Computational Visual Media
https://doi.org/10.1007/s41095-018-0106-z Vol. 4, No. 2, June 2018, 173–184
Research Article

Automatic texture exemplar extraction based on global and local textureness measures

Huisi Wu[1], **Xiaomeng Lyu**[1], **and Zhenkun Wen**[1](✉)

Abstract Texture synthesis is widely used for modeling the appearance of virtual objects. However, traditional texture synthesis techniques emphasize creation of optimal target textures, and pay insufficient attention to choice of suitable input texture exemplars. Currently, obtaining texture exemplars from natural images is a labor intensive task for the artists, requiring careful photography and significant post-processing. In this paper, we present an automatic texture exemplar extraction method based on global and local textureness measures. To improve the efficiency of dominant texture identification, we first perform Poisson disk sampling to randomly and uniformly crop patches from a natural image. For global textureness assessment, we use a GIST descriptor to distinguish textured patches from non-textured patches, in conjunction with SVM prediction. To identify real texture exemplars consisting solely of the dominant texture, we further measure the local textureness of a patch by extracting and matching the local structure (using binary Gabor pattern (BGP)) and dominant color features (using color histograms) between a patch and its sub-regions. Finally, we obtain optimal texture exemplars by scoring and ranking extracted patches using these global and local textureness measures. We evaluate our method on a variety of images with different kinds of textures. A convincing visual comparison with textures manually selected by an artist and a statistical study demonstrate its effectiveness.

Keywords texture exemplar extraction; textureness; GIST descriptor; binary Gabor pattern (BGP)

1 Introduction

In the booming virtual reality industry, texture synthesis techniques play an important role in modeling and providing visual textures. For example, texture synthesis is heavily used in generating backgrounds for virtual reality scenes. In particular, exemplar-based texture synthesis is popular as it can quickly generate impressive textures of arbitrary sizes and shapes from a small exemplar, as shown in Fig. 1(a). Various exemplar-based texture synthesis algorithms [1–3] have been proposed in the last two decades, and encouraging improvements in both quality and efficiency of exemplar-based texture synthesis have been presented. Currently, it is easy to generate a texture with desired variation in scale or shape using existing exemplar-based texture synthesis techniques. However, the quality of the input texture exemplar has a strong impact on the final texture synthesis results. Without suitable high-quality texture exemplars as input, users cannot easily obtain a high-quality texture result. Unfortunately, automatically creating texture exemplars (see Fig. 1(b)) from natural images is still a labor intensive task for artists, requiring careful photography, cropping, and significant post-processing.

Most traditional exemplar-based texture synthesis

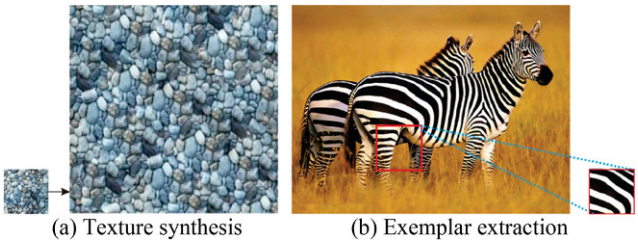

Fig. 1 Texture synthesis and exemplar extraction.

1 Shenzhen University, Shenzhen 518000, China. E-mail: H. Wu, hswu@szu.edu.cn; X. Lyu, luxiaomeng2016@email.szu.edu.cn; Z. Wen, wenzk@szu.edu.cn (✉).

Manuscript received: 2017-12-25; accepted: 2017-12-31

techniques emphasize optimality of generated textures (they should be seamless in color, match in gradient and feature domains, etc.) and efficiency. They typically pay insufficient attention to obtaining ideal exemplars from natural images, and little work on automatic texture exemplar extraction is reported in the literature [4, 5]. Although several algorithms have been proposed for extracting dominant textures from an image [6–8], automatic texture exemplar extraction systems for synthesis applications are still lacking. Artists typically can only acquire exemplars manually by a process of image cropping and careful post-processing, which is both labor intensive and tedious, especially when many exemplars are needed to create complex virtual scenes.

In this paper, we present an automatic texture exemplar extraction method based on global and local textureness measures. Our method first performs Poisson disk sampling to efficiently perform dominant texture identification, randomly and uniformly cropping a number of patches from a natural image. For global textureness assessment, we employ SVM prediction (trained on the UIUC database) on the cropped patches to differentiate textured patches from non-textured patches, based on GIST descriptors. We further measure the local textureness of a patch by extracting and matching the local structure (using BGP) and dominant color features (using a color histogram). This allows identification of suitable texture exemplars consisting solely of the dominant texture. The final optimal texture exemplars are obtained based on both global and local textureness measures by scoring and ranking the extracted patches.

We evaluate our method on a variety of images with different kinds of textures. A visual comparison with textures manually selected by an artist and a statistical study demonstrate its effectiveness.

2 Related work

In the last two decades, a number of texture synthesis methods [9–12] have been presented for texture synthesis, relying on optimizing the target texturing effect (it should be seamless in color or gradient domains). Turk [13] gave a sophisticated algorithm to synthesize a texture on a geometric model, which may have irregular deformations on the surface. Liu et al. [14] proposed a user-assisted texture synthesis method based on modeling the target geometry deformation, lighting, and color with a set of near regular lattices, allowing texture synthesis with varying effects. Karthlkeyani et al. [15] paid more attention to the regularity of the synthesized target textures, controling the regularity of the appearance of the target texture using simple parametric models. Lin et al. [16] provided a survey which analyzed the regularity of textures and proposed a classification algorithm to distinguish regular from irregular textures.

More recently, several researchers have considered evaluating the quality of different texture synthesis methods, and explored optimal combinations of existing methods. As a result, target texture-driven methods are still the most popular research direction for texture synthesis. Noting that existing methods often break boundary structure continuity between adjacent patches, Wu and Yu [10] developed an algorithm to maintain boundary structures by feature matching and alignment. Latif-Amet et al. [17] detected defects encountered in textile images and optimized results based on wavelet theory and co-occurrence matrices. Dai et al. [18] evaluated the quality of a texture based on a set of target texture properties.

Unlike the above texture synthesis methods which mainly consider the output textures, other researchers have paid attention to extracting the dominant textures in an image. Lu et al. [6] first employed diffusion distance manifolds to identify the dominant textures in an input image, but their method is quite time-consuming, taking about 18 minutes to process an image of size 125×94. Wang and Hua [7] proposed a faster dominant texture extraction algorithm based on multi-scale hue–saturation–intensity histograms, but it may fail when the main colors in the dominant texture and the outliers are similar. Similarly, Lockerman et al. [4] proposed a fast iteration method using diffusion manifolds to locate textures in unconstrained images, requiring user input to specify the initial location and scale of the desired texture. In addition, Lockerman et al. [8] presented an unsupervised method for extracting good textures from natural images. Moritz et al. [5] suggested employing local histogram matching to extract textures from input photographs. However, these dominant texture extraction algorithms usually

require the target textures to cover the majority of the image, as shown by the results in Refs. [5–8]. As they mainly focus on extracting the dominant texture, optimal texture exemplar patches containing a number of textures are not always extracted as the final results.

In this paper, we present a novel system to accurately extract optimal texture exemplars from natural images. Little existing work reports auto-extraction of source texture exemplars. We emphasize the importance of the exemplar in example-based texture synthesis.

3 Method

3.1 Overview

An overview of our system is given in Fig. 2. To efficiently and uniformly crop the dominant texture, we first perform Poisson disk sampling [19] within the given image. To compute a global textureness measure, we perform GIST feature extraction based on the UIUC database [20], and train a linear vector collection (LVC) model using SVM to measure the global textureness for an image patch. Furthermore, we also extract the local structure (using BGP) and match dominant color features (using a color histogram) to measure the local textureness of a patch. Finally, real texture exemplars consisting solely of the dominant texture are identified by scoring and ranking both global and local textureness measures for each extract patch.

3.2 Global textureness measure

Given the cropped image patches, we perform scene classification to differentiate textured patches from non-textured patches, based on a global textureness measure. This is a high level measure in which each image patch is treated as a whole (at the patch level). We use GIST features [21] for patch classification. As they contain enough information to identify the scene in a low-dimensional representation of the image, GIST features can extract coarse information from images in a similar way to human vision. Specifically, GIST feature values are calculated using image convolution and mean low level feature values for patches, so obviously provide effective global

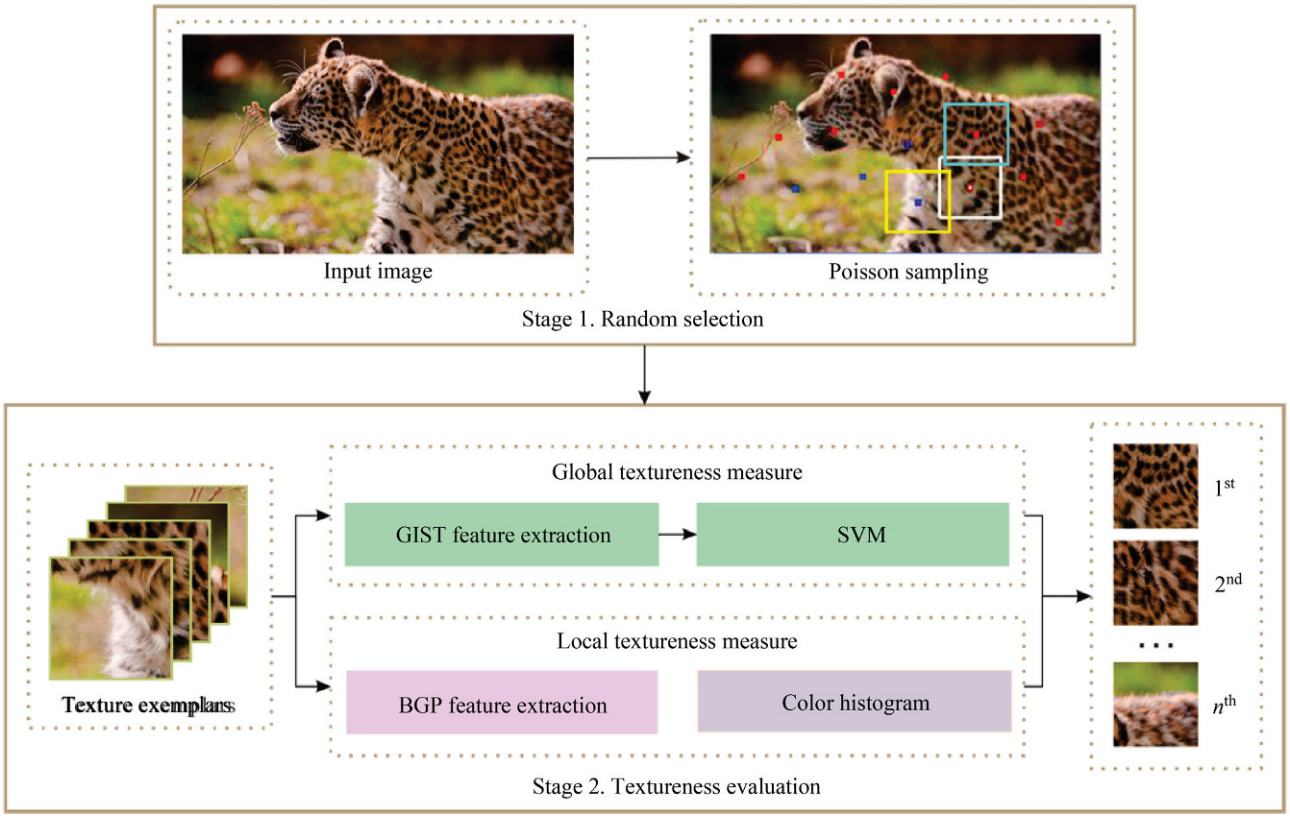

Fig. 2 System overview.

features for a textureness measure. After computing the Fourier transform of the input image, we can obtain the GIST descriptor using K Gabor filters with different directions and scales. The final score of the GIST feature is the average result of image convolution. Detailed operation of GIST feature extraction is shown in Fig. 3.

Given an input image $f(x, y)$ with a resolution of $h \times w$, we convolve it with a Gabor filter with n_c channels . The GIST feature vector is then obtained by cascading the eigenvectors as follows:

$$G(x, y) = \underbrace{\text{cat}}_{n_c}(f(x, y) * g(x, y)) \tag{1}$$

where n_c is the product of the number of different directions with the number of different scales of Gabor filters. cat() represents the cascade operator, $g(x, y)$ represents the Gabor filters, and $*$ is the convolution operation.

We also train a linear SVM [22], which is a popular machine learning method for this texture classification in computer vision. We used the UIUC texture database and the 15-scene dataset to train a classifier to distinguish textures, using the GIST descriptors as features. We can use the SVM's output to assess the global textureness for each image patch, as shown in Fig. 3.

3.3 Local textureness measure

The GIST descriptor is useful for assessing global features, but lacks local information and color information. Thus, we define a local textureness measure to assess the locally detailed textureness for sub-regions (at pixel level) of each patch. For improved local features to measure the differences in local textureness, we apply BGP to extract structural texture features for each patch. BGP is a rotationally invariant texture representation scheme. As BGP uses differences between two regions instead of two individual pixels, it is much more robust than local binary pattern (LBP) [23].

Firstly, we apply Gabor filters to the image patches to perform BGP feature extraction. 2D Gabor filters [24] measure characteristics in both space and frequency domains, so are well suited to describing local structural information which corresponds to spatial frequencies (scale), location, and direction. 2D Gabor filters usually have even-symmetry and odd-symmetry, and can be expressed as

$$g_e(x, y) = \exp\Big[-\frac{1}{2}\Big(\frac{x'^2}{\sigma^2} + \frac{y'^2}{(\gamma\sigma)^2}\Big)\Big]\cos\Big(\frac{2\pi x'}{\lambda}\Big) \tag{2}$$

$$g_o(x, y) = \exp\Big[-\frac{1}{2}\Big(\frac{x'^2}{\sigma^2} + \frac{y'^2}{(\gamma\sigma)^2}\Big)\Big]\sin\Big(\frac{2\pi x'}{\lambda}\Big) \tag{3}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$.

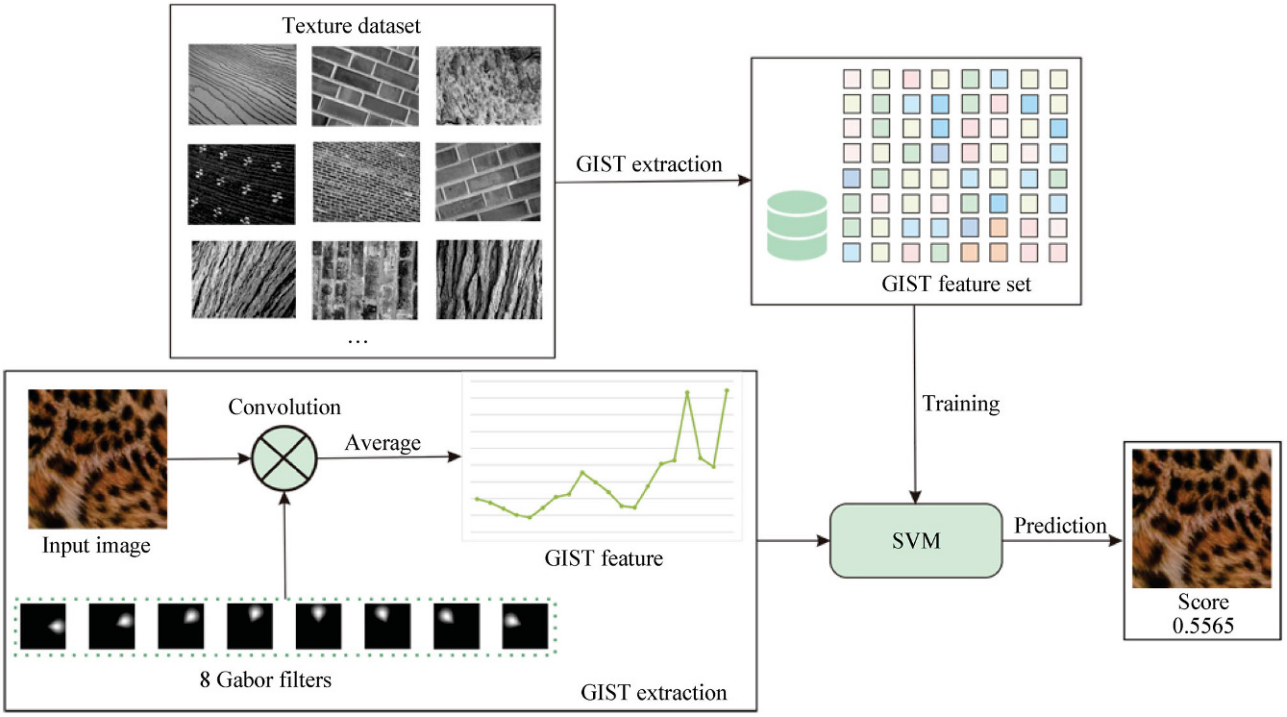

Fig. 3 Global textureness measure.

λ gives the frequency of the sinusoidal factor. σ represents the width of the Gaussian envelope and γ is the spatial aspect ratio. θ is the normal to the parallel stripes of the Gabor function. Equations (2) and (3) allow us to choose different directions and scales for the Gabor filters to be convolved with the texture images. We use J Gabor filters with J different orientations expressed as $g_0, \ldots, g_{J-1}$. By applying J Gabor filters to the texture image, we obtain a response vector $r = \{r_j\}(j = 0, \ldots, J-1)$.

The second step is binarization. A binary vector is written as $b = \{b_j\}(j = 0, \ldots, J-1)$, where b_j is either 1 or 0. Based on the binary value b_j and a binomial factor 2^j, a unique BGP can be used to describe the spatial structure of the texture image as follows:

$$B = \sum_{j=0}^{J-1} b_j \cdot 2^j \tag{4}$$

Using Eq. (4) results in 2^J output values. To achieve rotation invariance, we adopt a scheme similar to LBP: we define rotationally-invariant BGP (B_{r}) as

$$B_{\mathrm{r}} = \max\{\mathrm{ROR}(B, j)\}, \quad j = 0, \ldots, J-1 \tag{5}$$

where $\mathrm{ROR}(x, j)$ indicates a circular bitwise right shift of x by j bits. If $J = 8$, this results in 36 different values. We illustrate the calculation in Fig. 4.

Local textureness is the texture property within a patch, and it describes the relationship between structure and color feature of sub-regions consisting to the whole image patch. Texture exemplar should have the similar structural and color information among each sub-region within the patch. To explain the relationship, we compared the whole image patch with its sub-regions. For the structural feature and color information of texture image, we perform BGP and color histogram to extract the whole structure and color features of the image patch. The next step is to segment the patch into a number of sub-patches and we also calculated BGP and color histogram for each sub-patch. Based on above BGP and color histogram in two levels of patches, we perform a similarity calculation on the local textureness measure. The process is as shown in Fig. 5.

We compute BGP feature similarity using cosine distance, which is invariant to the length of the vectors, and can be expressed as

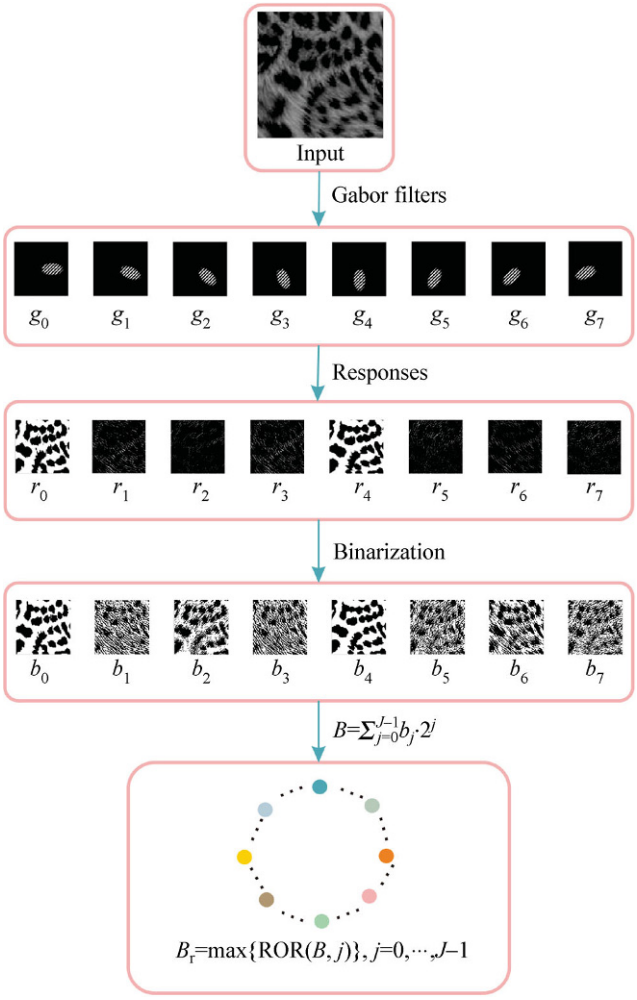

Fig. 4 BGP feature extraction.

$$\cos(x, y) = \frac{xy}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{6}$$

where x and y represent the BGP feature vectors. x_i and y_i are the components of the vectors. Cosine distances lie between 0 and 1. For two feature vectors with high similarity, the distance will be close to 1. To compute the structural texture similarity between the whole texture patch and each sub-patch, we calculate BGP feature similarity between the image patch and its sub-regions. We sum the BGP feature cosine distance for the image patch and each sub-patch as follows:

$$S(w, p, n) = \sum_{p=1}^{n} \cos(B_{\mathrm{r}(w)}, B_{\mathrm{r}(p)}) \tag{7}$$

where S is the similarity distance for BGP features, $B_{\mathrm{r}(w)}$ is the BGP feature for the whole image patch, and $B_{\mathrm{r}(p)}$ is the BGP feature for each sub-patch.

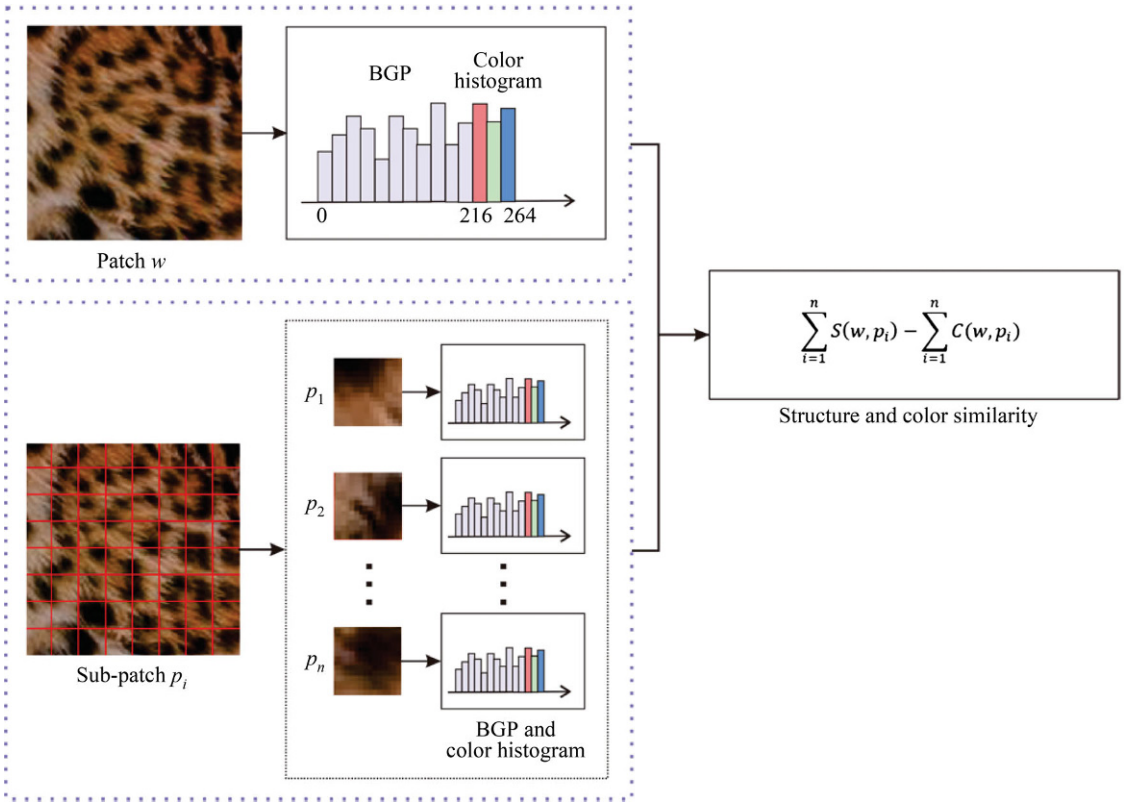

Fig. 5 Local textureness measure.

Corners and edges of the image may lack the desired texture, as illustrated in Fig. 6. We thus apply texture defect detection in our textureness evaluation. By examining a large number of such texture exemplars, we found that all share a common deficiency in their color features. We thus calculate the distances between the color histograms of each sub-patch and the whole image patch, and overcome this problem based on color similarity filtering. If the color histogram distance is large between the whole patch and a sub-patch, we apply a penalty to the local textureness measure. We apply the chi-square measure to calculate the color histogram distance:

$$\chi^2(u, v) = \frac{1}{2} \sum_{i=1}^{n} \frac{(u_i - v_i)^2}{u_i + v_i} \tag{8}$$

where $u_i = u / \sum_{j=1}^{n} u_j$ and $v_i = v / \sum_{j=1}^{n} v_j$, and u and v are the color histograms. As RGB histograms have three color channels, we calculate the similarity using the chi-square distance for each channel and sum them:

$$C(w, p, n) = \sum_{i=1}^{n} \chi^2(R_w, R_{p_i}) + \sum_{j=1}^{n} \chi^2(G_w, G_{p_j}) + \sum_{k=1}^{n} \chi^2(B_w, B_{p_k}) \tag{9}$$

where C represents the color histogram similarity distance, R_w, G_w, B_w represent RGB color histograms for the whole image patch, and R_p, G_p, B_p are the RGB color histograms of each sub-patch.

3.4 Overall textureness evaluation

Using the global textureness measure (see Section 3.2) and the local textureness measure (see Section 3.3),

Fig. 6 Local color deficiencies in texture exemplars.

we formulate the overall textureness T as

$$T = G + S - C \tag{10}$$

where G is the GIST feature score representing the global textureness of the cropped patches, and for the local textureness measure, and S and C represent the inner structure and color similarity between the overall patch and sub-patches. In our experiments, we found that equal weights for G, S, and C provide optimal texture patches comprising the dominant textures in natural images, when finding patches with the highest T scores.

4 Experiments

We have implemented our automatic texture exemplar extraction method using MATLAB R2014a on Windows 10, and evaluated it using hundreds of natural images.

Specifically, we applied our method to natural images collected from the Internet, to demonstrate its effectiveness in texture identification. Our datasets contain different kinds of textures with different resolutions. Typical examples and results are as shown in Fig. 7. To standardize evaluation, all selected input images were resized to a resolution of 800×600. Then, a number of texture exemplars of size 128×128 were cropped based on Poisson disk sampling. For each input image, the five texture exemplars with the highest T scores were collected, as shown in Fig. 7. From the results, we can see that our method provides excellent texture exemplars based on the given natural images; they always include the

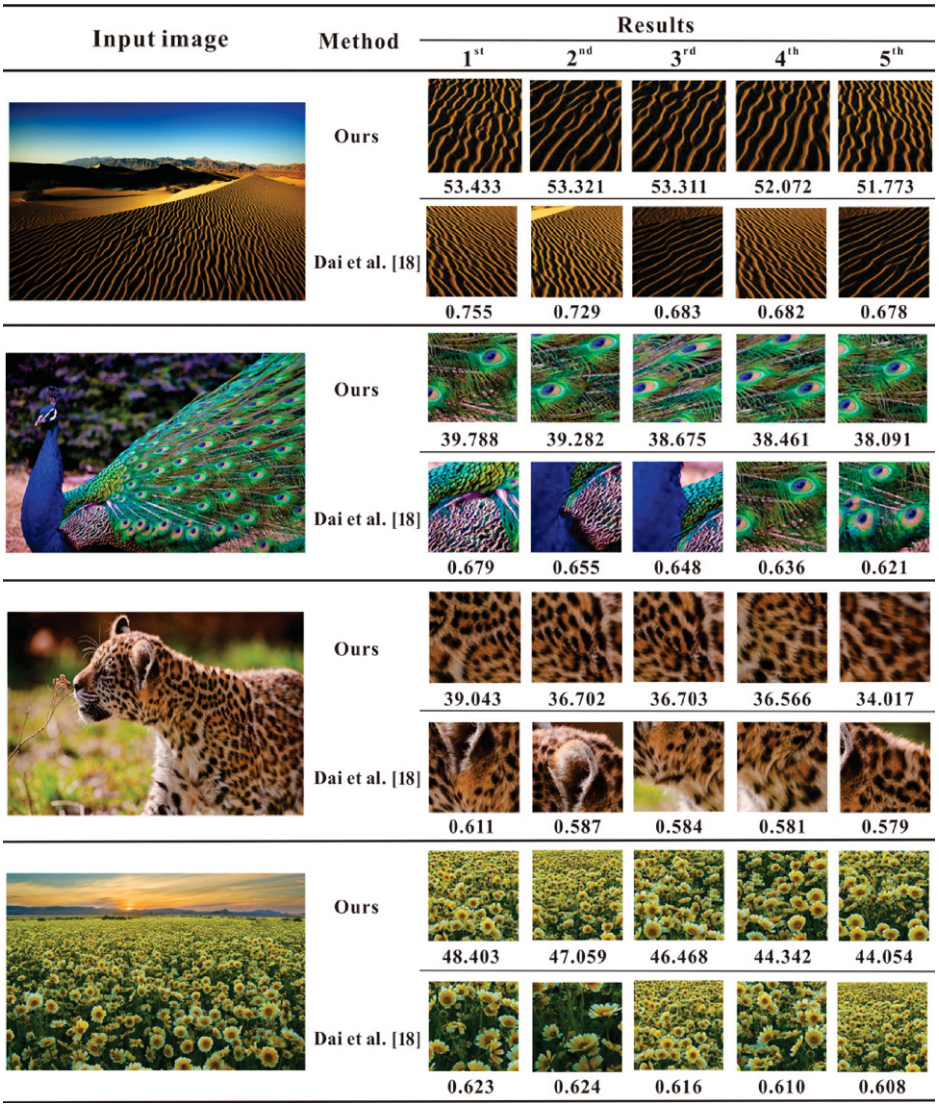

Fig. 7 Patches chosen by our method and that of Dai et al. [18].

dominant textures in the input images.

We also compare our method with two state-of-the-art methods for textureness evaluation. Firstly, we implemented the method proposed by Dai et al. [18] and compared its results with those of our method, as shown in Fig. 7. Both our method and the competitor can extract desirable texture exemplars containing the dominant textures in the input images. Nevertheless, the results in Fig. 7 indicate how our method outperforms the competitor in the scores for the extracted exemplars. As our method can filter out exemplars with deficiencies, better texture exemplars with less non-texture content can be obtained, resulting in higher scores. Dai et al.'s method does not avoid exemplars with deficiencies, e.g., those lacking textured content in the corners.

We have also compared our method with that of Lockerman et al. [4]. Lockerman et al.'s method requires user input to specify the initial location and scale of the desired texture, and employs a fast iteration method using diffusion manifolds to locate textures from unconstrained images. We selected typical images from Lockerman's web page, ran our method on them, and compared the results with Lockerman et al.'s. As shown in Fig. 8 our method also outperforms Lockerman's method in extracting optimal texture exemplars. Our method can extract several meaningful exemplars with different texture contents. As Lockerman et al.'s method mainly focuses on extracting textures for the dominant texture, smaller exemplars were extracted, which do not provide a meaningful exemplar for texture synthesis: optimal texture exemplar patches contain a number of textures. See Fig. 8. More importantly, our method is automatic while Lockerman et al.'s method requires user input to specify the initial location and scale of the desired texture [4].

In addition, we compared our method with textures manually selected by three artists. We instructed them to select a patch which is the best texture exemplar: see Fig. 9. We treat this as ground-truth

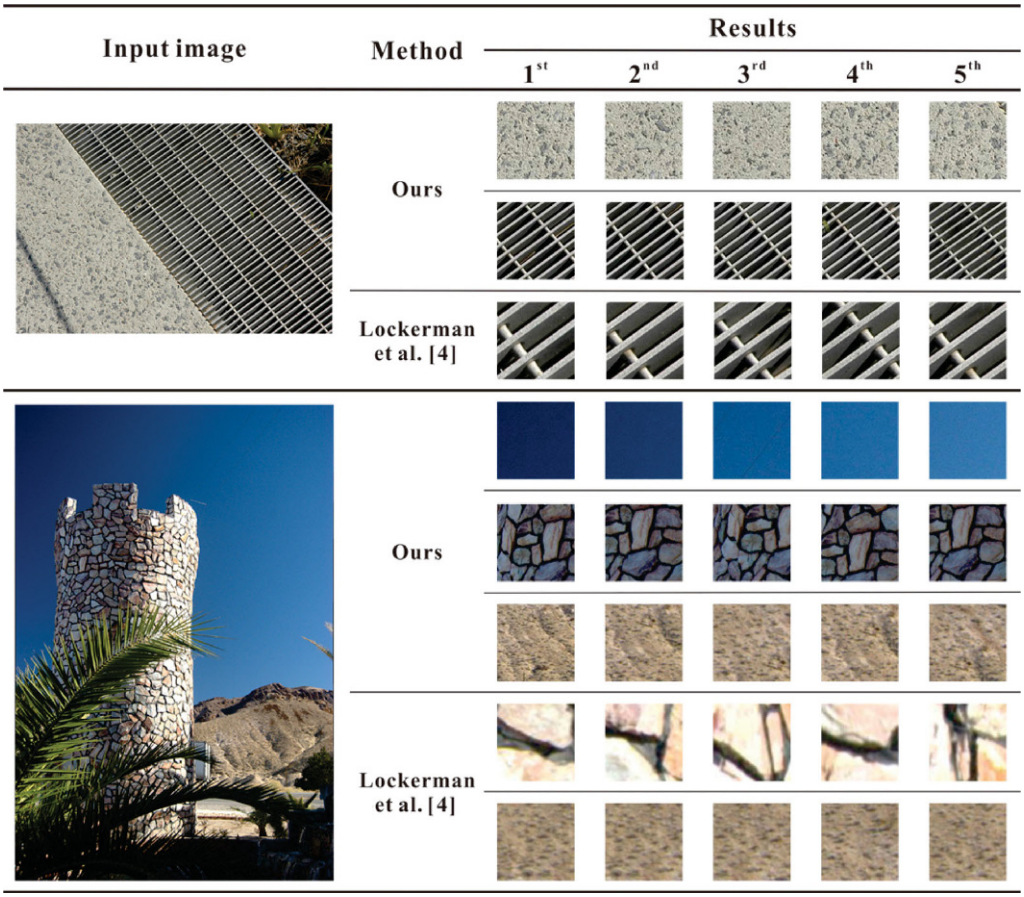

Fig. 8 Patches chosen by our method and that of Lockerman et al. [4]. Input images and the results of Lockerman et al.'s method were obtained from `http://graphics.cs.yale.edu/site/tr1483`.

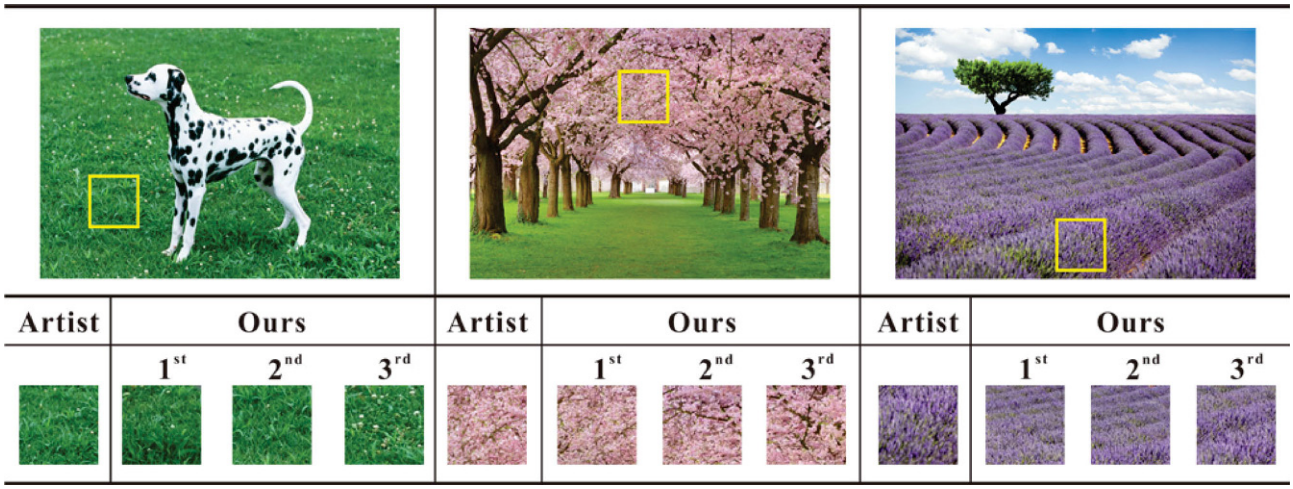

Fig. 9 Patches chosen by our method and those chosen by artists.

and compare it with our results. Figure 9 shows that our method can obtain desirable texture exemplars which are very close to the ground-truth. Due to random selection in Poisson sampling, our final results may be shifted by a few pixels, but they do not include non-textured content.

We also randomly selected 100 natural images, and ran our method and Dai et al.'s method on them in turn. We then asked the artists to choose the satisfactory exemplars. The number of satisfactory exemplars for our method and Dai et al.'s method are plotted as a function of the total number of test images in Fig. 10. Our method outperforms Dai et al.'s method, in that the artists choose more of our exemplars.

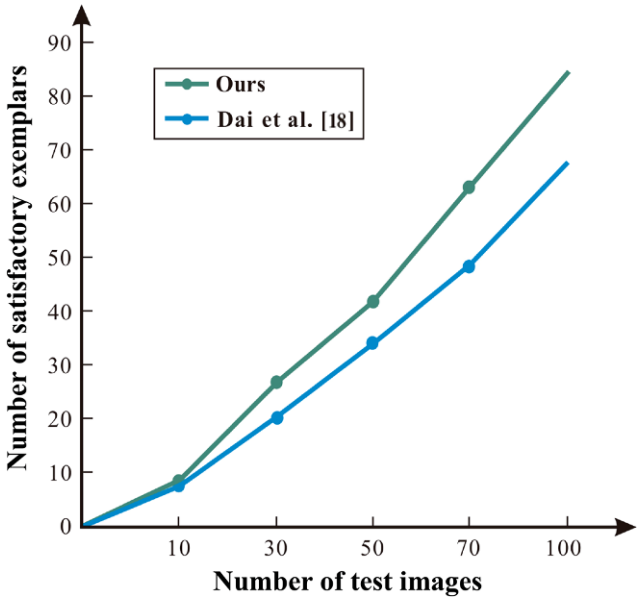

Fig. 10 Statistical comparison between Dai et al.'s method and ours.

To further evaluate the extracted texture exemplars, we created textures with varying resolutions for application in texture synthesis and replacement, as shown in Fig. 11. The results in the fourth and fifth columns in Fig. 11 demonstrate that our extracted texture exemplars can satisfy the requirements of real texture synthesis and replacement applications.

Finally, we timed our method and the competitors' methods. For dominant texture extraction, Lu et al. [6] take 18 minutes to process a 125×94 image. Although Wang and Hua [7] and Moritz et al. [5] give real-time dominant texture extraction algorithms, they require the target textures to covering most of the image. Time for automatic texture exemplar extraction methods (Dai et al.'s and ours) was measured for 800×600 images, for each step of texture exemplar extraction. Table 1 gives these values in ms. As training is done off-line for both methods, we do not include it in Table 1. Timing for Dai et al.'s method includes GIST detection and SVM steps, while our method includes Poisson disk sampling, GIST, BGP, and SVM. We can see that both methods are very fast, and although two more steps are needed for our method, we can still achieve real-time performance.

5 Conclusions

This paper has presented a novel method for automatic texture exemplar extraction based on global and local textureness measures. Unlike

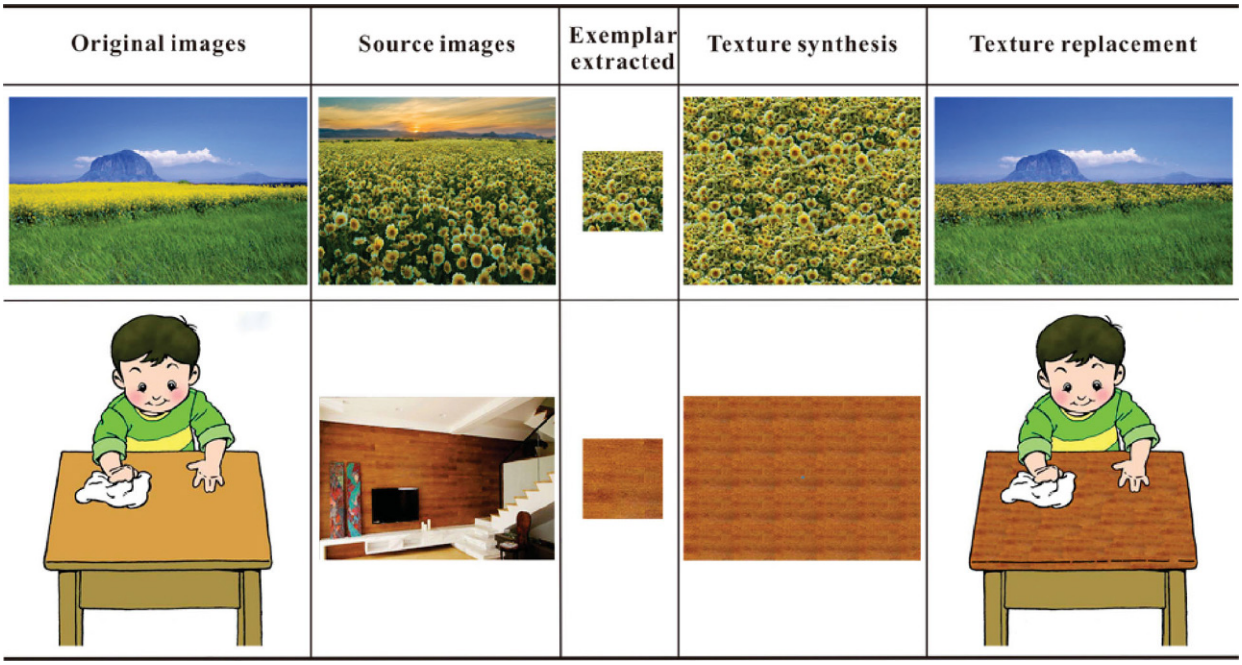

Fig. 11 Applications of texture synthesis and replacements using our extracted texture exemplars.

Table 1 Time (in ms) for our method and that of Dai et al., for 800 × 600 images

Patch size	GIST	SVM	BGP	Poisson disk sampling (R= 64)	Total (Dai et al. [18])	Total (ours)
64×64	65.02	3.19	21.26	2.79	68.21	92.26
128×128	68.68	3.26	22.88	1.25	71.94	96.07
256×256	69.74	3.45	23.45	0.53	73.19	97.17

Total (Dai et al. [18]): GIST+SVM

Total (ours): GIST+SVM+BGP+Poisson disk sampling

traditional methods for example-based texture analysis, our system pays more attention to automatic extraction of texture exemplars based on a textureness evaluation. Our global textureness measure uses SVM training and prediction based on GIST feature extraction from image patches which are uniformly cropped with Poisson disk sampling. Our local textureness measure considers structural and color similarity between patches and sub-patches based on BGP and color histograms. Our method has been validated using a variety of images with different kinds of textures. Comparisons with state-of-the-art methods and with artists' manual selections demonstrate its effectiveness.

Acknowledgements

This work was supported in part by grants from the National Natural Science Foundation of China (Nos. 61303101 and 61572328), the Shenzhen Research Foundation for Basic Research, China (Nos. JCYJ20150324140036846, JCYJ20170302153551588, CXZZ20140902160818443, CXZZ20140902102350474, CXZZ20150813151056544, JCYJ20150630105452814, JCYJ20160331114551175, and JCYJ20160608173051207), and the Start-up Research Fund of Shenzhen University (No. 2013-827-000009).

References

[1] Tartavel, G.; Gousseau, Y.; Peyré, G. Variational texture synthesis with sparsity and spectrum constraints. *Journal of Mathematical Imaging and Vision* Vol. 52, No. 1, 124–144, 2015.

[2] Gatys, L. A.; Ecker, A. S.; Bethge, M. Texture synthesis using convolutional neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 1, 262–270, 2015.

[3] Aguerrebere, C.; Gousseau, Y.; Tartavel, G. Exemplar-based texture synthesis: The Efros–Leung algorithm. *Image Processing on Line* Vol. 3, 223–241, 2013.

[4] Lockerman, Y. D.; Xue, S.; Dorsey, J.; Rushmeier, H. Creating texture exemplars from unconstrained images.

In: Proceedings of the International Conference on Computer-Aided Design and Computer Graphics, 397–398, 2013.

[5] Moritz, J.; James, S.; Haines, T. S. F.; Ritschel, T.; Weyrich, T. Texture stationarization: Turning photos into tileable textures. *Computer Graphics Forum* Vol. 36, No. 2, 177–188, 2017.

[6] Lu, J.; Dorsey, J.; Rushmeier, H. Dominant texture and diffusion distance manifolds. *Computer Graphics Forum* Vol. 28, No. 2, 667–676, 2009.

[7] Wang, W.; Hua, M. Extracting dominant textures in real time with multi-scale hue-saturation-intensity histograms. *IEEE Transactions on Image Processing* Vol. 22, No. 11, 4237–4248, 2013.

[8] Lockerman, Y. D.; Sauvage, B.; Allègre, R.; Dischler, J.-M.; Dorsey, J.; Rushmeier, H. Multi-scale label-map extraction for texture synthesis. *ACM Transactions on Graphics* Vol. 35, No. 4, Article No. 140, 2016.

[9] Efros, A. A.; Freeman, W. T. Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 341–346, 2001.

[10] Wu, Q.; Yu, Y. Feature matching and deformation for texture synthesis. *ACM Transactions on Graphics* Vol. 23, No. 3, 364–367, 2004.

[11] Campisi, P.; Scarano, G. A multiresolution approach for texture synthesis using the circular harmonic functions. *IEEE Transactions on Image Processing* Vol. 11, No. 1, 37–51, 2002.

[12] Fišer, J.; Jamriška, O.; Simons, D.; Shachtman, E.; Lu, J.; Asente, P.; Lukáč, M.; Sýkora, D. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 155, 2017.

[13] Turk, G. Texture synthesis on surfaces. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 347–354, 2001.

[14] Liu, Y.; Lin, W.-C.; Hays, J. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics* Vol. 23, No. 3, 368–376, 2004.

[15] Karthlkeyani, V.; Duraiswamy, K.; Kamalakkannan, P. Texture analysis and synthesis for near-regular textures. In: Proceedings of the International Conference on Intelligent Sensing and Information Processing, 134–139, 2005.

[16] Lin, W. C.; Hays, J.; Wu, C.; Liu, Y.; Kwatra, V. Quantitative evaluation of near regular texture synthesis algorithms. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 427–434, 2006.

[17] Latif-Amet, A.; Ertüzün, A.; Erçil. A. An efficient method for texture defect detection: Sub-band domain co-occurrence matrices. *Image and Vision Computing* Vol. 18, Nos. 6–7, 543–553, 2000.

[18] Dai, D.; Riemenschneider, H.; Van Gool, L. The synthesizability of texture examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3027–3034, 2014.

[19] Bridson, R. Fast Poisson disk sampling in arbitrary dimensions. In: Proceedings of the SIGGRAPH Sketches, 22, 2007.

[20] Wei, L. Y. Parallel Poisson disk sampling. *ACM Transactions on Graphics* Vol. 27, No. 3, Article No. 20, 2008.

[21] Oliva A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* Vol. 42, No. 3, 145–175, 2001.

[22] Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* Vol. 2, No. 3, Article No. 27, 2011.

[23] Ojala, T.; Pietikinen, M.; Menp, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 24, No. 7, 971–987, 2000.

[24] Sanchez-Avila, C.; Sanchez-Reillo, R. Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation. *Pattern Recognition* Vol. 38, No. 2, 231–240, 2005.

Huisi Wu is currently an associate professor in the College of Computer Science and Software Engineering, Shenzhen University. He received his B.Sc. and M.Sc. degrees in computer science from Xi'an Jiaotong University in 2004 and 2007, respectively. He obtained his Ph.D. degree in computer science from the Chinese University of Hong Kong in 2011. His research interests are in computer graphics, image processing, and medical imaging.

Xiaomeng Lyu received her B.S degree in software from Fujian Normal University in 2016. Currently she is studying at Shenzhen University for her master degree. Her research interests include computer vision, texture analysis and pattern recognition.

Zhenkun Wen received his M.Sc degree in science and technology from Tsinghua University in 1999. Since 1987, he has been engaged in computing research and teaching in Shenzhen University. He is a professor of computing and software, and director of the Science and Technology Department of Shenzhen University. His research interests are in video tampering detection and location, video information security, and information management system design and implementation.