Texel-based Texture Segmentation

Sinisa Todorovic
School of EECS
Oregon State University
sinisa@eecs.oregonstate.edu

Narendra Ahuja
Beckman Institute
University of Illinois at Urbana-Champaign
n-ahuja@uiuc.edu

Abstract

Given an arbitrary image, our goal is to segment all distinct texture subimages. This is done by discovering distinct, cohesive groups of spatially repeating patterns, called texels, in the image, where each group defines the corresponding texture. Texels occupy image regions, whose photometric, geometric, structural, and spatial-layout properties are samples from an unknown pdf. If the image contains texture, by definition, the image will also contain a large number of statistically similar texels. This, in turn, will give rise to modes in the pdf of region properties. Texture segmentation can thus be formulated as identifying modes of this pdf. To this end, first, we use a low-level, multiscale segmentation to extract image regions at all scales present. Then, we use the meanshift with a new, variable-bandwidth, hierarchical kernel to identify modes of the pdf defined over the extracted hierarchy of image regions. The hierarchical kernel is aimed at capturing texel substructure. Experiments demonstrate that accounting for the structural properties of texels is critical for texture segmentation, leading to competitive performance vs. the state of the art.

1. Introduction

Images generally consist of distinct parts representing different surfaces in the scene. Surfaces made of an optically homogeneous material and under smoothly varying illumination give rise to image parts with a smooth variation of image brightness, referred to as non-texture subimage. In contrast, surfaces with discontinuities in depth, material, illumination, etc., give rise to texture subimages. The spatial variation of intensity in each image part contains different information about the scene, often requiring different types of algorithms to be invoked on these parts. Depending on its purpose, an algorithm may apply to either texture or non-texture subimages, or both. For example, if a surface in the scene gives rise to the texture subimage it may be better to estimate the surface's shape by shape-from-texture algorithms than by shape-from-shading methods. Also, in order for image smoothing and compression algorithms to be applicable to the entire image they should account for differences among textured and non-textured image parts, and adjust their parameters accordingly.

This paper is about texture segmentation, i.e., delineating the boundaries of all distinct texture subimages in an arbitrary image. The discovered boundaries will automatically delineate the remaining, non-texture image parts. If all distinct texture and non-texture image parts have been identified, then the decision as to which higher-level algorithms need to be invoked on which parts will become easier.

A texture may be coarse. In this case, it is characterized by a spatial repetition of texture elements – 2D patterns, called texels – within the image area occupied by the texture [11, 2, 15, 28, 16, 18, 20, 12, 19, 3]. Alternatively, the texture surface in the scene and imaging conditions may be such that the texels appear with pixel or subpixel sizes. In this case, the texture subimage has fine granularity with a relatively high degree of pixel-level noise. This makes such fine-granularity texture subimages appear more similar to non-texture image parts than to coarse-texture ones. In this paper, we focus on segmenting distinct coarse-texture subimages, where pixels form local, distinct clumps corresponding to texels. After identifying these textures, the remaining parts of the image will correspond to non-texture and fine-granularity-texture subimages.

The Problem: Texels in natural scenes, in general, are not identical, and their spatial repetition is not strictly periodic. Instead, texels are only statistically similar to one another, and their placement along a surface is only statistically uniform. Also, texels are not homogenous-intensity regions, but may contain hierarchically embedded structure. Therefore, image texture can be characterized by a probability density function (pdf) governing the (natural) statistical variations of the following texel properties: (1) geometric and photometric – referred to as intrinsic properties (e.g., color, area, shape); (2) structural; and (3) relative orientations and displacements of texels – referred to as placement.

The Rationale: Given an image, suppose we used a low-level, multiscale segmentation to identify homogeneous-intensity regions at all photometric scales present. A subset of these regions may be texels, subtexels or groups of texels, while some other regions may not be part of any texture. Each region is characterized by certain intrinsic properties, and spatial layout properties relative to other regions. These properties can be used to define a descriptor vector of each region. If the underlying joint pdf of these descriptors contains a mode, this means that the image contains many statistically similar regions that belong to that mode. In turn, if these similar regions belong to a certain, cohesive subimage, then, by definition, they can be interpreted as texels, and the subimage can be taken as texture. It follows that detection of texture subimages can be formulated as identifying modes of the pdf of region descriptors. Since the pdf is defined in terms of regions, detecting the mode simultaneously identifies texels, and the corresponding texture subimage. The rest of the image consists of other homogeneous regions that do not belong to any coarse texture, and is said to be non-texture and fine-granularity texture image parts.

Contributions: As discussed in the next section, most related work assumes that textures have deterministic texels with nearly periodic placement, and that size and placement of texels are uncorrelated [18, 12, 21, 19]. A few approaches allow statistical variations in texel properties, but using restrictive models and computationally intensive inference algorithms [30]. In contrast, we do not make any assumptions about the functional form of intrinsic and placement properties of texels. *Both appearance and placement of texels are allowed to be stochastic and correlated.* For such textures, we propose unsupervised texture segmentation based on identifying the pdf modes of image regions. Most prior work on pdf mode detection does not account for hierarchical relationships between data points [26, 8, 7]. However, since texels typically contain substructure, capturing structural properties of regions is critical for identifying texture. To detect the pdf modes, we propose to modify and use the meanshift algorithm [8, 7]. Unlike the original, our new formulation is able to explicitly account for any presence of hierarchical embedding of subtexels within the texels. To this end, we define a new hierarchical kernel in terms of region-subregion hierarchical properties. The new kernel also has a locally-varying bandwidth. This variable bandwidth is estimated by partitioning the feature space of region descriptors into Voronoi polytopes. To the best of our knowledge, texture segmentation under relatively unrestricted assumptions about statistical properties of texels, and the mentioned novel aspects of the structure-extracting meanshift have never been reported in the literature.

Overview of Our Approach: The block diagram of our approach is shown in Fig. 1. (1) A multiscale segmentation algorithm is used to extract homogeneous-intensity regions at all photometric scales present. The multiscale segmentation does not impose any constraints on region shapes, sizes, contrasts, and topological contexts. Each region thus obtained is characterized by a descriptor vector of intrinsic and spatial-layout properties (e.g., color, area, shape, location). (2) The meanshift is used to estimate modes of the pdf over the region descriptors. For the meanshift, we derive a new hierarchical kernel with locally adaptive bandwidth. The bandwidth is estimated via binning the feature space of region descriptors. The bins represent Voronoi polytopes around each descriptor. (3) The set of descriptors (i.e., regions) visited by all meanshift procedures converging to one of the identified modes, automatically delineates the associated texture subimage.

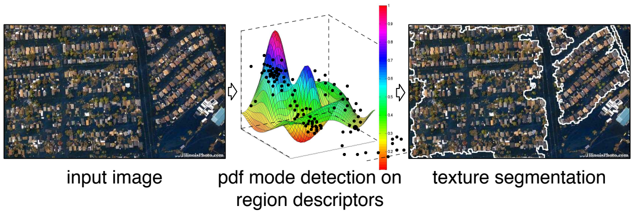


Figure 1. Our approach: A low-level segmentation partitions the image into regions, each characterized by a descriptor vector of region properties. The descriptors are viewed as samples from an unknown pdf. Modes of the pdf are identified using a new meanshift that explicitly accounts for structural properties of regions. This amounts to texture segmentation (our results marked white).

The paper is organized as follows. Sec. 2 reviews prior work. Sec. 3 describes the feature space of region properties. Sec. 4 specifies our pdf mode estimation. Experiments and concluding remarks are presented in Sec. 5 and Sec 6.

2. Relationships to Prior Work

This section reviews prior work in texture modeling and segmentation. Methods that explicitly encode texel properties in their models of texture typically use the following texel representations: (a) salient blobs [28, 5]; (b) interest points within texels [18, 12, 21]; (c) combination of interest points and Canny edges [25]; or (d) user-specified templates and filter functions [17, 30, 27, 19]. In contrast, we use segments that facilitate delineating the exact texel boundaries, and thus accurate identification of texel regions. Julesz and his colleagues [15] have argued the use of special features called textons (e.g., closure, line endpoints, corners) for texture modeling. There have been several attempts to mathematically define the notions of textons and texels. In [30], for example, texture is modeled as a superposition of Gabor base functions which are generated by a user-specified vocabulary of texton templates. The region-based, hierarchical texel model of [3] encodes only the intrinsic properties of texels. In contrast to this approach, we additionally consider the modeling of texel placement properties.

Regarding texel placement, most approaches make the assumption that texels form a (near-)regular lattice in the image[12, 19, 20]. They identify texel locations by searching for the most likely affine transform that may have distorted the ideal 2D lattice of texels. Some methods allow random placement of texels, based on random tessellations [24], or based on representing texels as points and then tracking similar points within heuristic pixel neighborhoods [17]. Work on modeling the placement of random closed sets in a plane (e.g., objects in the image) as the Poisson process, i.e., the "dead leaves" model (DL) [2, 16] is also related. However, texel orientations and relative displacements typically have spatially larger dependencies than the Poisson process, being memoryless, is capable of capturing. In addition, the DL regards the pdf of object sizes and the pdf of object locations in the image as being independent, which may not be, in general, justified for modeling texture.

Most prior work on texture segmentation does not account for intrinsic and spatial layout properties of texels. For example, filter-based statistical methods typically compute filter responses over heuristically defined pixel neighborhoods. This, in turn, may lead to mixing the statistics of neighboring texture segments [13, 29], or confusing strong responses in the direction of the boundary edge with the texture response [22]. This is because they perform neighborhood operations, whose results, in general, vary with the location of the neighborhood within the texel and with respect to the texel boundary. Consequently, filter-based texture segmentation often results in missing or hallucinating texture segments. Adaptive-scale filtering [6] only partially solves this problem, because some textures cannot be characterized by a single scale, especially when texels themselves are textured, as commonly encountered in real images. A multiscale aggregation of filter responses and shape elements [10], or hierarchical clustering of image segments and texture descriptors[9] reduces, but does not eliminate the aforementioned issues. Texture segmentation using active contours requires that at least one image texture must be present in the image [14], which we here relax.

3. The Feature Space of Region Properties

This section presents our Step 1. Since, in general, texels are not homogenous-intensity regions, but contain hierarchically embedded structure, their representation should be hierarchical [3]. Access to such image structure is provided by a strictly hierarchical, multiscale segmentation algorithm that partitions the image over a range of photometric scales (i.e., contrasts) [1, 4]. At each scale, the pixel-intensity variations within each region are smaller than those across the region boundary at each scale. The algorithm guaranties that regions obtained at lower contrasts will strictly merge into larger regions as the photometric scale increases. Therefore, the output of the segmenter is a hierarchy of recursively embedded smaller regions within the larger ones, called segmentation tree. Note that the segmenter produces regions of various sizes. Since texels cannot occupy relatively large image areas (otherwise they will not occur in large numbers), we discard all large-size regions ($> 50\%$ of image size). The remaining regions are our basic image features corresponding to subtexels, texels, groups of texels, textures, and other subimages.

A descriptor vector of properties $\boldsymbol{x}_i$ is associated with each image region i. We define $\boldsymbol{x}_i$ to contain the following intrinsic and spatial-layout properties: 1) average contrast across i's boundary; 2) area, excluding the total area of i's embedded children regions; 3) standard deviation of i's children areas; 4) displacement vector between the centroids of i and its parent region; 5) perimeter of i's boundary; 6) aspect ratio of the intercepts of i's principal axes with the i's boundary, where the principal axes are estimated by standard ellipse fitting to i, i.e., as eigenvectors of the covariance matrix of the second central shape moments of i; 7) orientation, measured as the angle between the major principal axis of i and the x-axis of the image; and 8) (x,y) coordinates of the centroid of i. The descriptors of all regions in the image are input to the standard PCA, and the most representative (95% accuracy) subspace is used as the feature space of region descriptors.

Unlike in [3], where the objective is texel recognition despite changes in scale and orientation, our goals are different, and thus we specify the region descriptor in terms of properties that are not scale and rotation-in-plane invariant. Specifically, we seek to segment a single image based on distinct textures present, whose perception and discrimination critically depend on a degree of texel variations. Any invariance to scale and rotation may in general lead to confusing distinct textures as being identical.

The descriptors, $\boldsymbol{x}_i$, i=$1, \ldots, N$, define data points in the feature space of region properties. Given N descriptors, our objective is to estimate modes of their underlying pdf, $f(\boldsymbol{x})$.

4. Voronoi-based Binned Meanshift

This section presents our Step 2 that introduces two modifications to the meanshift: (i) Variable-bandwidth Gaussian kernel, and (ii) New hierarchical kernel that uses (i). These modifications will allow us to explicitly account for structural properties of texels, which is beyond the scope of the original meanshift formulation [8, 7]. We begin by reviewing the meanshift algorithm.

4.1. Technical Rationale

The meanshift procedure starts from a random point in the feature space, $\boldsymbol{y}_1$, and then visits a sequence of points $\{\boldsymbol{y}_t\}$, t=$1, 2, \ldots$, where $\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \boldsymbol{m}(\boldsymbol{y}_t)$, and $\boldsymbol{m}(\boldsymbol{y}_t)$ is the meanshift vector pointing along the density gradient. The meanshift procedure is usually run in parallel, starting simultaneously from many data points. The sequence $\{\boldsymbol{y}_t\}$ is shown to converge to a stationary point. All points in the feature space visited by the meanshift along the trajectories toward the local maximum are taken to belong to the corresponding pdf mode.

One of the limitations of the original meanshift formulation is that it uses the fixed bandwidth kernel, chosen so as to globally balance the estimator bias and variance. Due to the sparseness of data in higher dimensions, however, multivariate neighborhoods are generally empty, particularly in the "tails" of the density. As the dimension increases, larger bandwidths are necessary to balance the estimator bias and variance. This in turn has negative effects of over-smoothing the density near its modes. Varying the amount of smoothing is widely regarded as a suitable solution in feature spaces with low to moderate dimensions. The meanshift can be improved by using a multivariate, sample-point estimator [8, 7]

$$\hat{f}_S(\boldsymbol{x}) = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} K_{\boldsymbol{H}_i}(\boldsymbol{x}-\boldsymbol{x}_i), \tag{1}$$

where $\boldsymbol{H}_i \triangleq \boldsymbol{H}(\boldsymbol{x}_i)$ is a varying smoothing matrix associated with each sample point $\boldsymbol{x}_i$, and where $K_{\boldsymbol{H}_i}(\boldsymbol{x}-\boldsymbol{x}_i)$ is a Gaussian kernel with mean $\boldsymbol{x}_i$ and covariance $\boldsymbol{H}_i$. In [8], $\boldsymbol{H}_i$ is defined to be isotropic, and only a function of the true density at $\boldsymbol{x}_i$, $\boldsymbol{H}_i \propto f(\boldsymbol{x}_i)^{-1/2}\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, while many other characteristics of $f(\boldsymbol{x})$ (e.g., curvature) are ignored. Also, the estimate of $f(\boldsymbol{x})$ is not readily available, since the meanshift estimates the gradient of $f(\boldsymbol{x})$. In [7], statistically stable Gaussian-mixture modes are estimated using a range of fixed bandwidths.

While we retain the assumption that $K_{\boldsymbol{H}_i}(\cdot)$ is Gaussian, in the sequel, we derive a new expression for the optimal $\boldsymbol{H}_i$ that is anisotropic, unlike in [8], and we relax the assumption of [7] that a mode is Gaussian. We also derive a hierarchical Gaussian kernel for capturing hierarchical relationships between region descriptors.

4.2. The New Variable-Bandwidth Matrix

Binned sample-point estimator (BSPE): We partition the feature space, defined in Sec 3, in a number of bins, B_j, $j{=}1,\ldots,M$, with volume $|B_j|$. Then, we compute a representative of each bin, $\boldsymbol{b}_j$, and use BSPE:

$$\hat{f}_B(\boldsymbol{x}) = \tfrac{1}{N}\textstyle\sum_{j=1}^{M} n_j K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j), \tag{2}$$

where $\boldsymbol{H}_j \triangleq \boldsymbol{H}(\boldsymbol{b}_j)$, $K_{\boldsymbol{H}_j}(\cdot)$ is Gaussian, and n_j is the number of region descriptors in jth bin. Before we derive the expression for $\boldsymbol{H}_j$, we first show that our BSPE based meanshift converges to a stationary point. Note that an estimate of the gradient of $f(\boldsymbol{x})$ is the gradient of $\hat{f}_B(\boldsymbol{x})$

$$\Delta\hat{f}_B(\boldsymbol{x}) = \tfrac{1}{N}\textstyle\sum_{j=1}^{M} n_j \boldsymbol{H}_j^{-1}(\boldsymbol{b}_j-\boldsymbol{x})K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j). \tag{3}$$

Following the derivation steps presented in [7], we introduce the auxiliary mean bandwidth matrix

$$\bar{\boldsymbol{H}}(\boldsymbol{x})^{-1} \triangleq \textstyle\sum_{j=1}^{M} \frac{n_j K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j)}{\sum_{j'=1}^{M} n_{j'} K_{\boldsymbol{H}_{j'}}(\boldsymbol{x}-\boldsymbol{b}_{j'})} \boldsymbol{H}_j^{-1}\ , \tag{4}$$

which immediately gives the expression of the variable-bandwidth meanshift vector (see [7] for details)

$$\boldsymbol{m}_B(\boldsymbol{x}) \triangleq \bar{\boldsymbol{H}}(\boldsymbol{x})\Delta\hat{f}_B(\boldsymbol{x})/\hat{f}_B(\boldsymbol{x})\ . \tag{5}$$

From (5), the magnitude of $\boldsymbol{m}_B(\boldsymbol{x})$ becomes larger where data are scarce (i.e. tails and valleys), and smaller near modes, as desirable. Since $K_{\boldsymbol{H}_j}(\cdot)$ is Gaussian with a convex and monotonic decreasing profile, from the theorem presented in [8], it immediately follows that the meanshift procedure, explained in Sec. 4.1, that uses our $\boldsymbol{m}_B(\boldsymbol{x})$, given by (5), converges to a stationary point.

The following theorem states how to compute the optimal anisotropic matrices $\boldsymbol{H}_j$, given a partitioning of the feature space into M bins, B_j, $j{=}1,\ldots,M$. This result will be used in (2)-(5) to run the meanshift. We derive $\boldsymbol{H}_j$, under the assumption that $f(\boldsymbol{x})$ can be approximated by a function that is piece-wise constant in each bin. This assumption seems reasonable if the bins are sufficiently small in the areas of the feature space with high values of $f(\boldsymbol{x})$. As we will show, our partitioning of the feature space satisfies this condition.

Theorem: *If we are given: a partition of the feature space into bins B_j, $j{=}1,\ldots,M$, representative vectors $\boldsymbol{b}_j$ of each bin, and a neighborhood system of the bins, $\eta(j)=\{i : (\boldsymbol{b}_i,\boldsymbol{b}_j){\in}Neighbors\}$, and if $\int_{B_j} f(\boldsymbol{x})d\boldsymbol{x} \approx f(\boldsymbol{b}_j)|B_j|$, then the optimal $\boldsymbol{H}_j$ used in (2) and (3), is given by*

$$\boldsymbol{H}_j = \sum_{i\in\eta(j)} \frac{3f(\boldsymbol{b}_i)|B_i|}{f(\boldsymbol{b}_j)|B_j| + \displaystyle\sum_{i'\in\eta(j)} f(\boldsymbol{b}_{i'})|B_{i'}|}(\boldsymbol{b}_i-\boldsymbol{b}_j)(\boldsymbol{b}_i-\boldsymbol{b}_j)^T, \tag{6}$$

Proof: See Appendix.

The above theorem states that $\boldsymbol{H}_j$ of jth bin should be computed as a weighted mean of the covariances between $\boldsymbol{b}_j$ and the representatives of neighboring bins $\boldsymbol{b}_i$, $i{\in}\eta(j)$. It can be shown that when using $\boldsymbol{H}_j$ given by (6) the estimation bias decreases and the estimation covariance remains the same in comparison with the case when a fixed-bandwidth kernel is used.

Consistent with our assumption that the bins are sufficiently small in the feature-space areas of high density, we expect very small variations in $f(\boldsymbol{x})$ across the neighboring bins. This allows us to simplify the expression in (6) as

$$\boldsymbol{H}_j \approx \sum_{i\in\eta(j)} \frac{3|B_i|}{|B_j| + \sum_{i'\in\eta(j)}|B_{i'}|}(\boldsymbol{b}_i-\boldsymbol{b}_j)(\boldsymbol{b}_i-\boldsymbol{b}_j)^T. \tag{7}$$

We use the expression in (7) to compute $\boldsymbol{H}_j$ for every bin $j{=}1,\ldots,M$, which is then plugged in (2)-(5) in order to run the meanshift procedure, explained in Sec. 4.1. Next, we explain our approach to partitioning the feature space into bins which satisfy the conditions of the above theorem.

Definition of bins: To partition the feature space, we use the Voronoi diagram of region descriptors $\{\boldsymbol{x}_i\}$, $i{=}1,\ldots,N$, defined in Sec. 3. The Voronoi diagram associates with each descriptor $\boldsymbol{x}_i$ a polytope B_i which is defined by all points $\boldsymbol{x}$ in the feature space closer to $\boldsymbol{x}_i$ than to any other point $\boldsymbol{x}_j$, $j{\neq}i$, $B_i{\triangleq}\{\boldsymbol{x}{:}\boldsymbol{x}{\in}\mathbb{R}^d, \forall j{\neq}i, \|\boldsymbol{x}_j{-}\boldsymbol{x}\|{>}\|\boldsymbol{x}_i{-}\boldsymbol{x}\|\}$. Thus, for any nondegenerate distribution of data, the Voronoi diagram tessellates the feature space into a set of polytopes B_i, each containing exactly one of the descriptors. Two Voronoi polytopes that share a part of their boundaries are called neighbors. For N descriptors, complexity of computing the Voronoi diagram is $O(N\log N)$.

For our purposes, the Voronoi polytopes are a good choice to define bins in the feature space, since they capture the global layout of and mutual relationships between the region descriptors. For example, size of the Voronoi polytopes is large in areas where the descriptors are sparse, and, conversely, size of the polytopes is small in densely populated areas. Also, the Voronoi diagram provides a natural definition of the neighborhood system of the bins, which does not require any thresholds on distances between the points, or any other input parameters.

In this paper, the representative of each bin is equal to the descriptor generating that bin, $\boldsymbol{b}_i = \boldsymbol{x}_i$, $i = 1,\ldots,N$ ($M{=}N$). Note that this does not make equations (1) and (2) equal, since they use different bandwidth matrices. Specifically, (2) uses the optimal, anisotropic bandwidth, defined by the Voronoi neighborhood system.

4.3. The Hierarchical Kernel

In the previous section, we have shown that the use of the Gaussian kernel with the optimal $\boldsymbol{H}_j$, given by (6), yields the meanshift procedure that converges. Below, we define a hierarchical kernel, $K^h_{\boldsymbol{H}_j}(\cdot)$, which can be used in the meanshift instead of the Gaussian one, $K^g_{\boldsymbol{H}_j}(\cdot)$. The motivation for using a hierarchical kernel is that texels, in general, are not homogenous-intensity regions, but may contain hierarchically embedded subregions. Therefore, the use of $K^h_{\boldsymbol{H}_j}(\cdot)$ may yield a more accurate estimation of modes of $f(\boldsymbol{x})$. Since region descriptors represent image regions, we can define hierarchical relationships between the descriptors based on the embedding of corresponding smaller regions within larger regions in the image. Formally, each descriptor $\boldsymbol{x}_i$ defines a tree-structured graph of descriptors $\boldsymbol{x}_k$ corresponding to subregions k embedded within region i in the image. These hierarchical relationships can be extended between any two arbitrary points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the feature space, which do not correspond to any particular regions in the image. To this end, we use our Voronoi partitioning of the feature space. Specifically, suppose $\boldsymbol{x}$ belongs to B_i, and $\boldsymbol{x}'$ belongs to B_j, then the hierarchical relation between any $\boldsymbol{x}$ and $\boldsymbol{x}'$ is equivalent to that between the descriptors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. This extension allows us to use the hierarchical kernel, $K^h_{\boldsymbol{H}_j}(\cdot)$, on all points in the feature space.

Given that $\boldsymbol{x}$ is in bin B_i, we compute $K^h_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{x}_j)$ by finding the maximum subtree isomorphism between two trees rooted at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

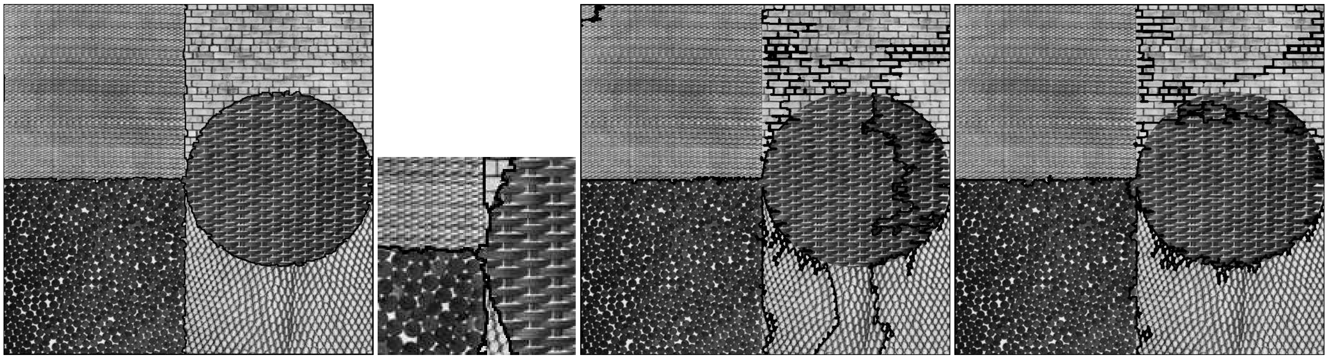
$$K^h_{\boldsymbol{H}_j}(\boldsymbol{x}{-}\boldsymbol{x}_j){\triangleq}K^g_{\boldsymbol{H}_j}(\boldsymbol{x}{-}\boldsymbol{x}_j)\,\underbrace{\max_{\mathcal{M}_{ij}}\prod_{(k,l)\in\mathcal{M}_{ij}}K^g_{\boldsymbol{H}_l}(\boldsymbol{x}_k{-}\boldsymbol{x}_l)}_{\substack{\| \\ \min_{\mathcal{M}_{ij}}\sum_{(k,l)\in\mathcal{M}_{ij}}(\boldsymbol{x}_k{-}\boldsymbol{x}_l)^T\boldsymbol{H}_l^{-1}(\boldsymbol{x}_k{-}\boldsymbol{x}_l).}},\tag{8}$$

where mapping $\mathcal{M}_{ij}$ includes the matching descendant subregions k and l embedded within their respective ascendant regions i and j in the image. To compute (8), we use the standard tree matching algorithm described in [3], whose complexity is $O(n^2)$ in the number of nodes n in trees.

Since $K^h_{\boldsymbol{H}_j}(\cdot)$ is computed as a product of Gaussians, it is straightforward to see that $K^h_{\boldsymbol{H}_j}(\cdot)$ has a convex and monotonic decreasing profile. From the theorem presented in [8], it immediately follows that the meanshift procedure, explained in Sec. 4.1, that uses $K^h_{\boldsymbol{H}_j}(\cdot)$ in (2)-(5) converges to a stationary point.

5. Experimental Evaluation

This section presents our quantitative and qualitative evaluation on four datasets: (1) 100 collages of randomly mosaicked, 111 distinct Brodatz textures, where each texture occupies at least 1/6 of the collage (Fig. 2); (2) 180 collages of randomly mosaicked, 140 distinct Prague textures from 10 thematic classes (e.g., flowers, plants, rocks, textile, wood, etc.), where each texture occupies at least 1/6 of the collage (Fig. 3, 4); (3) 100 Aerial-Produce images, where 50 aerial images show housing developments, agricultural fields, and landscapes (Fig. 5), and 50 images show produce aisles in supermarkets (Fig. 5); (4) Berkeley segmentation dataset (Fig. 6, 7, fig:Comparison1). Datasets (1) and (2) provide ground truth texture segmentations. The texture mosaics of both datasets (1) and (2) are challenging for segmentation, because they contain complex layout topologies of subimages occupied by texture (e.g., boundaries of several regions meet at one point). The Prague dataset verifies our performance over a wide range of texture types, imaged under variations in scale, rotation, and illumination. Aerial-Produce and Berkeley present many well-known challenges of real images. Quantitative evaluation on Berkeley dataset is impossible, since its annotation



(a) Our results (b) Detail from (a) (c) Results of [8] (d) Results of [7]

Figure 2. Segmentation results on a collage of Brodatz textures. The identified texture boundaries are marked black and overlaid on the original image. The algorithms of [8] and [7] use variable bandwidth kernels, but do not account for structural properties of regions. In contrast, we do so by using the hierarchical kernel. The comparison on this collage suggests that this is a critical factor for successful texture segmentation. We succeed in delineating the texture boundaries even when several boundaries meet at a point.

Table 1. Unsupervised texture segmentation on Prague dataset

	CS	OS	US	ME	NE
[8]	15.13%	23.87%	10.58%	50.31%	51.36%
[9]	56.37%	11.93%	19.79%	11.55%	10.29%
Our results	59.13%	10.89%	18.79%	10.45%	9.93%

is only for object segmentation. Other common datasets, e.g., KTH-TIPS containing only random-pixel-field textures, CUReT containing only 3D textures, and PSU containing only near-regular textures, are aimed at testing different aspects of texture analysis that are beyond our scope.

Quantitative evaluation – Brodatz: Let G denote the area of true texture, and D denote the area of a subimage that our approach segments. Segmentation error per texture, ε, is estimated as $\varepsilon = \frac{\text{XOR}(G,D)}{\text{Union}(G,D)}$. Averaged over all mosaic parts, and over all 100 collages of Brodatz textures, we obtain $\bar{\varepsilon} = 93.3\% \pm 3.7$. Next, we evaluate $\bar{\varepsilon}$ when the meanshift kernel is not hierarchical, but simply Gaussian that uses the variable bandwidth matrix, given by (7), and immediate properties of the sample points. This tests the effect of explicitly accounting for structural properties of regions vs. ignoring them. When the kernel is not hierarchical, $\bar{\varepsilon}$ reduces to $77.9\% \pm 4.1$. Using the meanshift algorithm of [7], with a non-hierarchical kernel, in the same feature space, reduces $\bar{\varepsilon}$ to $62.3\% \pm 7.8$. This indicates that the Gaussian kernel that uses our variable bandwidth matrix, given by (7), gives better meanshift performance than the kernel presented in [7].

Quantitative evaluation – Prague: The standard metrics for evaluating texture segmentation on Prague dataset are: correct-, over-, and under-segmentation (CS, OS, US), and missed and noise error (ME, NE), among others. Using the above definitions of G and D, D is declared CS iff $G{\cap}D{\geq}0.75G$ and $G{\cap}D{\geq}0.75D$. OS (or US) counts every G (D) that is split into smaller regions D (G). ME (or NE) counts every G (D) that does not belong to CS, OS, and US. It is desired that CS is large and the remaining metrics small. In Tab. 1, we show comparison on Prague dataset with standard meanshift [8], and the state-of-the-art unsupervised texture segmentation [9]. The latter approach uses color and the standard covariance matrix as a texture descriptor. We do not use color information, but intensity contrasts. As can be seen, we outperform both approaches. All steps of our approach, starting from a low-level segmentation to texture segmentation, take about 5min for a $512{\times}512$ Prague texture mosaic, in MATLAB on a 3.1GHz, 2GB RAM PC.

Qualitative Evaluation: As can be seen in Figs. 2–8, we succeed in delineating texture boundaries even when they form complex-layout topologies. Filter-based methods, or approaches based on image decimation and smoothing would typically fail to accurately delineate topologically complex spots in which several texture boundaries meet. Our texture segmentation is also successful on real-world images shown in Figs. 5–7. For instance, despite using a low-level segmenter for feature extraction, our algorithm is not affected by abrupt changes in illumination or shadows (see the fish's fin in Fig. 6), because we use the intrinsic and placement properties of texels that are invariant to a wide range of local and global illumination changes. We illustrate comparison with [9] on Prague and Berkeley datasets in Figs. 3 and 8, and [10] on Berkeley dataset in Fig. 7. This comparison suggests that we produce better segmentations in terms of identifying perceptually more valid image textures. [10] does not report any quantitative results.

6. Conclusion

We have presented a texel-based approach to segmenting image parts occupied by distinct textures. This is done by capturing intrinsic and placement properties of distinct groups of texels. The scale or coarseness of texture is lower-

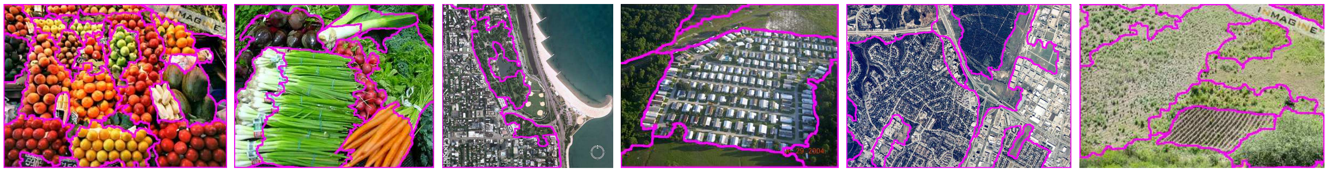


Figure 5. Aerial-Produce images: our texture segmentation results. Despite very similar colors of fruit on the leftmost image, and plants on the rightmost image, we succeed in identifying perceptually valid textures, because we account for texel placement and substructure. Color-based methods would find these examples challenging.

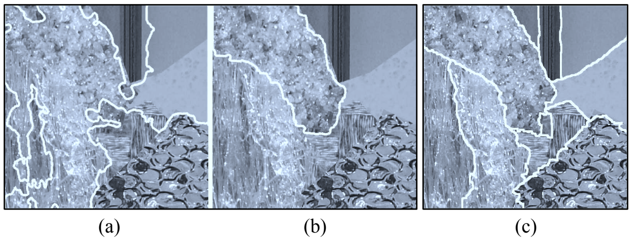


Figure 3. A mosaic from Prague dataset and segmentation results overlaid on the original image. The results are obtained using: (a) [9] without texture descriptors; (b) [9] with texture descriptors; and (c) our approach.

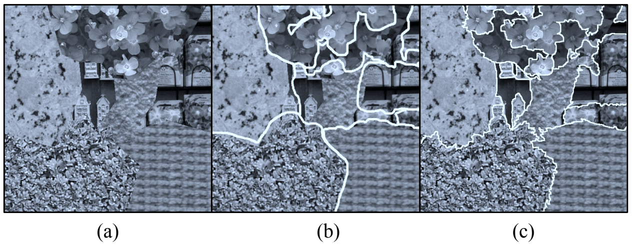


Figure 4. (a) A mosaic from Prague dataset. (b) Results obtained using [9] with texture descriptors. (c) Our results.

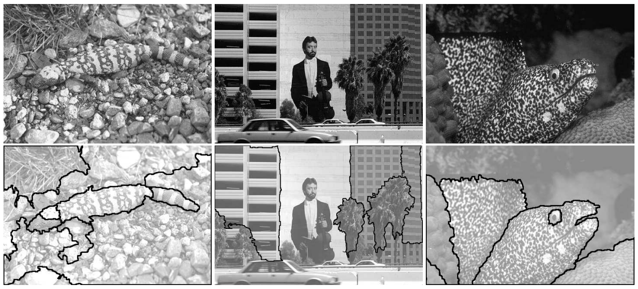


Figure 6. Examples from Berkeley dataset: the texture boundaries identified by our approach are marked black and overlaid on the original image.

bounded by the size of its texels. Since we define texels as regions, we do not address pixel- or subpixel-scale textures. Experimental evaluation on texture mosaics and real-world images suggests that capturing structural properties of texels is very important for texture segmentation. To account for texel substructure, we have derived and used a hierarchical, variable-bandwidth kernel in the meanshift.

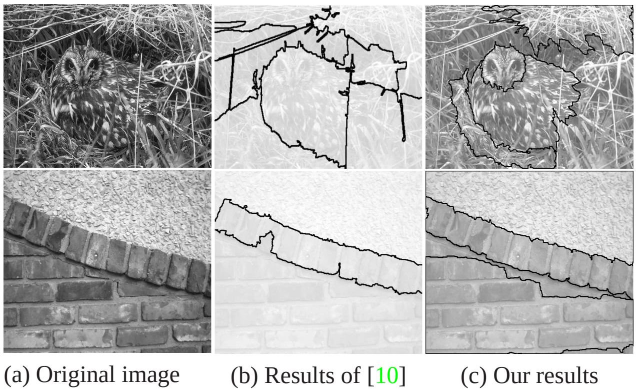


Figure 7. Comparison on Berkeley dataset with [10]. Our segmentation seems to yield more perceptually valid texture subimages.

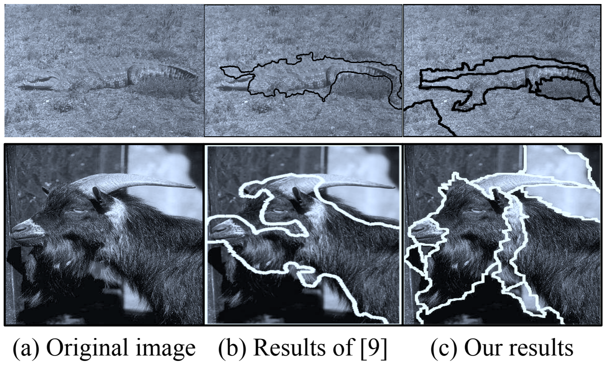


Figure 8. Comparison on Berkeley dataset with [9]. Our segmentation is successful on low-contrasted regions, because we account for all photometric scales present in the image.

Appendix

This section presents the proof of Theorem stated in Sec. 4. We derive $\boldsymbol{H}_j$, by minimizing the mean integrated squared error MISE $\triangleq E\{\int(\hat{f}_B(\boldsymbol{x})-f(\boldsymbol{x}))^2 d\boldsymbol{x}\}$ with respect to $\boldsymbol{H}_j$. We have:

$$\begin{aligned}\text{MISE}=&\tfrac{1}{N^2}\textstyle\sum_j E\{n_j\int K^2_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j)d\boldsymbol{x}\}\\ &+\tfrac{1}{N^2}\textstyle\sum_{i\neq j} E\{n_i n_j\int K_{\boldsymbol{H}_i}(\boldsymbol{x}-\boldsymbol{b}_i)K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j)d\boldsymbol{x}\}\\ &-\tfrac{2}{N}\textstyle\sum_j E\{n_j\int K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j)f(\boldsymbol{x})d\boldsymbol{x}\}+\int f(\boldsymbol{x})^2 d\boldsymbol{x},\end{aligned} \tag{9}$$

In (9), the only random variables are the numbers of data points in each bin, n_j, $j{=}1,\ldots,M$. They can be characterized by a multinomial distribution with parameters $p_j{\triangleq}\int_{B_j} f(\boldsymbol{x})d\boldsymbol{x}$, where B_j denotes jth bin. Since our kernel is Gaussian, following the derivation steps presented in [23], from (9) we obtain

$$\begin{aligned} \text{MISE}= & \tfrac{2}{N(2\pi)^{d/2}} \textstyle\sum_j [p_j(1-p_j)+Np_j^2]|\boldsymbol{H}_j|^{-1/2} \\ & +\tfrac{N-1}{N} \textstyle\sum_{i\neq j} p_i p_j K_{\boldsymbol{H}_i+\boldsymbol{H}_j}(\boldsymbol{b}_i-\boldsymbol{b}_j) \\ & -\tfrac{2}{N} \textstyle\sum_j p_j \int K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j) f(\boldsymbol{x}) d\boldsymbol{x} + \int f(\boldsymbol{x})^2 d\boldsymbol{x}. \end{aligned} \tag{10}$$

Next, we use the condition of the Theorem that $f(x)$ is piecewise constant within each bin B_j, $f(x) \approx \frac{p_j}{|B_j|}$, $\forall x \in B_j$. This affects only the third row of (10), $\int K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j) f(\boldsymbol{x}) d\boldsymbol{x} = \sum_i \frac{p_i}{|B_i|} \int_{B_i} K_{\boldsymbol{H}_j}(\boldsymbol{x}-\boldsymbol{b}_j) d\boldsymbol{x} \approx p_j K_{\boldsymbol{H}_j}(\boldsymbol{0}) = \frac{p_j |\boldsymbol{H}_j|^{-1/2}}{(2\pi)^{d/2}}$, yielding

$$\begin{aligned} \text{MISE}= & \tfrac{2}{N(2\pi)^{d/2}} \textstyle\sum_j [p_j+(N-2)p_j^2]|\boldsymbol{H}_j|^{-1/2} \\ & +\tfrac{N-1}{N} \textstyle\sum_{i\neq j} p_i p_j K_{\boldsymbol{H}_i+\boldsymbol{H}_j}(\boldsymbol{b}_i-\boldsymbol{b}_j) + \int f(\boldsymbol{x})^2 d\boldsymbol{x}, \end{aligned} \tag{11}$$

The optimal $\boldsymbol{H}_j$ can be found using the derivative of the asymptotic MISE (AMISE), when $N \to \infty$, as

$$\frac{\partial \text{AMISE}}{\partial \boldsymbol{H}_j} = \frac{-p_j^2 |\boldsymbol{H}_j|^{-3/2}}{(2\pi)^{d/2}} + p_j \sum_{i\neq j} p_i \frac{\partial K_{\boldsymbol{H}_i+\boldsymbol{H}_j}(\boldsymbol{b}_i-\boldsymbol{b}_j)}{\partial \boldsymbol{H}_j} = 0. \tag{12}$$

From (12) and the assumption that $K_{\boldsymbol{H}_i+\boldsymbol{H}_j}(\boldsymbol{b}_i-\boldsymbol{b}_j) \approx 0$ when B_i and B_j are not neighboring bins, we obtain (6)

$$\boldsymbol{H}_j = \frac{3\sum_{i\in\eta(j)} p_i (\boldsymbol{b}_i-\boldsymbol{b}_j)(\boldsymbol{b}_i-\boldsymbol{b}_j)^T}{p_j + \sum_{i\in\eta(j)} p_i}. \quad \square \tag{13}$$

Acknowledgement

The support of the National Science Foundation under grant NSF IIS 08-12188 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. 3

[2] N. Ahuja and B. J. Schachter. Image models. *ACM Comput. Surv.*, 13(4):373–397, 1981. 1, 3

[3] N. Ahuja and S. Todorovic. Extracting texels in 2.1D natural textures. In *ICCV*, 2007. 1, 2, 3, 5

[4] H. Arora and N. Ahuja. Analysis of ramp discontinuity model for multiscale image segmentation. In *ICPR*, volume 4, pages 99–103, 2006. 3

[5] D. Blostein and N. Ahuja. Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE TPAMI*, 11(12):1233–1251, 1989. 2

[6] G. Caenen, V. Ferrari, A. Zalesny, and L. Van Gool. Analyzing the layout of composite textures. In *Workshop on Texture Analysis in Machine Vision*, pages 15–20, 2002. 3

[7] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE TPAMI*, 25(2):281–288, 2003. 2, 3, 4, 6

[8] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. *ICCV*, 1:438, 2001. 2, 3, 4, 5, 6

[9] M. Donoser and H. Bischof. Using covariance matrices for unsupervised texture segmentation. In *ICPR*, 2008. 3, 6, 7

[10] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV*, pages 716–723, 2003. 3, 6, 7

[11] R. M. Haralick. Statistical and structural approaches to texture. *Proc. of the IEEE*, 67(5):786–804, 1979. 1

[12] J. H. Hays, M. Leordeanu, A. A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *ECCV*, volume 2, pages 522–535, 2006. 1, 2, 3

[13] T. Hofmann, J. Puzicha, J. M. Buhmann, and R. Friedrich. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE TPAMI*, 20:803–818, 1998. 3

[14] N. Houhou, J.-P. Thiran, and X. Bresson. Fast texture segmentation model based on the shape operator and active contour. *CVPR*, 2008. 3

[15] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981. 1, 2

[16] A. Lee, D. Mumford, and J. Huang. Occlusion models for natural images: A statistical study of a scale-invariant Dead Leaves model. *IJCV*, 41(1-2):35–59, 2001. 1, 3

[17] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001. 2, 3

[18] T. K. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *ECCV*, volume 1, pages 546–555, 1996. 1, 2

[19] W.-C. Lin and Y. Liu. A lattice-based MRF model for dynamic near-regular texture tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):777–792, 2007. 1, 2, 3

[20] Y. Liu, R. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE TPAMI*, 26(3):354–371, 2004. 1, 3

[21] A. Lobay and D. Forsyth. Shape from texture without boundaries. *IJCV*, 67(1):71–91, 2006. 2

[22] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. Comput. Vision*, 43(1):7–27, 2001. 3

[23] S. R. Sain. Multivariate locally adaptive density estimation. *Comput. Stat. Data Anal.*, 39(2):165–186, 2002. 8

[24] B. Schachter and N. Ahuja. Random pattern generation processes. *CGIP*, 10(2):95–114, 1979. 3

[25] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, volume LNCS 1681, pages 165–181, 1999. 2

[26] A. Touzani and J. G. Postaire. Mode detection by relaxation. *IEEE TPAMI*, 10(6):970–978, 1988. 2

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

[28] H. Voorhees and T. Poggio. Computing texture boundaries from images. *Nature*, 333:364–367, 1988. 1, 2

[29] L. Wolf, X. Huang, I. Martin, and D. Metaxas. Patch-based texture edges and segmentation. In *ECCV*, pages II: 481–493, 2006. 3

[30] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV*, 62(1-2):121–143, 2005. 2