

Unsupervised Detection and Localization of Structural Textures Using Projection Profiles

Ismet Zeki Yalniz¹, Selim Aksoy*

Department of Computer Engineering, Bilkent University, Ankara, 06800, Turkey

Abstract

The main goal of existing approaches for structural texture analysis has been the identification of repeating texture primitives and their placement patterns in images containing a single type of texture. We describe a novel unsupervised method for simultaneous detection and localization of multiple structural texture areas along with estimates of their orientations and scales in real images. First, multi-scale isotropic filters are used to enhance the potential texton locations. Then, regularity of the textons is quantified in terms of the periodicity of projection profiles of filter responses within sliding windows at multiple orientations. Next, a regularity index is computed for each pixel as the maximum regularity score together with its orientation and scale. Finally, thresholding of this regularity index produces accurate localization of structural textures in images containing different kinds of textures as well as non-textured areas. Experiments using three different data sets show the effectiveness of the proposed method in complex scenes.

Key words: Structural texture analysis, texture periodicity, textons, regularity detection, wavelet analysis

1. Introduction

Texture has been acknowledged to be an important visual feature used for classifying and recognizing objects and scenes. It can be characterized by textural primitives as unit elements and neighborhoods in which the organization and relationships between the properties of these primitives are defined. Haralick [1] defined texture as the uniformity, density, coarseness, roughness, regularity, intensity and directionality of discrete tonal features and their spatial relationships. He grouped the approaches for characterizing and measuring texture into two: *statistical* approaches like autocorrelation functions, transform methods, textural edgeness, and autoregressive models, and *structural* approaches that use the idea that textures are made up of primitives appearing in a near-regular repetitive arrangement.

Numerous applications of these approaches to image classification and object recognition exist in the literature. An important problem has been the definition and detection of textural primitives [2]. Most of the previous work have concentrated on statistical methods where pixels were used as the unit elements and features were extracted for pixel neighborhoods. These methods were mainly applied to the identification of stochastic textures or micro-textures where the texture primitives appeared at fine scales. The most widely studied statistical texture models involved the use of co-occurrence matrices [3], wavelets [4], Gabor filters [5, 6, 7, 8], Fourier transform [9, 10], histograms of filter responses [11, 12, 13], and Markov

random fields [14, 15]. Recent methods also included features extracted using local binary patterns [16, 17, 18] and covariance matrices [19]. The classification problem was usually defined as the identification of the texture class observed in a small patch that contained a single type of texture. The classification framework was also extended to include feature selection and to study invariance to rotation, scale, and illumination. However, the common choice for performance evaluation in most of the studies still involved the use of individual texture patches [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 18] or texture mosaics [7, 14, 17, 19] consisting of simple textures such as the ones in the Brodatz album.

Structural approaches, on the other hand, have aimed to model macro-textures where the texture primitives were distinguishable at coarser scales. The main goal of these approaches has been the identification of the texture primitives, also called texels or textons, and their placement patterns, also called lattice or grid layout, in a given structural texture. For example, Kim and Park [20] used projection profiles for a set of orientations to estimate parallelogram-shaped grid structures. Chetverikov and Haralick [21] used gray level difference statistics for anisotropy, symmetry, and regularity detection. Starovoitov et al. [22] extracted the displacement vectors of the lattice structure using the maxima of several features based on co-occurrence matrices computed at multiple orientations and scales for binarized images. Lin et al. [23] used the peaks of the autocorrelation function to identify candidate texture primitives, and applied the generalized Hough transform to find two displacement vectors from these peaks to generate the lattice structure. Liu et al. [24] extended this approach by defining a region of dominance for each peak in the autocorrelation function so that only the dominant peaks with no other peak within a certain neighborhood were used. Han et al. [25]

*Corresponding author. Tel: +90 312 2903405; fax: +90 312 2664047.

Email addresses: zeki@cs.umass.edu (Ismet Zeki Yalniz),
saksoy@cs.bilkent.edu.tr (Selim Aksoy)

¹Present address: Department of Computer Science, University of Massachusetts, Amherst, MA, USA

also generated hypotheses for the texture elements based on the peaks of the autocorrelation function of the image, and then used the Bayesian information criterion to select the best lattice according to its likelihood in the image and its complexity. As a frequency domain alternative, Charalampidis [26] used two fundamental frequencies obtained from the Fourier spectrum to identify the texture elements that form the lattice structure.

Such methods that exploit the global texture structure formed by repeating scene elements have been shown to produce good results when the free parameters were tuned for specific textures. However, an important common assumption and a very limiting setting in all of these approaches [20, 21, 22, 23, 24, 25, 26] was that the input image contained a single texture patch with an ideal (i.e., near-perfect) arrangement of the texture elements. Assuming that the input was an instance of a single structural texture, these methods concentrated on the identification of the repeating texture elements and their placement rules in a lattice.

Some approaches allowed some variation in the texture primitives and the placement patterns. For example, Leung and Malik [27] used the eigenvalues of the second moment matrix to identify distinctive scene elements with a large intensity variation, used the sum of squared differences criterion for matching neighboring patches after estimating an affine transform for the match, and propagated the growing procedure to neighboring patches using several thresholds. Hays et al. [28] identified texture elements using interest point detection and normalized cross correlation operators, found potential matches between pairs of neighboring texture elements, and iteratively refined the lattice structure by finding higher-order correspondences. Lin and Liu [29] required the user to provide the initial texel, and then used a Markov random field model with a lattice structure to model the topological relationships among the texels. However, all of these approaches [27, 28, 29] also assumed a single dominant texture in the image, and tried to estimate its model.

Even though a large body of literature on texture analysis exists with examples discussed above, automatic identification of structural textures and the quantification of their regularity in complex scenes still need to be explored further as these textures can be observed in a wide range of applications involving objects such as buildings, fences, walls, bricks in outdoor urban settings, fabrics, textiles, tiled floors, carpets, bookshelves in indoor settings, different kinds of materials in industrial vision applications, and artificially planted areas as opposed to natural vegetation in remotely sensed images. In general, most textures of man-made objects can be considered as regular whereas most natural textures can be considered as irregular [30].

This paper focuses on the detection of structural textures that are formed by texture primitives in a *near-regular* arrangement in real images. Extending the definition that regular textures refer to periodic patterns, near-regular textures involve certain amount of irregularity in both radiometric and geometric properties [31]. Unlike existing studies that try to classify texture patches or model the structure in an image that contains a single type of texture, we aim to obtain an accurate *localization* of multiple structural textures together with estimates of their *orientations* and *scales* in real images that exhibit many



Figure 1: Examples of structural textures, formed by near-regular arrangements of texture primitives, cropped from Google Earth images. We aim to obtain an accurate localization of such textures in complex scenes along with estimates of their orientations and scales in this paper.

different kinds of textures along with non-textured areas. Our model allows deformations in both the appearances of the texture primitives and the geometric properties such as local orientation and scale variations in their arrangements with examples shown in Figure 1.

The proposed approach starts with a pre-processing step involving a set of multi-scale isotropic filters for enhancing the texon-like objects in a grayscale image. We follow the distinction made between texels and textons by Hays et al. [28] that texels define a full partitioning (i.e., tiling) of the texture with each texel having a non-overlapping extent whereas textons are statistical features that are computed at every pixel without concern for overlap. Therefore, the local extrema in the filter responses are assumed to correspond to potential texton locations without any strict requirement for their exact detection (Section 2). The next step uses the observation that the locations of these extrema along a scan line with an orientation that matches the dominant direction of a regular structural texture also have a regular structure. Consequently, the existence of such regularity along a particular orientation at a particular scale is measured using projection profiles within oriented sliding windows where the image data in a window are converted into a 1D signal using the profile, and the regularity of the textons is quantified in terms of the periodicity of this profile using wavelet analysis (Section 3). The periodicity analysis of projection profiles is performed at multiple orientations and scales to compute a regularity score at each pixel for each orientation and scale (Section 4). Finally, a regularity index is computed for each pixel as the maximum regularity score and the principal orientation and scale for which this score is maximized by also requiring consistency of these scores among neighboring pixels for a certain range of orientations and scales (Section 5). The image areas that contain a structural texture composed of near-regular repetitive arrangements of textons can be localized by thresholding this regularity index.

The major contributions of this paper are as follows. We present a novel, unsupervised, multi-orientation and multi-scale

regularity analysis framework that uses wavelet analysis of projection profiles and results in a regularity index for each pixel along with estimates of the orientation and scale of the structure around that pixel. Thresholding of this regularity index produces an accurate simultaneous localization of multiple structural texture areas in real images containing different kinds of textures as well as non-textured areas even when no sharp boundaries exist in the image data. Experiments with quantitative and qualitative results using three different data sets (Section 6) show that similar high performances for similar parameter values are possible for different data sets because the proposed algorithm exploits the regularity in the structure in the projection profiles in a way that is invariant to contrast, scale, and orientation differences in the raw image data. The rest of the paper describes the details of the proposed approach and presents experimental results.

2. Pre-processing

The texton model is assumed to correspond to a filter for which the image areas with a high response are more likely to contain this texton than areas with a low response. Popular such filters in the literature include edge, bar, and spot filters at multiple scales and orientations. For example, Leung and Malik [11] used a set of 48 filters including first and second derivatives of Gaussians at 6 orientations and 3 scales, 8 Laplacian of Gaussian (LoG) filters, and 4 Gaussian filters; Schmid [32] used 13 isotropic Gabor-like filters; Varma and Zisserman [12] used a set of 38 filters including an edge and a bar filter each at 6 orientations and 3 scales, one Gaussian filter, and one LoG filter; Zhu et al. [33] used a set of 119 filters including 7 LoG filters, and Gabor sine and Gabor cosine filters each at 8 orientations and 7 scales; and Shotton et al. [13] used a set of 17 filters consisting of Gaussian, derivative of Gaussian, and LoG filters at different scales.

Following the common practice, we use the Laplacian of Gaussian filter as a generic texton model that is sensitive to contrast differences in any orientation. Note that any other filter can also be used because the following step uses the filter responses that enhance the texton-like objects in the image. The rest of the algorithm aims to model the arrangements of the textons using the local extrema in the response image, and can work with any texton model with its corresponding filter.

The isotropic LoG filter has a single scale parameter corresponding to the Gaussian function. Since the length of the cross-section between the zero crossings of the LoG filter is $2\sqrt{2}\sigma$, the σ parameter can be selected according to the sizes of the textons of interest. Figure 2 shows some of the LoG filters among the cross-sections (scales) of 2 to 9 pixels used in this study.

3. Projection profiles and regularity detection

After the texton-like objects are enhanced in an image, the pixels having high responses (local maxima) on a scan line along the image indicate possible locations of such objects. In

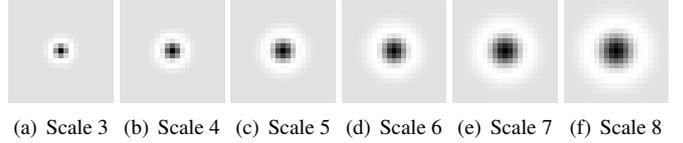


Figure 2: Laplacian of Gaussian filters for different scales.

a neighborhood with a regular repetitive structure, the locations of local maxima along the scan line with an orientation that matches the dominant direction of this structure will also have a regular repetitive pattern. The next step involves converting the image data into 1D signals using projection profiles at particular orientations, and quantifying the regularity of the textons along these orientations in terms of the periodicity of these profiles using wavelet analysis.

3.1. Projection profiles

The existence of the regularity of the local extrema along a particular orientation at a particular scale (particular LoG filter output) can be measured using the projection profile along that orientation in an image window. Given a scan line representing a particular orientation, the vertical projection profile is computed as the summation of the values in individual columns (in perpendicular direction to the scan line) of an oriented image window constructed symmetrically on both sides of this scan line.

The profile is denoted as $x[n], n = 1, \dots, N_p$ where N_p is the window width in terms of the number of pixels. This profile will contain successive peaks with similar shapes if the orientation of the scan line matches the orientation of the structural texture pattern. Furthermore, regularity along multiple image rows that are parallel to the selected scan line and are covered by the corresponding window will enhance these peaks as well. For an ideal structural texture, similar peaks can also be observed in 90 degree and 45 degree rotated projections. If the orientation of the scan line and the corresponding projection do not match that of the structural texture, or if there is no significant regular pattern in the window, the peaks will have arbitrary shapes.

When the proposed texture model is applied to a real image, there may not be a particular orientation where all textons align perfectly. The direction of alignment may also gradually change in the image. Moreover, the sizes of the textons and the distances between them may not always be the same. As long as there is a sequence of textons with similar sizes and similar placement patterns, the projection profile is expected to produce a near-periodic signal corresponding to the near-regular repetitive arrangement.

Observing such periodic signals is necessary but not sufficient for detecting structural texture patterns. The widths of the peaks in the projection profile should also match the sizes of the textons of interest as much as possible. Moreover, the periodic signal should be observed for some duration, not only for only one window, but also for a set of overlapping windows using the same or similar projection directions. In practice, it may be quite unlikely to observe perfectly periodic signals in the projection profiles of real images with natural textures. Therefore,

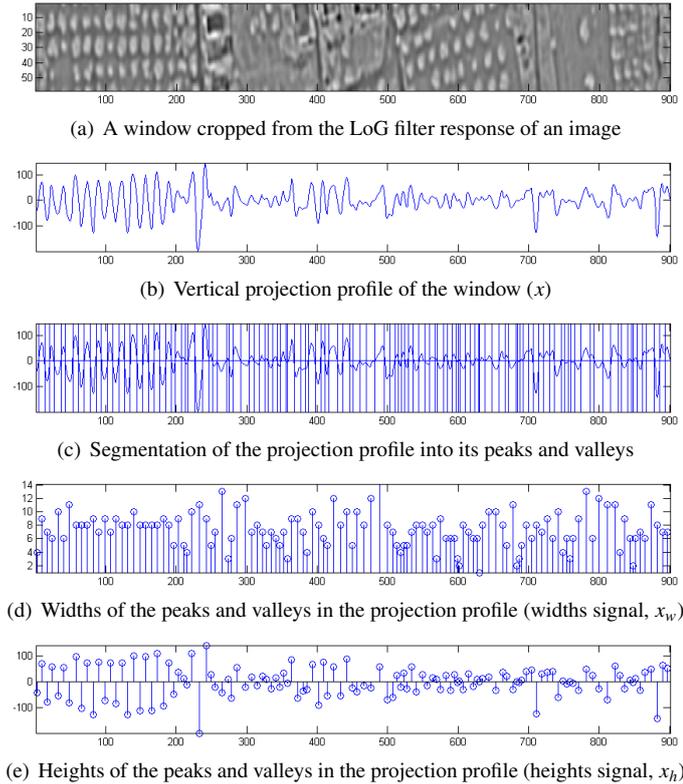


Figure 3: Segmentation of the projection profile of an example image window and the corresponding width and height features of the resulting peaks and valleys.

analysis of projection profiles for periodicity should use this relaxed definition for the structural pattern for robust detection of a wide range of highly distorted and noisy structural textures.

Figure 3(a) shows a window cropped from an image taken from Google Earth, and Figure 3(b) shows the vertical projection profile of an LoG filter response of this window. It can be observed that the projection signal becomes periodic over the region on the left part of the window where the textons, i.e., trees in this image, are arranged regularly in rows and columns. For this particular case, the alignment of the textons and the projection direction matches. However, no significant periodicity is observed for the structural pattern on the right part of the window because the orientation of the window does not match the dominant direction of the structure.

3.2. Profile segmentation

The regularity of the texture along a particular orientation is assumed to be represented in the periodicity of the corresponding projection profile. Since it may not always be possible to find a perfect period, especially for natural textures, we designed an algorithm that measures the amount of periodicity and locates the periodic part within the larger profile signal.

The algorithm uses an additional layer of abstraction by analyzing the peaks and valleys of the profile because a periodic signal can be coarsely defined as a sequence of similar peaks and valleys where peaks are always followed by valleys in an alternating manner. In addition to the alternation property, the

width and height values of the peaks should also be similar to each other because the peaks correspond to high responses in the LoG filter output and the textons in this output are expected to be of the same size (scale). The same argument is also valid for the valleys. The valleys correspond to the distances between consecutive textons because they are formed by low responses in the LoG filter output. Therefore, their sizes are also expected to be close to each other in a periodic signal corresponding to a regular texture pattern. However, in a near-periodic signal, the widths and heights of the peaks or valleys may not be exactly equal so the algorithm must be tolerant to local variations, distortions, and noise.

The segmentation of the profile signal into its peaks and valleys is achieved by finding the zero crossings, local minima in the positive plane, and local maxima in the negative plane. The zero crossings correspond to the alternation of peaks and valleys in the projection signal. Segmentations over local minima and maxima occur when the signal is not periodic, since peaks and valleys are expected to be prominent with symmetric shapes around their unique maximal and minimal points, respectively. The output of the segmentation step consists of the locations of the starting pixel location of each peak or valley, denoted as $n_i, i = 1, \dots, N_s$ where N_s is the total number of peaks and valleys in the segmented projection signal. Peak and valley segmentation examples are shown in Figure 3(c).

After obtaining all peaks and valleys, their width and height features are calculated and stored according to their order in the projection signal and are denoted as $x_w[i]$ and $x_h[i], i = 1, \dots, N_s$, respectively. These signals are descriptive enough to analyze the general behavior and the periodicity of the original projection signal as shown in Figures 3(d) and 3(e).

In order to avoid false or over segmentation of the peaks and valleys, the projection signal may be smoothed by using an averaging filter. In this way, the periodicity analysis can focus more on the general trends observed in the course of the projection signal. However, in our case, no smoothing was applied because the LoG filter already includes a Gaussian component for pre-smoothing.

3.3. Periodic signal analysis

Pairs of peaks and valleys in the projection profile are regarded as the basic unit of the periodic signal analysis because the structural texture patterns of interest produce an alternating sequence of peaks and valleys in the profile. The peaks and valleys are paired according to their order in the sequence. It should be noted that a pair in the profile of a real texture may include two peaks, two valleys, or one peak and one valley in this sequence.

The initial steps of the peak-valley pair analysis focus on the width feature signal x_w and do not use the height feature signal x_h because the values of x_h may be affected by the local changes in the image contrast whereas the values of x_w depend only on the scales of the textons and their arrangements, and are invariant to such changes. Given a peak-peak, valley-valley or peak-valley pair, the width pair signal is computed using the difference between the consecutive width values in the pair. This corresponds to the detail coefficients of the wavelet transform

of the width feature signal $x_w[i], i = 1, \dots, N_s$, computed using the Haar wavelet filter. Note that the ranges of these difference values in the width signal depend on the local scales of the textons. Therefore, a normalization step is used to obtain compatible values for different scales that may exist in the image. This is achieved by dividing the detail coefficients by their respective average coefficients in the Haar wavelet transform. This computation of the width pair signal as

$$x_{wp}[i] = \frac{x_w[2i-1] - x_w[2i]}{x_w[2i-1] + x_w[2i]}, i = 1, \dots, N_s/2 \quad (1)$$

enables the values to be in the $[-1, 1]$ range while preserving the relative local changes in the features of the peak-valley pairs.

A projection signal may be composed of periodic and non-periodic intervals of varying lengths. The context of individual peak-valley pairs is important for determining periodic, near-periodic, or non-periodic areas. The periodic intervals that we are interested in contain a train of peak-valley pairs with similar characteristics. The more peak-valley pairs with similar characteristics follow each other in the projection profile, the longer the interval of the periodicity is. Not only the duration of the periodic interval but also the quality of the periodic signal is important.

It is possible to assign scores to the peaks and valleys of a projection profile for being part of a periodic interval using the normalized width pair feature signal x_{wp} in Equation (1). In particular, the existence of high-frequency components in this signal indicates irregular peak pair instances. The irregularities can also be quantified using the detail coefficients of a second level of wavelet transform computed using the Haar filter. These detail coefficients correspond to fine changes in x_{wp} . Over irregular regions, the detail coefficients tend to get higher values, whereas these coefficients are close to zero for regions with a regular behavior.

The absolute values (L_1 norm) of these coefficients are computed as the wavelet energies representing their high-frequency content, and are used as the irregularity score

$$x_{irreg}[i] = \left| \frac{x_{wp}[2i-1] - x_{wp}[2i]}{2} \right|, i = 1, \dots, N_s/4 \quad (2)$$

where $x_{irreg}[i] \in [0, 1]$. Each value of x_{irreg} corresponds to a sequence of 4 consecutive peaks and/or valleys (corresponding to two levels of Haar wavelet analysis described above), and can be upsampled by 4 to reconstruct an irregularity score for each peak and valley. Finally, we convert this irregularity score to a regularity score as

$$x'_{reg}[i] = 1 - x'_{irreg}[i], i = 1, \dots, N_s \quad (3)$$

where $x'_{reg}[i], i = 1, \dots, N_s$, is the upsampled version of x_{irreg} in Equation (2) from a length of $N_s/4$ to a length of N_s , resulting in $x'_{reg}[i] \in [0, 1]$ as shown in Figure 4(b). The peaks and valleys whose regularity scores are close to 1 are candidates to be part of a regular periodic signal. These scores can be thresholded for locating the periodic areas of interest.

In addition to the peaks and valleys that are decided to belong to irregular areas with respect to the wavelet energies in

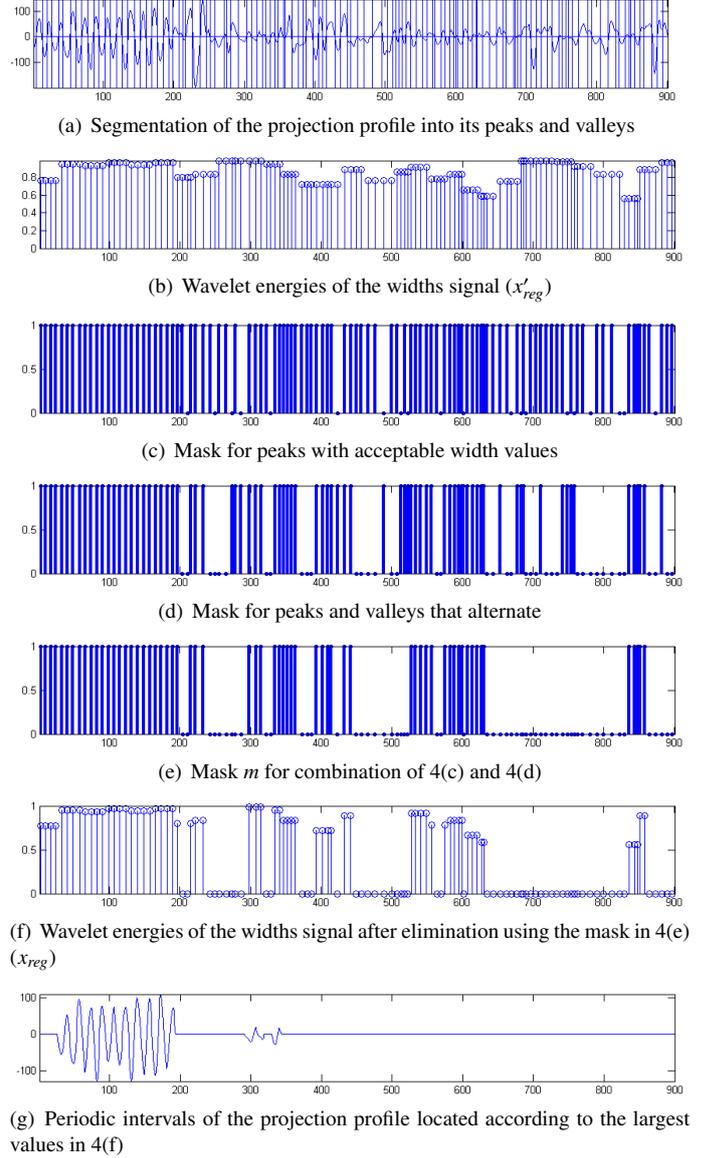


Figure 4: Periodicity analysis of the projection profile of an image window.

Equation (2), some more peaks and valleys can be eliminated according to the expected shape of the corresponding periodic signal. As pointed out earlier, the projection profile of a regular texture is a sequence of peaks and valleys alternating between the positive and negative planes. In addition, the peaks whose widths are significantly smaller or greater than the scale of interest (corresponding to the scale of the LoG filter) can be eliminated. If the width values of the peaks are not in the specified interval or they are not in an alternating sequence, a masking signal $m[i], i = 1, \dots, N_s$, is constructed as

$$m[i] = \begin{cases} 0 & (x_w[i] > s + \epsilon) \vee (x_w[i] < s - \epsilon) \vee \\ & (\text{sign}(x_h[i]) = \text{sign}(x_h[i+1])) \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

where s is the scale in pixels and ϵ is a small integer (e.g., 1 or

2), and the regularity scores are updated as

$$x_{reg}[i] = x'_{reg}[i] \times m[i]. \quad (5)$$

The mask, the resulting regularity scores, and the part of the projection profile detected to be regular are illustrated in Figures 4(c)–4(g).

4. Multi-orientation and multi-scale regularity analysis

The regularity detection using the periodicity analysis of projection profiles as described in Section 3 is done on a particular profile computed using a particular LoG filter output (particular scale) and a particular orientation in an image window. However, the orientation of the texture pattern and the projection direction may not always match. Furthermore, an image may contain structural textures at multiple orientations composed of textons at multiple scales. Therefore, the projection profiles for different orientations and different scales should be analyzed so that a structural pattern at an arbitrary orientation and an arbitrary scale can be detected with periodic signal analysis.

4.1. Multi-orientation regularity analysis

For a particular scale approximated using a particular LoG filter output image, we perform multi-orientation regularity analysis by sliding image-wide oriented windows called strips over that image. Each strip is defined by a scan line corresponding to the symmetry axis of the strip and a height parameter defining the extent of the strip on both sides of this scan line. In the formulation below, a distance parameter d and an orientation parameter θ define the scan line, and the strip height is denoted as δ .

Given an image with N_r rows and N_c columns, and $r' = r - N_r/2$ and $c' = c - N_c/2$ being the normalized row and column coordinates, respectively, with respect to an origin at the center of the image, the strip is defined using the inequality

$$|r' \cos(\theta) - c' \sin(\theta) - d| < \frac{\delta}{2} \quad (6)$$

where θ is measured relative to the horizontal axis in clockwise direction. For each pixel, all combinations of $d \in [-\sqrt{(N_r/2)^2 + (N_c/2)^2}, \sqrt{(N_r/2)^2 + (N_c/2)^2}]$ and $\theta \in [-90^\circ, 90^\circ]$ values produce a set of strips with scan lines passing through that pixel at 180 different orientations where positive values of d cover the lower half of the image and negative values of d cover the upper half of the image. Example strips for different values of d and θ are illustrated in Figure 5.

The projection profile corresponding to each strip is computed using summation along $\theta + 90$ degrees. Given the profile denoted as $x[i], i = 1, \dots, N_p$ in Section 3.1, the periodic signal analysis is performed on this profile as described in Section 3.3, and the regularity scores are calculated as $x_{reg}[i], i = 1, \dots, N_s$ using Equations (3) and (5). Then, the scores for each peak and valley in the profile signal are recorded back to the corresponding pixels on the scan line defining the strip using the list

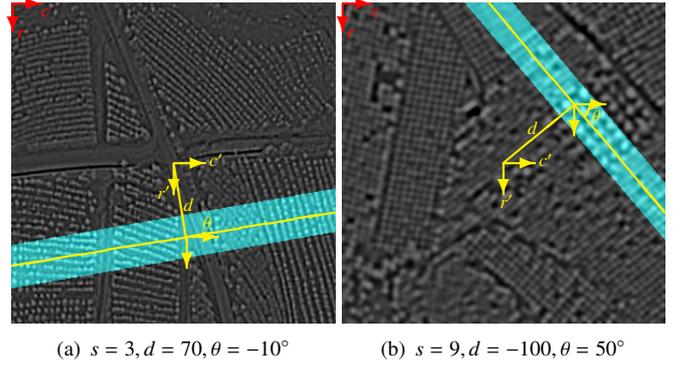


Figure 5: Example strips for computing the projection profiles of LoG filter outputs. Each strip is marked as green together with the scan line that passes through its symmetry axis that is marked as yellow. The strip height δ is selected as 40 pixels in these examples.

of starting pixel locations $n_i, i = 1, \dots, N_s$ of these peaks and valleys as described in Section 3.2. The result of this step is a three dimensional matrix storing 180 regularity scores in the $[0, 1]$ range for each pixel for a particular scale.

The strip height δ is a design parameter. If the height of the strip is increased, it is possible to find only texture patterns occupying larger areas. If the texture pattern is noisy or warped, then using smaller strip sizes should be preferred. However, decreasing the strip size too much is also not desirable because the projection is no longer effective for such cases. In this work, we use a strip size that is adaptive to the scales of interest. In the experiments, a multiplier $k_\delta = 2$ of scale s is used to obtain strip sizes that are twice the size of the expected textons at that scale.

4.2. Multi-scale regularity analysis

The multi-orientation regularity analysis described in Section 4.1 is performed independently for each scale using the corresponding LoG filter output. The resulting regularity values for all orientations and all scales for all pixels are stored in a four dimensional matrix denoted as $\rho(r, c; \theta, s)$ where $(r, c), 1 \leq r \leq N_r, 1 \leq c \leq N_c$ denotes the pixel locations, $\theta \in [-90^\circ, 90^\circ]$ represents the orientations, and $s \in \mathcal{S}$ represents the scales with \mathcal{S} being the set of scales of interest such as $\mathcal{S} = \{2, \dots, 9\}$ as illustrated in Section 2.

5. Near-regular texture localization

The goal of the last step is to compute a regularity index for each pixel to quantify the structure of the texture in the neighborhood of that pixel along with estimates of the orientation of the regularity as well as its scale. For robustness, it is expected that this regularity index is consistent among neighboring pixels for a certain range of orientations and scales. In other words, a high regularity value at a particular pixel for a particular orientation and scale can be considered as noise if neighboring pixels do not have a high regularity value at similar orientations and scales. Such noisy cases can be suppressed by convolving $\rho(r, c; \theta, s)$ with a four dimensional Gaussian filter with size

$11 \times 11 \times 11 \times 3$ that expects consistency in a 11×11 spatial neighborhood for an orientation range of 11 degrees and a range of three scales. This filtering step also introduces contributions to the regularity values from neighboring pixels, orientations, and scales.

The final regularity index is defined as the maximum regularity score at each pixel and the principal orientation and scale for which this score is maximized. The regularity index is computed as

$$\rho^*(r, c) = \max_{\theta, s} \rho(r, c; \theta, s) \quad (7)$$

along with

$$\{\theta^*(r, c), s^*(r, c)\} = \arg \max_{\theta, s} \rho(r, c; \theta, s). \quad (8)$$

Note that there may be highly structured areas where the regularity index achieves similarly high values at 90 degree and even 45 degree rotated projections. In some cases, the principal orientations obtained using (8) for some of the pixels in the same neighborhood may be 90 degree rotated versions of each other. The values in such neighborhoods may alternate between these principal orientations, and may yield a noisy picture when visualized. In such cases, spatially consistent orientation values can be obtained by using a majority voting or a median filter as a post-processing step.

Finally, given $\rho^*(r, c) \in [0, 1]$, $\theta^*(r, c) \in [-90^\circ, 90^\circ]$, and $s^*(r, c) \in \mathcal{S}$, the image areas with a structural texture composed of a near-regular repetitive arrangement of textons can be localized by thresholding the regularity index at each pixel. This thresholding can be done either manually by the user or by using an automatic thresholding technique [34]. The final detection map can be produced by using morphological opening and closing operations for eliminating small isolated regular regions that most likely correspond to false alarms and to fill small isolated irregular regions that most likely correspond to a few missing textons within a structural texture.

6. Experimental results

The overall algorithm and the required parameters are summarized in Algorithm 1. Since the algorithm is fully unsupervised, i.e., no training is required, the final detection map for an input image can be computed once the parameters are set. All parameters except the threshold for the regularity index can easily be assigned intuitive values according to the resolution of the input image and the textons of interest.

The performance of the proposed structural texture model was evaluated using three different data sets obtained from the Prague benchmark, Google Earth, and the PSU near-regular texture database. We used the same values for all parameters for all data sets even though they had quite different characteristics. The set of scales \mathcal{S} corresponding to the sizes of the textons of interest was fixed as $\{2, \dots, 9\}$ pixels. The scale multiplier k_δ that is used to compute the strip height for the projection profile relative to the texton scale was fixed at 2. Similarly, the ϵ tolerance for eliminating the peaks in the projection profile whose width values are not compatible with the texton scale

Algorithm 1 Near-regular texture localization algorithm

Require: Grayscale image with N_r rows and N_c columns
for all scales $s \in \mathcal{S}$ **do** {parameter: set of scales \mathcal{S} }
 Apply LoG filter
 for all orientations $\theta \in [-90^\circ, 90^\circ]$ **do**
 for all distances $d \in [-\sqrt{(N_r/2)^2 + (N_c/2)^2}, \sqrt{(N_r/2)^2 + (N_c/2)^2}]$ **do**
 Compute projection profile {parameter: scale multiplier k_δ for strip height}
 Segment projection profile
 Compute regularity score {parameter: threshold ϵ for width mask}
 Store scores in $\rho(r, c; \theta, s)$
 end for
 end for
 Smooth scores $\rho(r, c; \theta, s)$ {parameter: smoothing filter size}
 Compute regularity index, principal orientation and scale $\rho^*(r, c)$ $\theta^*(r, c)$, $s^*(r, c)$
 Threshold regularity index {parameter: threshold}
 Eliminate small isolated regular regions {parameter: threshold}
 Fill small isolated irregular regions {parameter: threshold}

was set to 2 pixels. The smoothing filter that is used for introducing contributions to the regularity values from neighboring pixels, orientations, and scales, as well as for suppressing inconsistent values among neighboring pixels for a certain range of orientations and scales was fixed to a Gaussian filter with size $11 \times 11 \times 11 \times 3$. The regularity index threshold for the localization of the structural texture areas was varied from 0.6 to 1 with increments of 0.01. Finally, the minimum allowable area of a regular region was varied between 0 and 5000 pixels with increments of 1000, and the minimum allowable area of an irregular region within a regular region was also varied between 0 and 5000 pixels with increments of 1000. These settings corresponded to 1440 different parameter combinations for each data set.

The rest of the section presents detailed quantitative and qualitative results for individual data sets. Given ground truth data where pixels belonging to structural texture areas are labeled as positive and the rest of the image is labeled as negative, quantitative evaluation was performed using receiver operating characteristics (ROC) curves plotting true positive rates (TPR)

$$\text{TPR} = \frac{\text{positives correctly detected}}{\text{total positives}} \quad (9)$$

versus false positive rates (FPR)

$$\text{FPR} = \frac{\text{negatives incorrectly detected}}{\text{total negatives}} \quad (10)$$

for different values of the parameters [35]. The performances of different settings were ranked using the overall accuracy rate (ACC)

$$\text{ACC} = \frac{\text{true positives} + \text{true negatives}}{\text{total number of pixels}}. \quad (11)$$

We also present results obtained with the JSEG [36] and EDISON [37] algorithms as two popular segmentation methods with publicly available code for comparison. The JSEG algorithm consists of a color quantization step that is followed by a spatial segmentation step that uses the quantized color values for modeling texture. The EDISON algorithm is a color-based segmenter that is based on the mean shift algorithm. Both of the methods aim to achieve a full segmentation of the whole image and do not provide a classification of structural versus stochastic texture areas (that may actually be achieved using a follow-up supervised classification step). Therefore, we only made a visual comparison of the detection and localization produced by the proposed algorithm with the region boundaries obtained using the JSEG and EDISON algorithms. The default parameter settings provided by the authors of the respective algorithms were used in the experiments.

We also experimented with several co-occurrence matrix-based texture features obtained using different displacement vectors at multiple orientations and scales. However, unsupervised methods such as thresholding or k-means clustering of the resulting features could not detect and localize the structural texture areas. Supervised classification, as commonly used in the literature, may provide better results but supervised methods are beyond the scope of this paper as the proposed method is fully unsupervised.

6.1. Prague benchmark data set

The first data set consists of 50 texture mosaic images, each with a size of 256×256 pixels, obtained using the Prague texture segmentation data generator [38]. The images were generated with randomly selected cut-outs from the nature, rock, stone, textile, wood, and bidirectional texture function (BTF) categories where randomly generated splines formed the texture boundaries. 20 of these images contained patches from 4 different texture classes where each patch consisted of 3 instances of the same type of texture at different rotations and scales. The remaining 30 images contained patches from 6 different texture classes where each patch consisted of a single instance of a texture class. The data generator produced a binary mask for each patch. We combined the masks for the patches that corresponded to the textile and BTF classes as positive ground truth for structural textures whereas the rest of the classes were considered as stochastic textures and formed the negative ground truth.

Figure 6(a) shows the ROC curves obtained by averaging the TPR and FPR values over the whole data set, and Table 1(a) summarizes the parameter settings that obtained the best performance among all combinations. Figure 7 presents example images and the corresponding results. The highest average accuracy over all 50 images was obtained as 95.28% using the proposed algorithm. The 4.72% error was mostly observed as some misdetections at the texture boundaries and some false alarms at a few of the nature, rock, stone, and wood patches that contained small areas with some repetitive patterns. Orientation estimates were also very highly accurate even for the patches that consisted of multiple instances of the same type

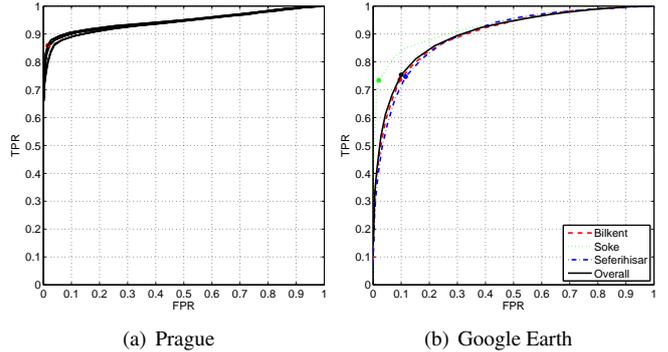


Figure 6: ROC curves obtained using the proposed algorithm for the Prague and Google Earth data sets. (a) shows multiple curves obtained by varying the regularity index threshold for all combinations of the minimum area thresholds. The setting reported in Table 1(a) is shown as a red dot. (b) shows the curves obtained by varying the regularity index threshold for the area threshold combinations reported in Table 1(b) for different data sets.

Table 1: The parameter settings that obtained the best performances for the Prague and Google Earth data sets. T: regularity index threshold. minReg: minimum allowable area of a regular region. minIrreg: minimum allowable area of an irregular region.

(a) Prague

Data	T	minReg	minIrreg	TPR(%)	FPR(%)	ACC(%)
Prague	0.84	3000	2000	85.88	1.55	95.28

(b) Google Earth

Data	T	minReg	minIrreg	TPR(%)	FPR(%)	ACC(%)
Bilkent	0.82	4000	5000	73.63	9.58	83.32
Soke	0.84	4000	4000	73.33	2.00	93.92
Seferihisar	0.82	4000	4000	74.68	11.57	83.20
Overall	0.82	4000	5000	75.34	10.00	84.56

of texture at different rotations and scales, with a clear identification of sharp orientation changes within these patches. We observed that the scale estimates were also accurate for most of the patches. The results showed that the proposed method could detect and localize the structural texture areas at different illumination and contrast levels as well.

The performance was similar when different parameter settings were considered. For example, different combinations of the minimum area thresholds for the last two steps of Algorithm 1 gave very similar results as shown in Figure 6(a). This leaves the regularity index threshold as the only significant parameter in the algorithm. However, a particular value for a given data set can easily be selected interactively when no ground truth exists, or by minimizing the classification error for a global threshold or by using an automatic thresholding technique for a local threshold when some ground truth (validation data) is available.

On the other hand, the JSEG and EDISON algorithms could not produce accurate segmentation boundaries for this data set. JSEG could detect some of the boundaries and was more accurate than EDISON, which is a purely color-based method, as expected. However, it could not identify most of the boundaries correctly, especially when the neighboring texture patches did not have a significant contrast difference. It could be possi-

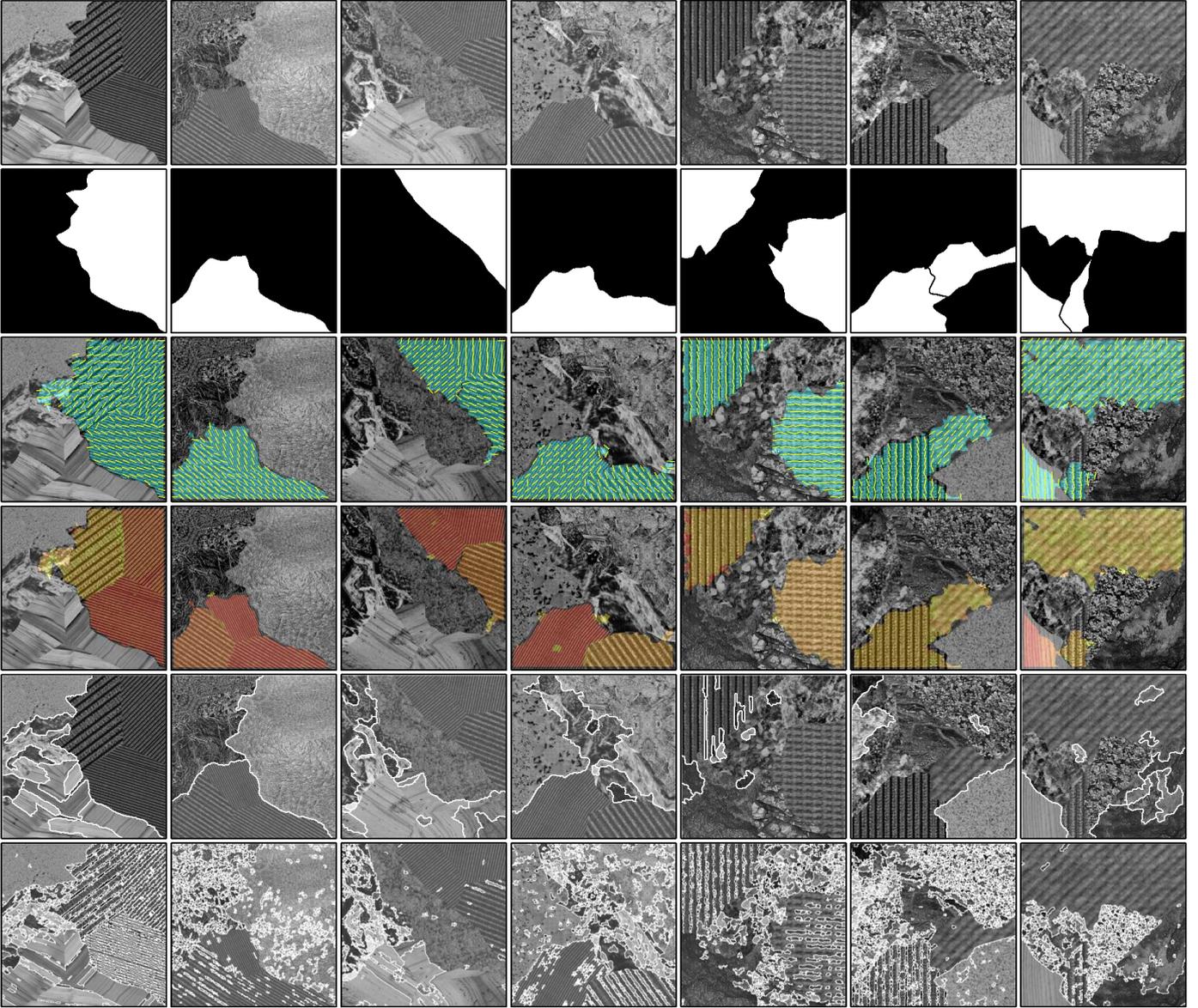


Figure 7: Example results for the Prague data set. Each column shows the results for a particular image. The first row shows the original texture mosaics. The second row shows the ground truth where the positive regions are marked as white. The third row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The fourth row shows the scale estimates using the color map given in Figure 9. The fifth and sixth rows show the segmentation boundaries obtained using the JSEG and EDISON algorithms, respectively.

ble to obtain slightly better results by tuning the parameters but this required a different set of parameters for each image, and still could not achieve a comparable accuracy for the structural textures with respect to the proposed method.

6.2. Google Earth data set

The second data set consists of 12 images, each with a size of 1680×1031 pixels, saved from Google Earth. 5 of these images were taken over the Bilkent University campus, 2 images were from the Soke region in the Aydin province, and 5 images were from the Seferihisar region in the Izmir province in Turkey. These images contained vegetation with different characteristics and planting patterns that could be considered as

challenging natural structural textures. The tree groups corresponding to artificially planted areas as well as orchards were manually labeled as the positive ground truth.

Figure 8 presents example images and the corresponding results. Figure 6(b) shows the ROC curves obtained by averaging the TPR and FPR values over the individual sites as well as the whole data set, and Table 1(b) summarizes the parameter settings that obtained the best performance among all combinations. This data set provided a significant challenge for the detection of real structural textures compared to commonly used data sets that contain a single almost ideal texture in each image. The highest average accuracies for the Bilkent and Seferihisar sites were obtained similarly at slightly above

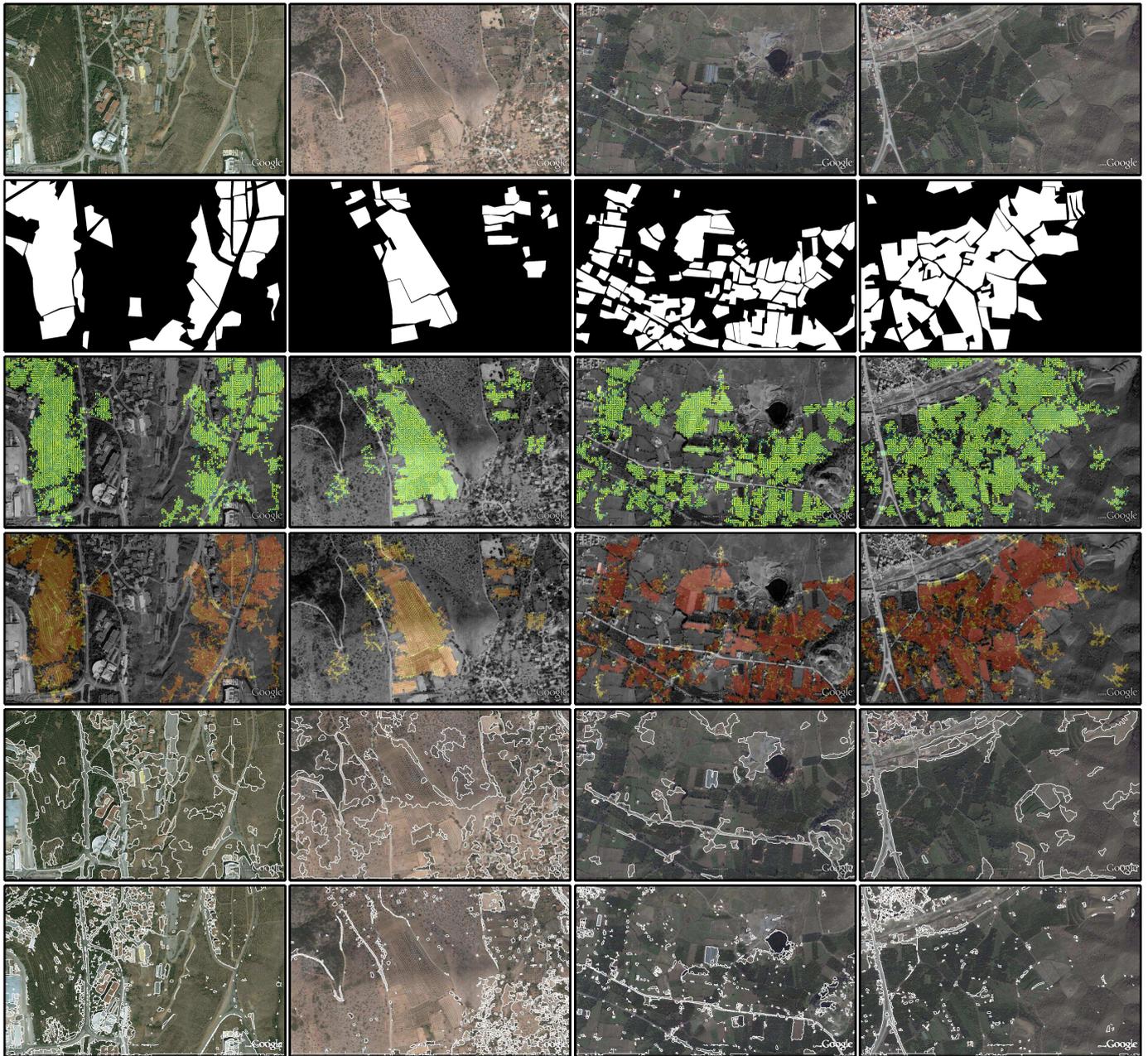


Figure 8: Example results for the Google Earth data set. Each column shows the results for a particular image. The first row shows the original images. The second row shows the ground truth where the positive regions are marked as white. The third row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The fourth row shows the scale estimates using the color map given in Figure 9. The fifth and sixth rows show the segmentation boundaries obtained using the JSEG and EDISON algorithms, respectively.

83%. The average accuracy for the 2 Soke images was about 94% due to lower false positive rate at a similar true positive rate. The average accuracy over all 12 images was obtained as 84.56%. Most of the false positives were observed along roads where there was a repetitive contrast difference on both sides, and around some residential developments where a similar regular contrast difference was observed due to neighboring buildings. The misdetections mostly occurred at small vegetation patches that were marked as positive in the ground truth due to a few rows of regularly planted trees but were eliminated

at the last step of the algorithm because of the minimum area thresholds.

The best parameter settings for individual sites as well as for the whole data set were very similar to those for the Prague data set. In particular, the regularity index thresholds were very close to each other, and the minimum area thresholds were slightly larger for the Google Earth images as the images and the structures they contained were larger. The similar performances for similar regularity index thresholds were possible because the proposed algorithm exploits the regularity in the

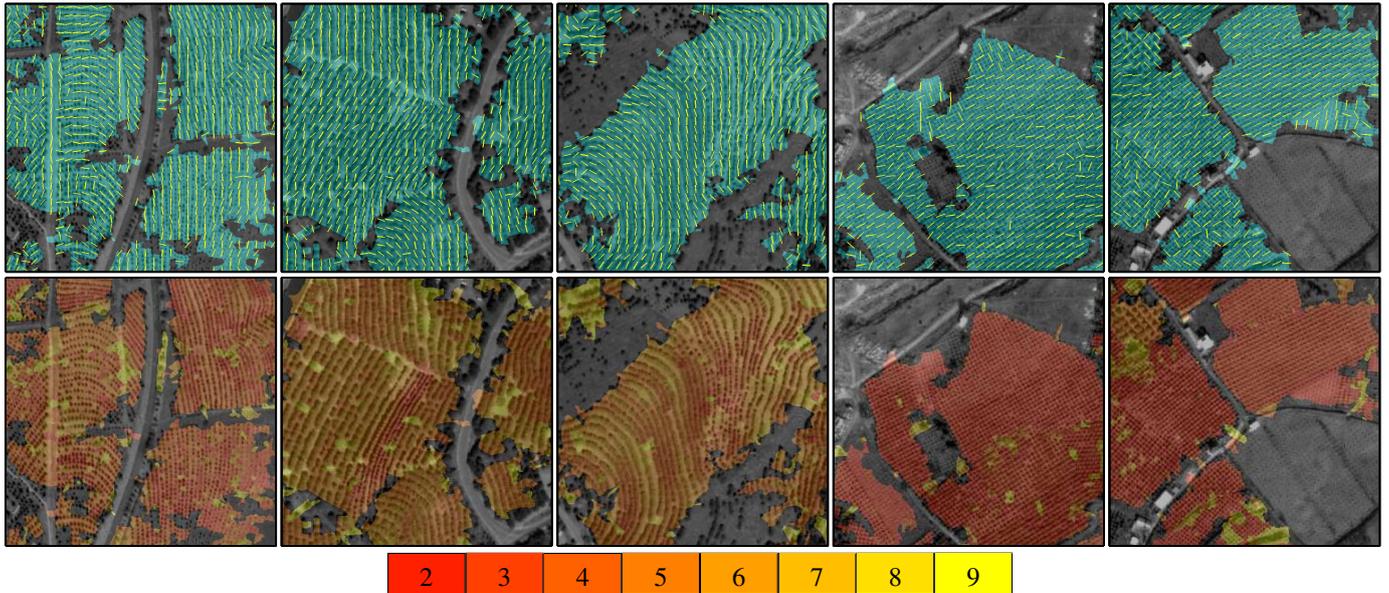


Figure 9: Local details of structural texture detection, orientation and scale estimation. The first row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The second row shows the scale estimates using the color map shown on the third row.

structure in the projection profiles using the periodicity analysis in a way that is invariant to contrast, scale, and orientation differences in the raw image data.

Orientation and scale estimates were also very accurate as in the Prague data set. Figure 9 illustrates the local details to observe the accuracy of these estimates. These examples show that even the gradually changing orientations could be estimated smoothly, and the localization of the structural texture areas was very accurate even when no sharp boundaries existed in the image data.

Figure 8 also shows the results for the JSEG and EDISON algorithms. Since the main assumption behind most segmentation algorithms is to obtain regions that are homogeneous in terms of color and/or micro-texture information, these algorithms mostly resulted in boundaries around areas having a high contrast difference with their surroundings. It can also be observed from these results that these algorithms could not find boundaries around areas with a near-regular repetitive arrangement of individual textons as expected.

6.3. PSU near-regular texture data set

The third data set contains samples taken from the near-regular texture database maintained at the Pennsylvania State University [39]. Most of this database contains images with a single synthetic or real texture for the evaluation of symmetry detection or lattice extraction. As examples for real textures within a different background, we collected several samples, each with a size of 800×600 or 600×800 pixels, from the building album of this database.

There is no ground truth for this data set so only qualitative examples are shown in Figure 10. The resulting detection and localization as well as the orientation and scale estimates for the structural texture of the buildings were quite accurate even

though the buildings had faces at different views and the textons (i.e., the windows) did not necessarily fit perfectly to the definition in Section 2. We believe that the results for all three data sets show the power of the proposed unsupervised method for the detection and localization of structural textures with different orientations and scales using only grayscale information.

6.4. Computational complexity

The proposed method was implemented in Matlab. The overall processing using the unoptimized Matlab code took 139 minutes on the average for 1000×1000 Google test images on a PC with a 2 GHz Intel Xeon processor. We performed a code profile analysis to investigate the time spent in different steps. Among the major steps, on the average, pre-processing using the LoG filters took 0.08% of the time, the multi-orientation and multi-scale regularity analysis took 91.35% of the time, and smoothing the scores before computing the regularity index took 8.51% of the time using the parameter settings in Algorithm 1.

The most time consuming step was the multi-orientation and multi-scale regularity analysis. The image-wide strips used for performing the multi-orientation regularity analysis were implemented by rotating the whole image at one degree increments, and by sliding image-wide windows with one pixel sliding interval vertically over the image. The projection profiles were computed incrementally by adding the values of the pixels in the row that entered the strip and subtracting those in the row that left the strip. It may be possible to make the method more efficient if data structures like integral images are used to compute the profiles at different orientations [40].

Within the multi-orientation and multi-scale regularity analysis step, the image rotations described above took 32.02% of the time, segmenting the projection profiles took 52.57% of the time, and the periodic signal analysis using wavelet energies

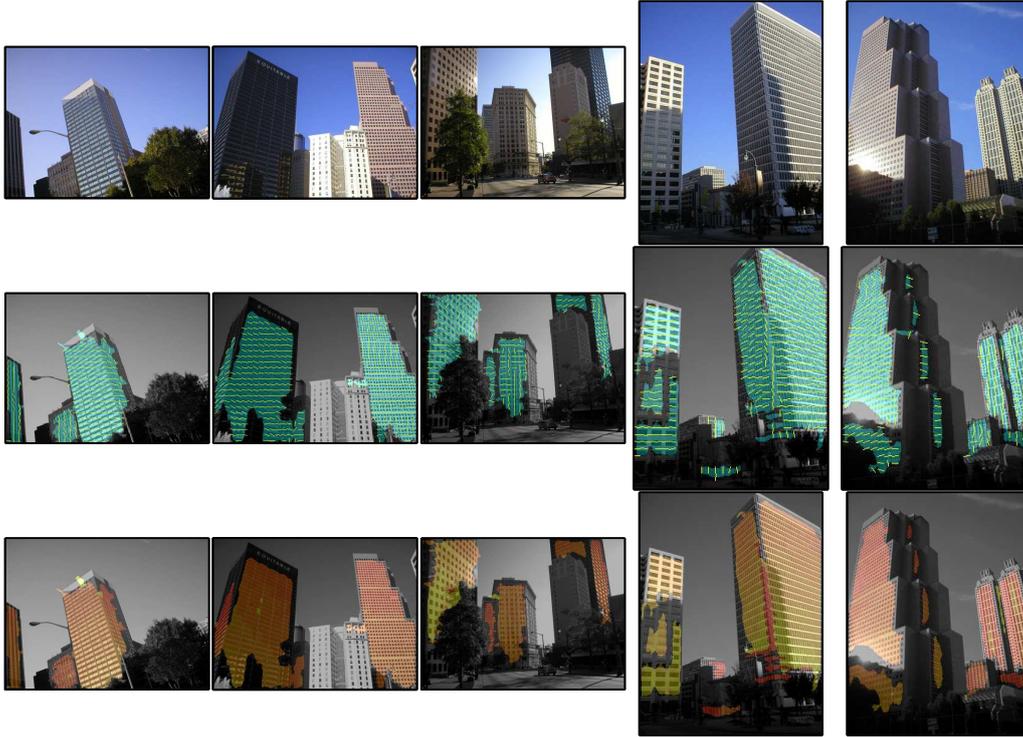


Figure 10: Example results for the PSU data set. Each column shows the results for a particular image. The first row shows the original images. The second row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The third row shows the scale estimates using the color map given in Figure 9.

took 2.39% of the time. To investigate potential improvements of a C version of the code, we re-implemented the projection profile segmentation step in C. This resulted in a 110 times reduction in the processing time of that step, and decreased the overall average processing time for 1000×1000 images to 90 minutes.

Significant reductions in computation time with a small change in accuracy are possible by using smaller sets of orientations and distances for the multi-orientation regularity analysis. For example, using only 36 different orientations instead of the full set of 180 by rotating the image at five degree increments and sliding the strips with two pixel increments instead of one pixel increments reduced the processing time from 139 minutes to 20 minutes on the average while having only approximately 1% change in the accuracy rate (the accuracy for some images increased slightly and the accuracy for some images decreased slightly) for a subset of Google images. Using a smaller set of scales will also decrease the computation time because the time complexity is linear in the number of scales. The method provides flexibility for the user’s adjustment of the parameters in Algorithm 1 for different trade-offs between computation time and localization accuracy.

7. Conclusions

We described a novel unsupervised method for the detection and localization of structural textures that were formed by near-regular arrangements of texture primitives. The method

used multi-scale Laplacian of Gaussian filters for the enhancement of potential texton locations, computed projection profiles of filter responses within oriented sliding windows, quantified the regularity of the textons in terms of the periodicity of these profiles using wavelet analysis, and resulted in a regularity score at each pixel for each orientation and scale. The final output was a regularity index that was computed for each pixel as the principal orientation and scale for which this score was maximized. Thresholding of this regularity index produced an accurate simultaneous localization of multiple structural texture areas, along with estimates of their orientations and scales, in real images containing different kinds of textures as well as non-textured areas. Unlike existing studies that aimed to model the structure in texture patches that contained a single type of texture, the performance of the proposed method was evaluated using three different data sets, and the quantitative and qualitative results showed its effectiveness for the detection and localization of structural textures in real images containing complex scenes.

Acknowledgment

This work was supported in part by the TUBITAK CAREER Grant 104E074.

References

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

- [2] J. Zhang, T. Tan, Brief review of invariant texture analysis methods, *Pattern Recognition* 35 (3) (2002) 735–747.
- [3] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, *IEEE Transactions on Systems, Man, and Cybernetics SMC-3* (6) (1973) 610–621.
- [4] S. Arivazhagan, L. Ganesan, Texture classification using wavelet transform, *Pattern Recognition Letters* 24 (9–10) (2003) 1513–1521.
- [5] B. S. Manjunath, W. Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8) (1996) 837–842.
- [6] G. M. Haley, B. S. Manjunath, Rotation-invariant texture classification using a complete space-frequency model, *IEEE Transactions on Image Processing* 8 (2) (1999) 255–269.
- [7] D. A. Clausi, M. E. Jernigan, Designing Gabor filters for optimal texture separability, *Pattern Recognition* 33 (11) (2000) 1835–1849.
- [8] F. Bianconi, A. Fernandez, Evaluation of the effects of Gabor filter parameters on texture classification, *Pattern Recognition* 40 (12) (2007) 3325–3335.
- [9] T. Matsuyama, S.-I. Miura, M. Nagao, Structural analysis of natural textures by fourier transformation, *Computer Vision, Graphics, and Image Processing* 24 (3) (1983) 347–362.
- [10] A. A. Ursani, K. Kpalma, J. Ronsin, Texture features based on Fourier transform and Gabor filters: An empirical comparison, in: *Proceedings of International Conference on Machine Vision, 2007*, pp. 67–72.
- [11] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, *International Journal of Computer Vision* 43 (1) (2001) 29–44.
- [12] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, *International Journal of Computer Vision* 62 (1–2) (2005) 61–81.
- [13] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, *International Journal of Computer Vision* 81 (1) (2009) 2–23.
- [14] A. Speis, G. Healey, An analytical and experimental study of the performance of Markov random fields applied to textured images using small samples, *IEEE Transactions on Image Processing* 5 (3) (1996) 447–458.
- [15] H. Deng, D. A. Clausi, Gaussian MRF rotation-invariant features for image classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (7) (2004) 951–955.
- [16] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7) (2002) 971–987.
- [17] M. Li, R. C. Staunton, Optimum Gabor filter design and local binary patterns for texture segmentation, *Pattern Recognition Letters* 29 (5) (2008) 664–672.
- [18] Z. Guo, L. Zhang, D. Zhang, Rotation invariant texture classification using LBP variance (LBPV) with global matching, *Pattern Recognition (to appear)*.
- [19] M. Donoser, H. Bischof, Using covariance matrices for unsupervised texture segmentation, in: *Proceedings of 19th IAPR International Conference on Pattern Recognition, Tampa, Florida, 2008*.
- [20] H.-B. Kim, R.-H. Park, Extracting spatial arrangement of structural textures using projection information, *Pattern Recognition* 25 (3) (1992) 237–245.
- [21] D. Chetverikov, R. M. Haralick, Texture anisotropy, symmetry, regularity: Recovering structure and orientation from interaction maps, in: *Proceedings of British Machine Vision Conference, 1995*, pp. 57–66.
- [22] V. V. Starovoitov, S.-Y. Jeong, R.-H. Park, Texture periodicity detection: Features, properties, and comparisons, *IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans* 28 (6) (1998) 839–849.
- [23] H.-C. Lin, L.-L. Wang, S.-N. Yang, Extracting periodicity of a regular texture based on autocorrelation functions, *Pattern Recognition Letters* 18 (5) (1997) 433–443.
- [24] Y. Liu, R. T. Collins, Y. Tsin, A computational model for periodic pattern perception based on frieze and wallpaper groups, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (3) (2004) 354–371.
- [25] J. Han, S. J. McKenna, R. Wang, Regular texture analysis as statistical model selection, in: *Proceedings of European Conference on Computer Vision, 2008*, pp. 242–255.
- [26] D. Charalampidis, Texture synthesis: Textons revisited, *IEEE Transactions on Image Processing* 15 (3) (2006) 777–787.
- [27] T. Leung, J. Malik, Detecting, localizing and grouping repeated scene elements from an image, in: *Proceedings of European Conference on Computer Vision, 1996*, pp. 546–555.
- [28] J. H. Hays, M. Leordeanu, A. A. Efros, Y. Liu, Discovering texture regularity as a higher-order correspondence problem, in: *Proceedings of European Conference on Computer Vision, 2006*, pp. 522–535.
- [29] W.-C. Lin, Y. Liu, A lattice-based MRF model for dynamic near-regular texture tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (5) (2007) 777–792.
- [30] M. Petrou, P. G. Sevilla, *Image Processing: Dealing with Texture*, John Wiley & Sons, Inc., 2006.
- [31] Y. Liu, Y. Tsin, W.-C. Lin, The promise and perils of near-regular texture, *International Journal of Computer Vision* 62 (1–2) (2005) 145–159.
- [32] C. Schmid, Weakly supervised learning of visual models and its application to content-based retrieval, *International Journal of Computer Vision* 56 (1–2) (2004) 7–16.
- [33] S.-C. Zhu, C.-E. Guo, Y. Wang, Z. Xu, What are textons?, *International Journal of Computer Vision* 62 (1–2) (2005) 121–143.
- [34] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging* 13 (1) (2004) 146–165.
- [35] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* 27 (8) (2006) 861–874.
- [36] Y. Deng, B. S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (8) (2001) 800–810.
- [37] C. Christoudias, B. Georgescu, P. Meer, Synergism in low-level vision, in: *Proceedings of 16th IAPR International Conference on Pattern Recognition, Vol. 4, Quebec City, Canada, 2002*, pp. 150–155.
- [38] M. Haindl, S. Mikes, Texture segmentation benchmark, in: *Proceedings of 19th IAPR International Conference on Pattern Recognition, Tampa, Florida, 2008*.
- [39] S. Lee, Y. Liu, PSU near-regular texture database, <http://vivid.cse.psu.edu/texturedb/gallery/> (2009).
- [40] C. Belezni, H. Bischof, Fast human detection in crowded scenes by contour integration and local shape estimation, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida, 2009*, pp. 2246–2253.