EDGE-BASED TEXTURE GRANULARITY DETECTION

Haoyi Liang and Daniel S. Weller
University of Virginia, Department of ECE, Charlottesville, VA, 22904, USA

***Abstract*—Directly connected to the texture appearance, texture granularity is an effective measurement for geographic resources classification, product quality monitoring and image compression ratio selection. However, the application of existing works on texture granularity is limited by intense computation and the dependence on empirically selected parameters that vary among different textures. This paper proposes an edge-based texture granularity detection algorithm that takes textures as homogeneous cells separated by prominent boundaries. Experiments on two datasets show that the proposed method yields granularity consistent with perceptual measures and is highly computationally efficient compared to existing methods.**

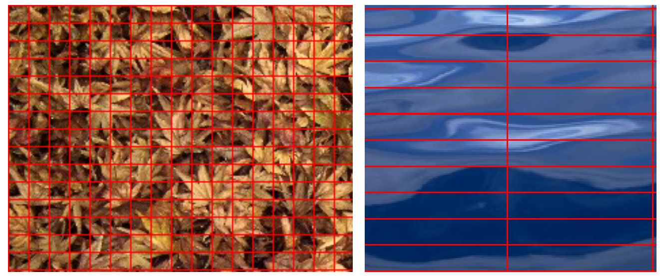
***Index Terms*—**texture, texture granularity, edge, scale space

I. INTRODUCTION

The texture granularity is a fundamental texture property describing the size of texture primitives, which are basic cells composing a texture [1]. Compared with other statistical descriptors like local binary pattern (LBP) [2] and its high dimensional variants [3]–[5], texture granularity only contains two numbers, the length and the width of the texture primitive. Therefore, the texture granularity is an inherently weak descriptor in terms of classification. However, the texture granularity is more closely associated with perception and has its own unique applications. First, the texture granularity plays an important role in characterizing various objects. For instance, the texture granularity in satellite images is valuable to distinguish different objects, like crops, buildings and roads; geologists identify minerals [6] with the help of the mineral granularity; the carpet granularity is used as an indicator of the wearing status for quality monitoring [7]. Second, texture granularity is an essential component of image quality perception [8]–[10]. For example, images with coarse texture granularity usually are more sensitive to noise [8], [9], [11] and texture granularity is highly related to the quality of compressed images [12].

However, texture granularity detection is nontrivial because texture granularity is usually defined as the average size of texture primitives. The variations of texture primitives in rotation, shape, size and color together make the definition of the texture primitive challenging. For example, the texture primitive of "maple" in Fig. 1 can be the whole maple leaf or the saw tooth in a maple leaf; in "wave", ripples vary so significantly in size and shape that a single texture primitive would lack meaning.

Existing texture granularity methods mainly follow one of two approaches: segmentation of the original texture or analysis of downsampled textures. One method [15] uses mean shift segmentation [16] to divide the texture image into connected regions with similar gray values. Through testing different clustering thresholds, the image is segmented into regions that are on the scale of the texture granularity. The main issue in [15] is that it directly aims at segmenting the texture primitives and suffers from the considerable computation involved in segmentation. In a more recent work [12], texture granularity is calculated by identifying the peaks in down-sampled texture images and counting the distance between these peaks both horizontally and vertically. The key step in [12] is determining the down-sampling factor. An iterative down sampling process continues as long as the structural similarity (SSIM) index [17] between the down-sampled image and the original image is above an threshold. However, the empirical selection of this threshold limits the application of [12].



(a) maple [13] (b) wave [14]

Fig. 1: Granularity detection is a difficult problem partially because it is a subjective quantity. In "maple", both the leaf and the tooth of the leaf are reasonable choices of the texture primitive. The proposed algorithm favors the tooth texture primitive because leaves are heavily occluded. In "wave", the texture primitive is hard to explicitly define. The detected granularity captures the average dimension of ripples.

Different from existing methods, our approach to measuring the texture granularity is based on edges. Edges, as the concise binary form of the gradient, represent the boundaries of texture primitives and thus are suitable to extract the texture granularity. In our method, the first step is an adaptive Canny edge detection that removes the edges corresponding to fine structures while preserving the boundaries of texture primitives; next, the texture granularity is approximated by analyzing the length of the main edges. Compared with existing algorithms, the proposed one involves less computation and relies less on empirically selected parameters.

The rest of the paper is organized as follows. Section II motivates our algorithm and elaborates on its implementation. Experiments and analysis are provided in Section III. Section IV concludes this work and discusses possible future directions.

II. Edge-based granularity detection

Because texture images are composed of smooth texture primitives separated by prominent boundaries, the strong edges actually characterize the texture pattern. We base our algorithm on edges and model the optimal edge map, $mask^*$, that contains the boundaries of texture primitives, in the following equation,

$$mask^* = \underset{mask}{\operatorname{argmin}} \sum_{i=1:M} \|D(S^i_{mask})\|_2^2 + \lambda \|mask\|_0. \quad (1)$$

In (1), $mask$ denotes the binary edge mask separating the texture image, S^i_{mask} is the i^{th} connected region in $mask$, M is the number of connected regions, D stands for the gradient operator, λ is the regularization parameter to limit the total length of boundaries in $mask$. Eqn. (1) aims at minimizing the summation of total variation in each connected regions in $mask$ and constraining the total edge length. If x is the original texture image, (1) can be simplified into,

$$mask^* = \underset{mask}{\operatorname{argmin}}\, C - \|mask \cdot Dx\|_2^2 + \lambda \|mask\|_0. \quad (2)$$

where $C = \|Dx\|_2^2$. It is easy to see that (2) equals to,

$$mask^* = \underset{mask}{\operatorname{argmax}}\, \|mask \cdot Dx\|_2^2 - \lambda \|mask\|_0. \quad (3)$$

Our algorithm first filters out fine details with an adaptively selected λ and $mask^*$, and then estimates the texture granularity on layers where the boundaries of texture primitives remain.

A. Optimal Edge Map

Texture primitives are perceived because edges separating texture primitives have the greatest gradients. This fact justifies using simple edge detectors, such as Canny [18], to closely approximate $mask^*$ with proper parameters. By limiting $mask$ to be the output of an edge detector, the candidates of $mask^*$ are dramatically narrowed. Because edges obtained by a Canny detector are continuous, we adopt the Canny detector to approximate $mask^*$. The other remaining task is choosing λ in (3). In fact, if the we fix the edge detector, λ in (3) is related to edge detector parameters, t. If the parameters of the edge detector are properly set, the detected edge map is close to $mask^*$. In our implementation, t is chosen by

$$t^* = max_t \frac{\partial f(t)}{\partial t}, f(t) = \|mask_t \cdot Dx\|_2^2. \quad (4)$$

In our implementation, t is sampled by discrete values and a scale space of $mask$ is built with gradually increasing t. Therefore, Eqn. (4) means that the interval including t^* contains the greatest total variation. The rationale for this is that the edges with dominant gradient correspond to the boundaries of texture primitives and the smoothness within texture primitives concentrates the gradient amplitudes on primitive boundaries. In our method, the variance of the blurring kernel of Canny, σ, and the ratio between the high threshold and low threshold, r, are fixed as $\sigma = \sqrt{2}$, $r = 0.7$. Therefore, the only parameter of Canny detector is the high threshold, T_h. The edges disappear gradually as T_h increases. The edges disappearing before t^* are mainly the fine structures. Although the total length of these fine textures are large, their gradient amplitudes are small. The edges disappearing after t^* correspond to the boundaries of texture primitives but the total length of disappearing edges is decreasing. Thus, Eqn. (4) enables the Canny detector adaptively chooses a proper parameter to detect boundaries of texture primitives. Fig. 2 illustrates these three phases for the texture image "brick" shown in Fig. 5.

The implementation of the first step in our algorithm is Alg. 1. The pixel values are first normalized to $[0, 1]$. The number of T_h, N, is set as 50. The binary output of the Canny detector with high threshold, $T_{h(i)}$, is e_i. The disappearing edges, de_i, from the $T_h(i)$ to $T_h(i+1)$ is the exclusive disjunction of e_i and e_{i+1}.

Algorithm 1 Optimal Edge Map Detection

Initialization;
  Normalize input image x to $[0, 1]$
  $T_h(i) = i/N, i = 1, \ldots, N$

Scale Space;
  $e_1 = Canny_{T_h(1)}(x)$
  for i=2:N **do**
    $e_i = Canny_{T_h(i)}(x)$
    $de_i = XOR(e_i, e_{i-1})$
  end for

Scale Space analysis;
  $t^* = \max_{i, i=2:N} \|de_i \cdot Dx\|_2^2$

B. Texture Granularity Poll

Because texture primitives are smooth patches and disappearing edges at each step have similar gradient amplitudes, each disappearing edge after t^* naturally ends at the intersection of texture primitives. However, disappearing edges after t^* can not be directly used to estimate the texture granularity because one edge may correspond to boundaries of multiple texture primitives. Before calculating the texture granularity, disappearing edges at each step are pruned by the connectedness of each pixel. The connectedness of a pixel is the number of edge points in its 8-connected neighborhood. Pixels whose connectedness are larger than or equal to 4 are removed to break the edges that span multiple texture primitives. A histogram of the pruned edges is then taken to approximate the texture granularity.

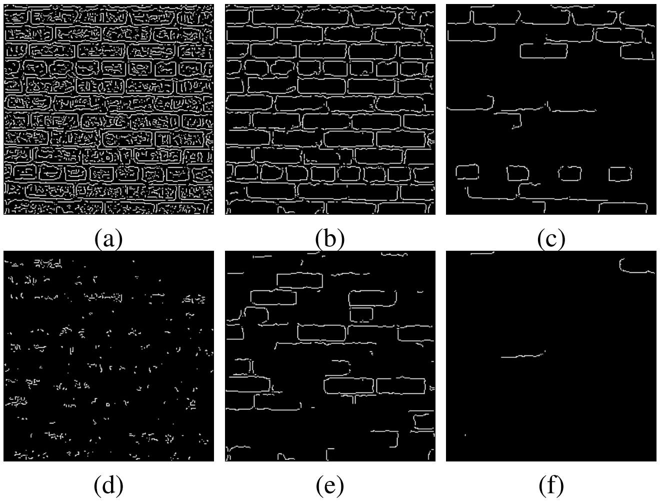


Fig. 2: (a): Edges before t^* contain many fine structures in the bricks. (b): Edges with t^* reflect the boundaries of bricks. (c): Edges after t^* are parts of the boundaries. (d) - (f): Disappearing edges corresponding to (a) - (c)

Algorithm 2 Texture Granularity Poll

Initialization;
$hist_x = 0, hist_y = 0$

Polling;
for $i = t^* : N$ **do**
Remove points whose connectedness are equal or larger than 4 in de_i
for each edge, l, in de_i **do**
if $xlen(l) > ylen(l)$ **then**
$hist_x(xlen(l)) = len(l)$
else
$hist_y(ylen(l)) = len(l)$
end if
end for
end for
$gra_x = median(hist_x)$
$gra_y = median(hist_y)$

In Alg. 2, $xlen(l)$ and $ylen(l)$ are spans of edge, l, on x and y directions respectively, $len(l)$ is the length of l, $hist_x$ and $hist_y$ record the length distribution of edges on two directions. Each edge can only contribute to the distribution of its main direction because the span of an edge on its main direction reflects the size of the texture primitive on one dimension. Finally, medians of these two distributions are taken as the texture granularity.

III. Experiments

We conduct two experiments to verify the effectiveness of our algorithm. First we compare the proposed algorithm with the segmentation-based work [15] and show that our algorithm reduces computation considerably, while producing more perceptually satisfying results. In the second experiment,

TABLE I: Computation efficient comparison

	brick	grass	gravel2	red_cloth	sand	wave
MS-Granularity	9.9s	9.9s	9.9s	14.8s	6.0s	13.81s
E-Granularity	**0.62s**	**0.67s**	**0.64s**	**0.67s**	**0.74s**	**0.60s**

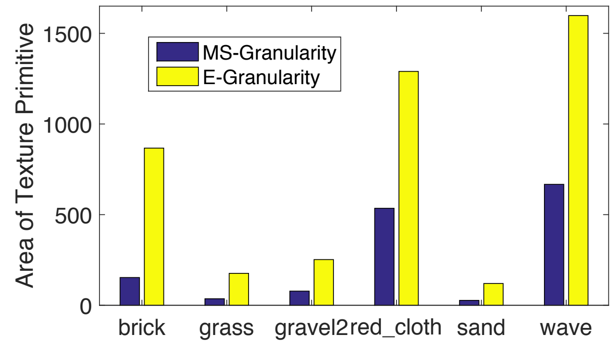
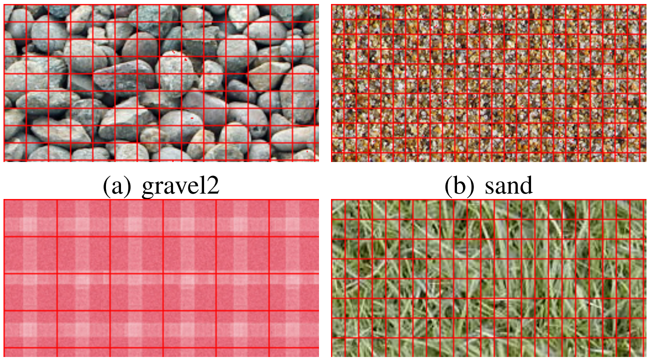


Fig. 3: Areas of the texture primitive by MS-Granularity and E-Granularity. The coherence between MS-Granularity and E-Granularity is 0.95, and rank orders by two methods are same.

we test our algorithm on the Ponce dataset [14] that contains six sets of texture images, each containing the same content but on different scales. In the following experiment, we refer to the method in [15] as MS-Granularity and the proposed method as E-Granularity. All experiments are run on a machine with Intel Core i7-4770 CPU of 3.40GHz and 16.0 GB RAM. The C++ codes of MS-Granularity are provided by Dosselmann [15] and E-Granularity is implemented using MATLAB.

A. Comparison with MS-Granularity

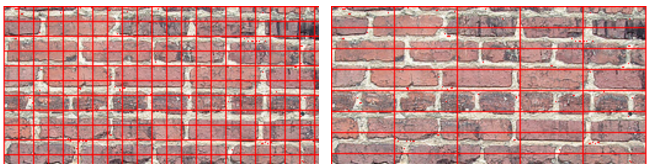
In [15], MS-Granularity properly ranks six texture images according to the subjective study of perceived granularity levels. Because MS-Granularity evaluates the texture granularity by area, we use the product of length and width granularity detected by E-Granularity for comparison. Fig. 3 shows that E-Granularity also properly ranks six textures by texture granularity, but the values of texture granularity detected by MS-Granularity and E-Granularity are different. In order to provide an intuitive impression about detected texture granularities, Fig. 4 shows gridded sample textures by E-Granularity. In "gravel2", the detected granularity is about a quarter of the full size of a gravel. This is because that due the occlusion, the contrast on the gravel edges changes roughly every half of the full gravel size. This agrees with human perception that the occlusion makes the texture more disordered and thus decreases the texture granularity. Other samples by E-Granularity also reflect perceived texture granularities. One comparison between MS-Granularity and E-Granularity is shown in Fig. 5. Since the output of MS-Granularity is the average area of texture primitives, the width and length of grids in Fig. 5 (a) are the square roots of the MS-Granularity output. In "brick", the fine wrinkles on the surface of bricks form a detailed texture, but the most impressive texture is characterized by the edges between bricks. Because the segmentation-based MS-Granularity is interfered by the minor patches within bricks and the crevices between bricks,



(a) gravel2 (b) sand

(c) red_cloth (d) grass

Fig. 4: Granularity detection samples by E-Granularity.



(a) Texture granularity detected by MS-Granularity (b) Texture granularity detected by E-Granularity

Fig. 5: Granularities detected by different methods on “brick”.

the granularity detected by MS-Granularity shown in Fig. 5 (a) is much smaller than the perceived size of bricks. The Optimal Edge Map detection model ensures E-Granularity successfully captures the texture formed by bricks in Fig. 5 (b).

The computation time of the two methods is shown in Table I. E-Granularity turns to be much faster that MS-Granularity because edge detection is a more computationally efficient operation than mean-shift segmentation.

B. Ponce dataset

In order to avoid the subjective discrepancy of texture granularity, we verify our algorithm on images with the same texture content but on different scales in this part. The Ponce dataset [14] provides six sets of such textures. Granularities detected by E-Granularity are shown in Table II and gridded textures are shown in Fig. 6. It is clear that the granularities decrease as the textures move to finer scales.

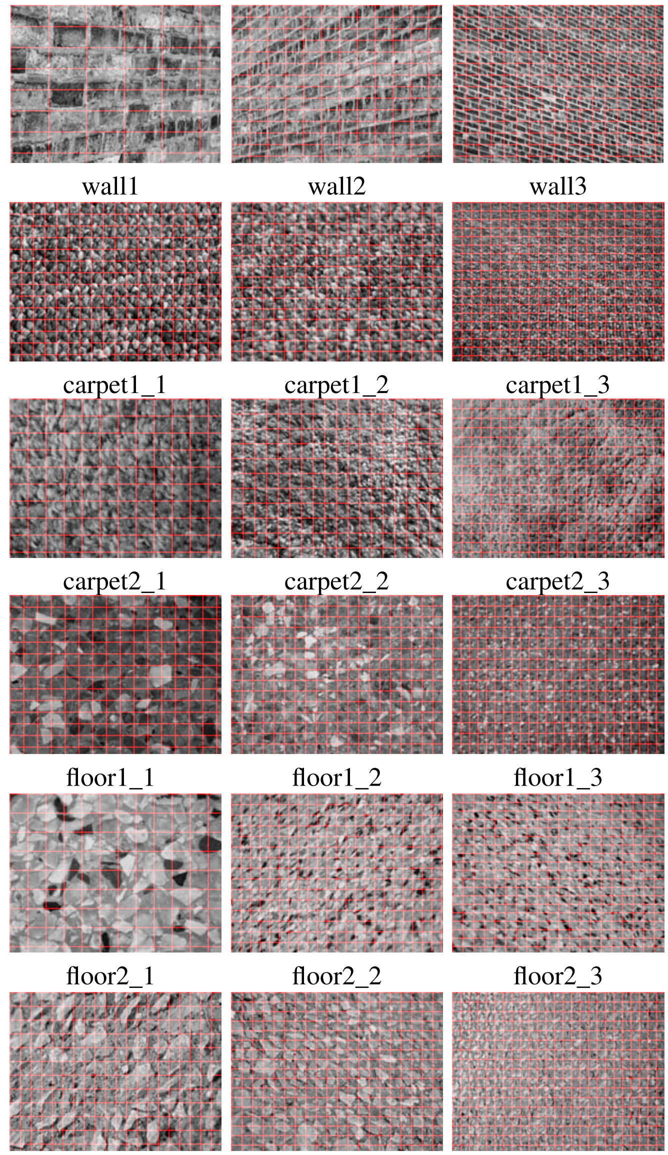
Among the textures in Fig. 6, “wall2” and “wall3” reveal an interesting phenomenon: the texture granularity in well-organized textures is related to the texture direction. The bricks in “wall3” are smaller in both height and width, but due to the higher angle of the bricks in “wall3”, the height of the texture granularity in “wall2” and “wall3” are the same.

IV. Conclusions

By exploiting the homogeneity within texture primitives and the prominence of boundaries, we proposed a novel edge-based texture granularity detection algorithm in this paper. Compared with previous works, our method obtains satisfying results while saving considerable computation. In order to further verify the effectiveness of the proposed method, we conducted experiments on a dataset with gradually size-changing textures and the results agreed with human perception.

TABLE II: Detected granularities of textures from Ponce

image	unit size	image	unit size	image	unit size
wall1	20×36	carpet2_1	16×17	floor2_1	18×17
wall2	11×17	carpet2_2	14×11	floor2_2	10×9
wall3	11×12	carpet2_3	9×9	floor2_3	8×10
carpet1_1	11×11	floor1_1	11×13	pebbles_1	13×10
carpet1_2	10×10	floor1_2	10×10	pebbles_2	9×10
carpet1_3	8×9	floor1_3	8×10	pebbles_3	8×9



wall1 wall2 wall3

carpet1_1 carpet1_2 carpet1_3

carpet2_1 carpet2_2 carpet2_3

floor1_1 floor1_2 floor1_3

floor2_1 floor2_2 floor2_3

pebbles_1 pebbles_2 pebbles_3

Fig. 6: Texture granularities detected by E-Granularity reflect the changing primitive size in each set of textures.

With the high efficiency of the proposed method, we plan to extend the proposed method to natural scene images with spatially varying texture granularities. Measuring the local texture granularity of a natural scene image is valuable to many image processing problems, such as image quality assessment, image compression and image recovery.

REFERENCES

[1] R. M. Haralick, "Statistical and structural approaches to texture," in *Proceedings of the IEEE*, 1979, vol. 67, pp. 786–804.

[2] T. Ojala, M. Pietika, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–988, Jul. 2002.

[3] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. on Image Process.*, vol. 19, no. 6, pp. 1657–1663, 2010.

[4] A. Fathi and A. Naghsh-Nilchi, "Noise tolerant local binary pattern operator for efficient texture analysis," in *Pattern Recognit. Letters*, 2012, pp. 1093–1100.

[5] G. Schaefer and N. P. Doshi, "Multi-dimensional local binary pattern descriptors for improved texture analysis," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 2500 – 2503.

[6] L. Lepisto, I. Kunttu, J. Autio, and A. Visa, "Rock image retrieval and classification based on granularity," in *5th Int. Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[7] L. H. Siew, R. M. Hodgson, and E. J. Wood, "Texture measures for carpet wear assessment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 1, pp. 92–105, Jan. 1988.

[8] A. Samet, M. A. B. Ayed, N. Masmoudi, and L. Khriji, "New perceptual image quality assessment metric," *Asian Journal of Information Technology*, pp. 996 – 1000, 2005.

[9] S. Winkler, *Digital Video Quality: Vision Models and Metrics*, 2005.

[10] A. R. Rao and G. L. Lohse, "Identifying high-level features of texture perception," *Graphical Models and Image Processing*, vol. 55, pp. 218–233, May 1993.

[11] W. Zuo, L. Zhang, C. Song, and D. Zhang, "Texture enhanced image denoising via gradient histogram preservation," in *Proc. Computer Vision and Pattern Recognition*, 2013, pp. 1203–1210.

[12] M. M. Subedar and L. J. Karam, "A no reference texture granularity index and application to visual media compression," in *Proc. Int. Conf. Image Process.*, 2015, pp. 750– 755.

[13] "https://en.wikipedia.org/wiki/Image_texture," *Wikipedia*.

[14] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[15] R. Dosselmann and X. Yang, "Mean Shift Particle-Based Texture Granularity," in *IEEE Int. Instrumentation and Measurement Tech. Conf.*, 2008, pp. 101–106.

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[18] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986.