive, π_k, is specified as a vector containing the mean and variance of the feature vectors of regions in a cluster, $\pi_k = [\text{mean}(\{\boldsymbol{y}_i\}_{i\in C_k}), \text{var}(\{\boldsymbol{y}_i\}_{i\in C_k})]$.

3. TSBNs, Model Reduction and Classification

In this paper, texture class $\mathcal{T}$ is modeled with a TSBN parameterized over viewpoints $\mathcal{V}$ and illumination $\mathcal{I}$. The TSBN is fully specified by its joint distribution of hidden, $X=\{x_i\}$, and observable, $Y=\{\boldsymbol{y}_i\}$ random variables, $\forall i \in T$, where i denotes a node in the segmentation tree T.

A hidden variable, x_i, represents the label of a texture primitive. The label of node i is conditioned on the label of its parent j, and is specified by the conditional probability tables, $P(x_i|x_j)$. The joint probability of X of a given texture class $\mathcal{T}$ is specified as

$$P(X|\mathcal{T},\mathcal{V},\mathcal{I}) = \prod_{i,j\in T} P(x_i|x_j,\mathcal{T},\mathcal{V},\mathcal{I}) \ , \qquad (1)$$

where for the roots we use priors $P(x_i|\mathcal{T},\mathcal{V},\mathcal{I})$. Since we assume that observables $\boldsymbol{y}_i$ are conditionally independent given the corresponding x_i, the joint likelihood of Y can be expressed as

$$P(Y|X,\mathcal{T},\mathcal{V},\mathcal{I}) = \prod_{i\in T} P(\boldsymbol{y}_i|x_i{=}k,\mathcal{T},\mathcal{V},\mathcal{I}) \ , \qquad (2)$$

where $P(\boldsymbol{y}_i|x_i{=}k,\cdot)$ is modeled as the Gaussian distribution with parameters encoded in the texture primitive π_k. It follows that the TSBN for texture $\mathcal{T}$ and imaging parameters $\mathcal{V}$ and $\mathcal{I}$ is fully characterized by

$$P(X,Y|\mathcal{T},\mathcal{V},\mathcal{I}) = \prod_{i,j\in T} P(\boldsymbol{y}_i|x_i,\cdot)P(x_i|x_j,\cdot) \ . \qquad (3)$$

The parameters of likelihoods $P(\boldsymbol{y}_i|x_i,\cdot)$ can be learned by using the ML algorithm over the clusters $\{C_k\}_{k=1}^{K}$, obtained in the first modeling stage. Next, the transition probabilities, $P(x_i|x_j,\cdot)$, can be learned by the standard belief propagation algorithm [6, 8], the details of which are omitted for space reasons. Despite the irregular structure of the TSBN, the computational complexity of the belief propagation is polynomial in time, since its structure is known and equal to that of the segmentation tree.

Note that in the above formulation the number of models per class is the same as the number of training images that differ in $\mathcal{V}$ and $\mathcal{I}$ parameters. For the purposes of texture classification, this represents a modeling redundancy, since even considerable variations in texture appearance of one class may not reduce the classification accuracy if the other classes are sufficiently different from it. To reduce the number of models per texture class, we employ the standard K-Medoid algorithm [9]. In particular, the set of $P(X|\mathcal{T},\mathcal{V},\mathcal{I})$ values, over all texture classes $\mathcal{T}$ and parameters $\mathcal{V}$ and $\mathcal{I}$, may be clustered by the K-Medoid into M clusters, and represented by M cluster centers. The update rule of the K-Medoid always moves the cluster center to the nearest data point in the cluster, but does not integrate over the points as the K-Means algorithm. Indeed, in the K-Medoid, the cluster centers are always data points themselves. Therefore, a selected cluster center can be uniquely identified as an individual $P(X|\mathcal{T},\mathcal{V},\mathcal{I})$ point, which, in turn, determines the most representative TSBN model with $\mathcal{V}$ and $\mathcal{I}$ values. Note that the outlined procedure yields a different number of representative models per texture class.

To classify an unknown image, we select the texture class, $\hat{\mathcal{T}}$, for which the posterior distribution $P(\mathcal{T},\mathcal{V},\mathcal{I},X|Y)$ is maximum:

$$\hat{\mathcal{T}} = \arg\max_{\substack{\mathcal{T},\mathcal{V},\\ \mathcal{I},X}} P(\mathcal{T},\mathcal{V},\mathcal{I},X|Y) \approx \arg\max_{\substack{\mathcal{T},\mathcal{V},\\ \mathcal{I},X}} P(X,Y|\mathcal{T},\mathcal{V},\mathcal{I}). \qquad (4)$$

In Eq. (4), the prior $P(\mathcal{T},\mathcal{V},\mathcal{I})$ is assumed uniform over all possible values of $\mathcal{T}$, $\mathcal{V}$, and $\mathcal{I}$. Moreover, $P(Y)$ is assumed a smooth, slow changing function, which seems reasonable considering our premise that texture appearances undergo considerable variations, i.e., a wide range of Y values are equally likely.

Thus, in the classification stage, a given image is first segmented to obtain Y values, then $P(X,Y|\mathcal{T},\mathcal{V},\mathcal{I})$ values are computed using the belief propagation for all M representative models of all texture classes, and, finally, the image is classified as in Eq. (4).

4. Experiments

For experimental validation of our approach, we use the CUReT database [4], which contains images of 61 real-world 3D textures, each imaged under 205 different combinations of viewing and illumination directions. As in [2, 9], the same subset of 20 textures is selected–specifically, samples: 1, 4, 6, 10, 12, 14, 16, 18, 20, 22, 25, 27, 30, 33, 35, 41, 45, 48, 50, and 59. Out of the existing 205 images per class, only 92 are chosen that contain a sufficiently large texture region (whose viewing angle is less then 60 degrees). These 92 images are then manually cropped, to ensure that they contain only texture information, and randomly divided into two distinct sets of 46 training and 46 test images. The classification results are averaged over a set of 5 experiments, each conducted for different random partitioning of images into the training and test set. Each experiment consists of four stages: (i) Segmentation of $46{\times}20{=}920$ training images and finding their segmentation trees, (ii) Generation of the dictionary of texture primitives, (iii) Learning TSBNs from the training images and reducing the total number of the learned models, and (vi) Classification of 920 test images. For describing the boundary shape of a segmented region, we use $L = 40$ histogram bins.

Fig. 4 presents the classification results over a range of values for the number of texture primitives, K, for the average number of representative models per class, and for the number of available uniformly sampled training images per class from the training set. For comparison, when one parameter is varied, the other two are fixed to the same values

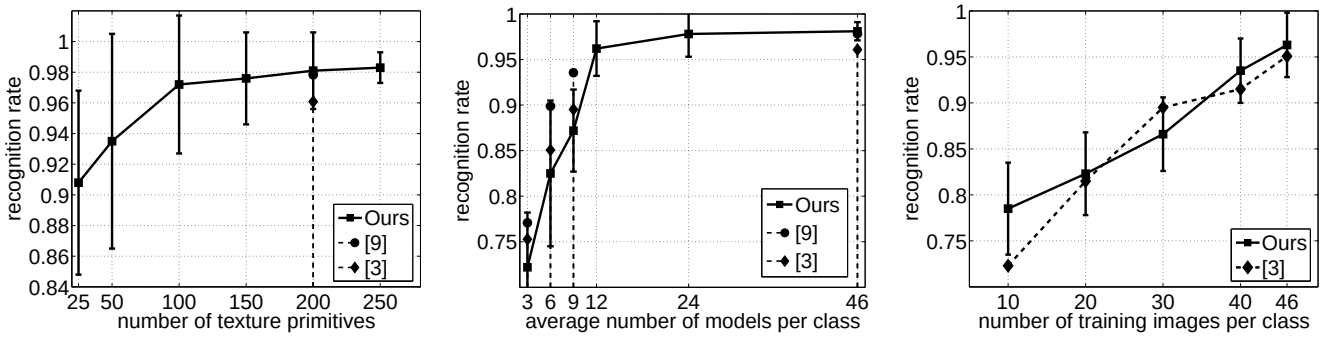

Figure 4. Contrasting our averaged global classification results for all 20 texture classes with those reported in [2,9]; (left) 46 models per class on average and 920 training images; (middle) 920 training images and 200 texture primitives; (right) 46 models per class on average and 100 texture primitives.

as reported in [2,9], and given in the caption of Fig. 4. When 46 models per class on average, and 200 texture primitives are learned from 920 training images, our recognition rate averaged over 5 experiments and over 20 classes is 98.09%, outperforming 97.83% in [9], and 96.08% in [2]. From Fig. 4, a small number of models per texture class (less than 12) renders our approach inferior to both [2] and [9]; however, when that number is sufficiently large (above 24), our recognition rate exceeds those of [2, 9]. In the right plot of Fig. 4, the global recognition rate increases as the number of training images per class becomes larger, where, interestingly, ours is greater than the one reported in [2] for a very small training set. This suggests that TSBNs are capable of capturing more statistically significant information from a few training images than frequency histograms used in [2].

5. Conclusion

The appearance of 3D texture varies significantly as the viewpoint and lighting directions change, and must be explicitly accounted for, in order to accomplish a reasonable classification accuracy on a single sample with unknown imaging parameters. To this end, in this paper, an optimal set of TSBNs per texture class is learned from training images, represented as segmentation trees, and indexed by the most significant viewpoints and illumination. The presented experimental results obtained on the CUReT database demonstrate that our approach yields higher recognition rates than the state-of-the-art methods in the same experimental settings.

Two key factors, proposed in this paper, lead to the improved classification performance. First, feature extraction is done by the multiscale segmentation algorithm, which dynamically finds segmentations at all saliency scales present in the image, instead of using a pre-selected filter bank, as in previous work. Second, TSBNs capture spatial dependencies among texture regions of varying size, which makes them more expressive than frequency histograms used in prior work.

Acknowledgment

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE TPAMI*, 18(12):1211–1235, 1996.

[2] O. G. Cula and K. J. Dana. 3D Texture recognition using bidirectional feature histograms. *Int. J. Comput. Vision*, 59(1):33–60, 2004.

[3] K. J. Dana and S. Nayar. Correlation model for 3D texture. In *ICCV*, volume 2, pages 1061–1067, 1999.

[4] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, 1999.

[5] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, 2001.

[6] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*, chapter 4, pages 143–236. Morgan Kaufamnn, San Mateo, 1988.

[7] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Trans. Image Processing*, 6(5):642–655, 1997.

[8] S. Todorovic and M. C. Nechyba. Dynamic trees for unsupervised segmentation and matching of image regions. *IEEE TPAMI*, 27(11):1762–1777, 2005.

[9] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62(1-2):61–81, 2005.