Perceptual Texture Similarity Learning Using Deep Neural Networks

Ying Gao*, Yanhai Gan*, Junyu Dong*, Lin Qi* and Huiyu Zhou†
*College of Information Science and Engineering, Ocean University of China, Qingdao, China
†School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, United Kingdom
Email: dongjunyu@ouc.edu.cn

***Abstract*—The majority of studies on texture analysis focus on classification and generation, and few works concern perceptual similarity between textures, which is one of the fundamental problems in the field of texture analysis. Previous methods for perceptual similarity learning were mainly assisted by psychophysical experiments and computational feature extraction. However, the calculated similarity matrix is always seriously biased from human observation. In this paper, we propose a novel method for similarity prediction, which is based on convolutional neural networks (CNNs) and stacked sparse auto-encoder (S-SAE). The experimental results show that the predicted similarity matrixes are more perceptually consistent with psychophysical experiments compared to other predicting methods.**

I. Introduction

Texture is one of the most important cues for human visual recognition. Textures are known to be useful information in many visual tasks, such as material recognition [1], scene understanding, and image segmentation [2]. Texture appearance varies extremely much, so it becomes hard to describe a texture using common language. Therefore, many early researches focused on finding the most suitable perceptual dimensions for texture describing [3] [4]. Texture similarity indicates how similar two textures look. Learning perceptual similarity accurately helps to study the similarities and differences between different images, and it can be well applied to image retrieval, object recognition and other fields.

Texture similarity data are generally collected by performing psychophysical experiments, such as free-grouping experiments and pairwise comparison [5] [6] [7]. Based on the results of psychophysical experiments, researchers constructed a perceptual similarity matrix [8] [9] for textures, which was used to quantify texture similarity. Meanwhile, the perceptual texture space (PTS) [6] [7] was constructed based on the perceptual similarity data obtained from psychophysical experiments. The perceptual texture space is a low-dimensional space, in which a single dimension corresponds to a specific perceived attribute, such as contrast, direction and regular [10]. Each point in the perceptual texture space represents the PTS feature of the corresponding texture. The perceptual texture space is supposed to be consistent with human perception, which can provide an accurate perceptual texture similarity.

In recent years, deep learning has achieved state-of-the-art results in many fields. Zagoruyko et al. proposed a general similarity function based on CNN [11] and decision networks to compare image patches directly from image data [12]. Han et al. proposed a unified approach for unifying feature and metric learning for patch-based matching [13]. These methods focus on predicting the similarity of image patches while our tasks differs from the small-scale image patches similarity learning. In contrast to previous work, we proposed a method based on convolutional neural networks to directly predict similarity between textures, with an auto-encoder based algorithm to output the similarity value from the computational features. We have performed extensive experiments on publically available texture datasets, including the perceptual texture database(PTD) [10], PerTex dataset [14] and a natural textures dataset DTD [15]. Experimental results show that our method is accurate and effective to estimate the similarity between different textures.

II. Methods

A. Learning Texture Similarity via Convolutional Neural Networks

We proposed two different models for texture similarity learning. In this section, we introduce the first model based on convolutional neural networks. Given a pair of texture images, we can obtain the perceptual similarity between the textures through our trained model. We designed a regression model based on convolutional neural networks for the purpose of similarity learning. This model has two input images, which are the textures whose perceptual similarity (a real value representing the degree of similarity) is going to be estimated. Accordingly, we use a two-channel input in the model; each channel represents a single texture image, and the two images will be fed into the network simultaneously. The proposed model includes convolutional layers, max pooling layers, full connection layers and nonlinear regression layer. The loss layer has been set to an Euclidean loss. The definition of the Euclidean loss is: $Loss = \frac{1}{2N}\sum_{i=1}^{N} \parallel y_n - \widehat{y}_n \parallel_2^2$, where N is the number of the texture image pair, y_n is the predicted similarity of the texture pair, while the $\widehat{y}_n$ is the ground truth, i.e. the real similarity obtained from the psychophysical experiments. The value of perceptual similarity ranges from 0 to 1. A value close to 1 means the two texture images are perceived very similar, whereas a value of 0 represents the two textures are not similar. The proposed network architecture is based on GoogleNet [16], which has 22 layers with three softmax loss

978-1-5386-2165-3/17/$31.00 ©2017 IEEE

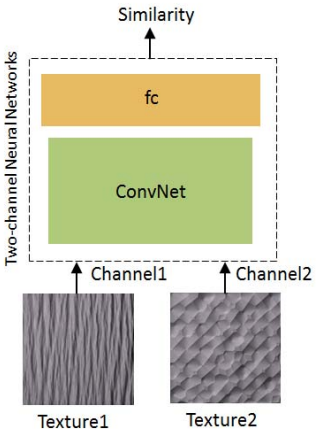

Fig. 1. Similarity learning model based on CNNs.

layers. We truncated the network layers before the second softmax loss layer as the foundation of our model. Then we made some modification to the model: the two softmax loss layers are replaced with Euclidean loss layers, and a sigmoid function is added as the activation function. The architecture of the model is shown in Fig. 1

B. Learning Texture Features and Similarity via Stacked Sparse Auto-Encoder

Many computational texture features have been proposed over the last forty years [17] [18] [19]. Computational features have been successfully used in many texture understanding tasks, such as texture classification [20] and material recognition [1]. Researchers also used the computational features to predict perceptual similarity [21]; however, the predicted similarity is not consistent with human observation. Previous work has shown that using computational features to train Random Forest can predict perceptual similarity more accurately [22] than other methods, but it requires a large amount of training samples and takes a long training time. In order to further improve the efficiency and accuracy of [22], we propose to use stacked sparse auto-encoder in addition to CNNs [23]. This architecture for perceptual similarity learning is shown in Fig. 2.

In this new architecture, five typical computational features were used: Local Binary Pattern (LBP) [24], Gabor [25], AlexNet [11], PCANet [26] and Bag-of-Words (BoW) [27]. For each texture dataset, the computational features of each texture are extracted. The dimension of Gabor feature and BoW feature is 48. As for AlexNet feature and PCANet feature, the network finally output a 4,096-dimensional feature and a 32,768-dimensional feature, the dimension of the two features are reduced to 48 with PCA reduction method. The dimension of LBP feature is 36 and it was rising to 48 in order to fit the network. Each pair of features represents the corresponding pair of images. The dimension of the paired feature is 96. The stacked sparse Auto-Encoder consists of five layers, including the input layer, output layer and three

Fig. 2. Similarity learning model based on stacked sparse auto-encoder.

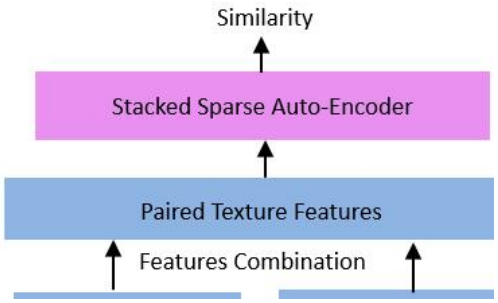

Fig. 3. Texture pairs in procedural texture dataset with their similarity values.

hidden layers. The five layers are composed of 96, 192, 96, 48, and 1 neuron respectively. The neurons of the first layer correspond to the combined computational features of two textures, and the neuron of the last layer corresponds to the similarity value of the texture pair. The whole architecture was trained by optimizing a cost function defined as the Mean Square Error (MSE) between the predicted similarity value and the real one (ground truth). MSE is defined as: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_n - \widehat{y}_n)^2$, where n is the number of the pairs, y_n is the predicted similarity value, and $\widehat{y}_n$ is the ground truth. In our experiments, we found that this method based on staked sparse auto-encoder produced the best results.

III. Experiment Results

The perceptual texture database [10] contains 450 textures generated by 23 different procedural texture generation models. Fig. 3 shows four pairs of textures in the dataset with their perceptual similarity values.

Firstly, we calculated the distance between computational features of the textures and compared the distance with the ground truth similarity obtained through psychophysical experiments. The L2 normalization and Euclidean distance were applied for the features extracted by Gabor, AlexNet

TABLE I
CORRELATION COEFFICIENT BETWEEN COMPUTATIONAL FEATURES AND REAL SIMILARITY VALUES.

Feature	Correlation Coefficient
BoW	-0.1069
LBP	-0.2207
Gabor	-0.4447
AlexNet	-0.6266
PCANet	-0.5782

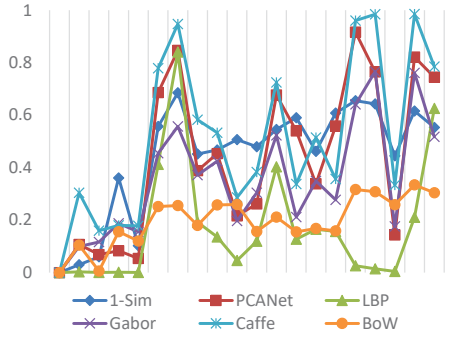

Fig. 4. Distribution of distance between computational features and similarity values.

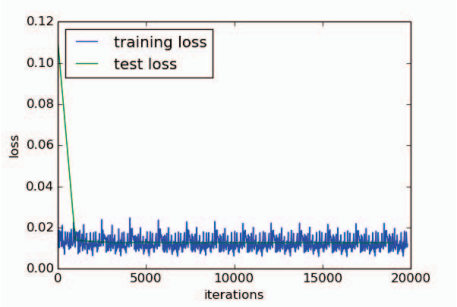

Fig. 5. The loss of procedural texture dataset.

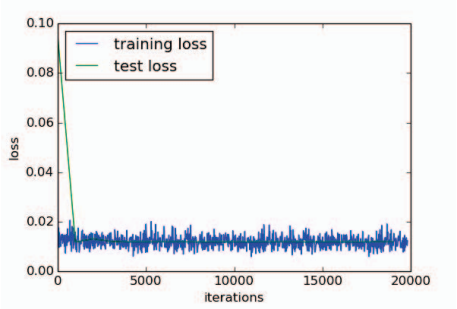

Fig. 6. The loss of Pertex dataset.

and PCANet, while L1 normalization and chi-square(χ^2) static [28] were applied for BoW and LBP. Since the distance measurement is opposite to perceptual similarity, that is, as the distance becomes smaller, the similarity gradually increases, the distance measurement result is negatively correlated with perceptual similarity. The correlation coefficient is shown in Table.1.

We randomly selected 20 pairs of textures, and compared the estimated similarities calculated by using five computational features with the perceptual similarity distances $(1 - sim)$, the distribution is shown in Fig. 4. We can see that the distribution of the similarities calculated by using Gabor, AlexNet and PCANet are relatively close to the ground truth. Whereas, the similarities calculated by using LBP and BoW are seriously biased from the true similarity values.

For convolutional neural network based perceptual similarity learning, we conducted experiments on both procedural texture dataset and Pertex. Pertex is natural texture dataset, which contains 334 real textures, such as woven wall coverings and building materials. We fed two images into the neural network as a two-channel input. The Euclidean loss curves of the two experiments are shown in Fig. 5 and Fig. 6 respectively. The Euclidean loss fluctuates in small magnitude during training. In the end, the test loss on procedural texture dataset and Pertex reach 0.0125 and 0.0116 respectively.

For the stacked sparse auto-encoder, we use 5-fold cross validation for hyper-parameters selection. There are 101475 pairs of images in the procedural texture dataset, and the pairs are randomly divided into 5 subsets. Each time, one subset is used as the validation set, while the other 4 subsets are gathered as the training set. The validation error is defined as: $error = \frac{1}{n}\sum_{i=1}^{n}|y_n - \widehat{y}_n|$. We made an average of the validation errors over 5 validation sets. Since the dimensionality of deep feature was too large, we applied Principal Component Analysis (PCA) to reduce the dimensionality of original features to 48. When AlexNet features are used, the smallest validation error 0.015 could be obtained with the experimental configuration: $sparsity(0.10.10.1)$, $lamda(1e-11)$, $beta(0.001)$. The smallest validation errors for different computational features are shown in Table.2. We randomly selected 100 pairs of textures, and compared their predicted similarity values with ground truth similarity values. The results are shown in Fig. 7.

To validate the availability of our model in other practical applications, we performed experiments on natural images and other procedural textures. We selected ten images from each of the 47 classes in the DTD dataset [15], and then we performed similarity prediction on the selected images. The experimental results are shown in Fig. 8. The size of this similarity matrix is 470×470; the samples are sorted by their categories. Different colors represent different similarity values. As can be seen from the predicted similarity matrix, textures from

TABLE II
TEST ERRORS OF DIFFERENT COMPUTATIONAL FEATURES IN PROCEDURAL TEXTURE DATASET.

Feature	MSE	Test Error
BoW	0.004	0.043
LBP	0.016	0.096
Gabor	0.005	0.051
AlexNet	**0.001**	**0.014**
PCANet	0.003	0.037

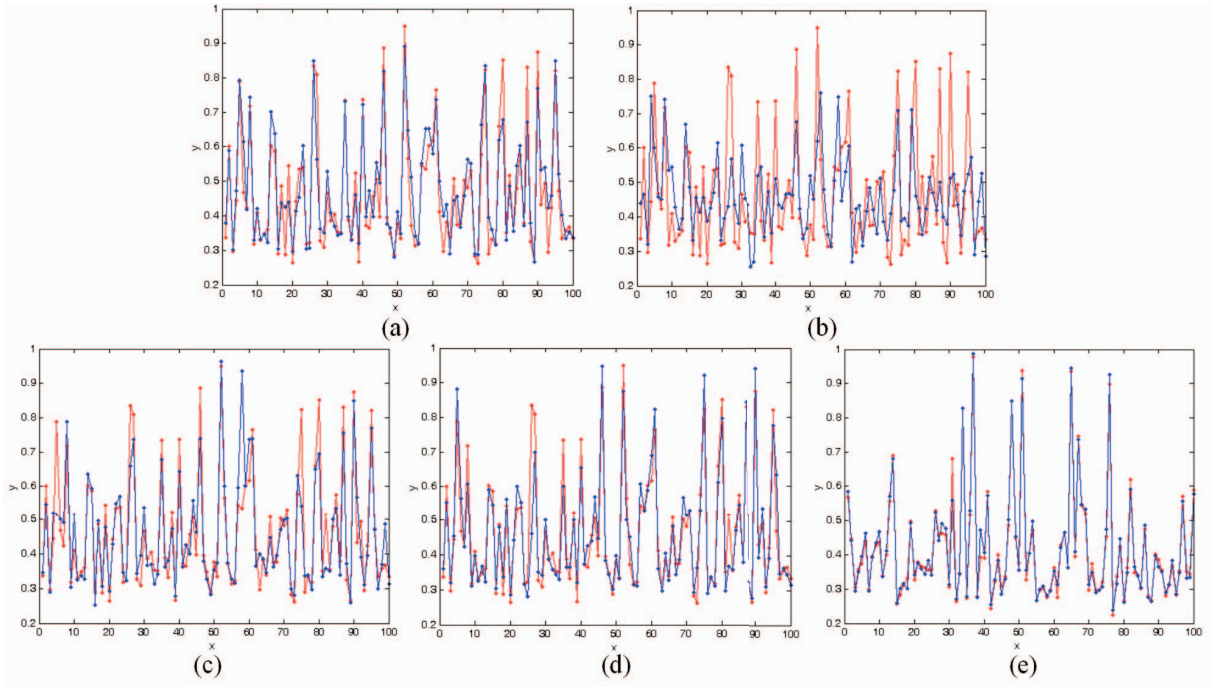

Fig. 7. The real similarity values and similarity values predicted by computational features, Fig(a),(b),(c),(d),(e) represent Gabor feature, LBP feature, BoW feature, PCANet feature and AlexNet feature respectively. The red line represents the real similarity values, and the blue line represents the predicted similarity values.

different categories are not so varied in appearance, which is also consistent with human observation.

In another experiment, we selected 4000 procedural textures generated by 10 procedural texture models. The procedural texture models [29] [30] can generate a large amount of texture samples through mathematical process. We choose 10 representative generation models, including Cellular Automaton (forest fire model), Cellular Automaton (surface tension model), Cellular Autonmaton (excitable media model), Cellular, Folding_texton, Folding_cellular, Folding_fractal, Folding_perlin, Fractal (one-over-fBeta-noise) and Fractal (Fourier spectral synthesis). The predicted perceptual similarity matrix is shown in Fig. 9. It can be seen that the colors of most sub-blocks are light, which means that the similarity between the texture samples generated by different models is very low. For example, the colors of sub-blocks coordinated by model 1 and model 4 are green, which means that the texture samples generated by these two models are not similar at all. However, some textures generated by different models own certain similarities, such as model 2 and model 3. The corresponding sub-blocks tends to be blue, meaning the texture samples are very similar to each other. The validity of the similarity matrix demonstrates that our similarity learning model can be generalized to other texture database.

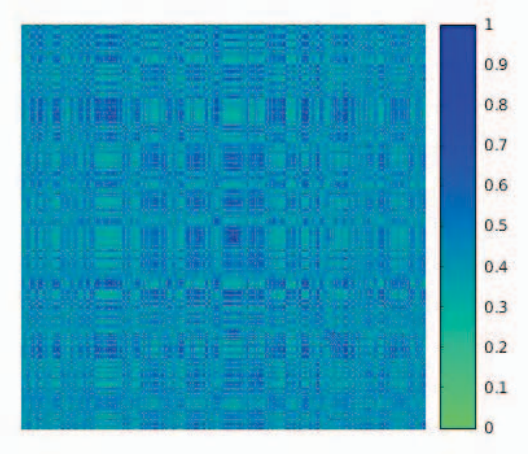

Fig. 8. The similarity matrix of 470 textures in DTD dataset.

IV. Conclusion

We presented a method based on deep neural networks for perceptual similarity prediction. The proposed method can directly estimate the similarity between two different texture images. We further introduce a method based on stacked sparse auto-encoder to combine texture features for texture similarity learning. We experimented with five different computational features, which were used to train the neural networks. Experimental results show that our method is more representative and accurate for perceptual similarity learning compared to other methods. Future work may investigate how to learn texture perceptual features and texture similarity simultaneously. More computational features and different architectures may also be verified.

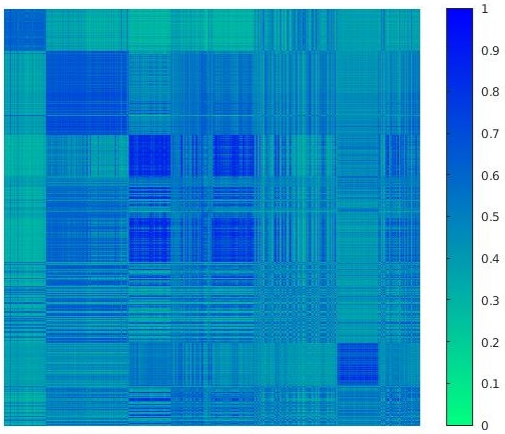

Fig. 9. The similarity matrix of 4000 procedural textures.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China(NSFC)(No.61271405).

REFERENCES

[1] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson, "Recognizing materials using perceptually inspired features," *International journal of computer vision*, vol. 103, no. 3, pp. 348–371, 2013.

[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[3] A. R. Rao and G. L. Lohse, "Identifying high level features of texture perception," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 3, pp. 218–233, 1993.

[4] B. Julesz, "Visual pattern discrimination," *IRE transactions on Information Theory*, vol. 8, no. 2, pp. 84–92, 1962.

[5] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[6] C. M. Heaps and S. Handel, "Similarity and features of natural textures," Master's thesis, University of Tennessee, Knoxville, 1996.

[7] A. R. Rao and G. L. Lohse, "Towards a texture naming system: identifying relevant dimensions of texture," in *Proceedings of the 4th conference on Visualization'93*. IEEE Computer Society, 1993, pp. 220–227.

[8] A. D. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler, "Perceptual similarity: A texture challenge." in *BMVC*, 2011, pp. 1–10.

[9] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[10] J. Liu, J. Dong, X. Cai, L. Qi, and M. Chantler, "Visual perception of procedural textures: Identifying perceptual dimensions and predicting generation models," *PloS one*, vol. 10, no. 6, p. e0130335, 2015.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.

[13] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.

[14] F. Halley, "Perceptually relevant browsing environments for large texture databases," Ph.D. dissertation, Citeseer, 2012.

[15] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[17] J. Bar-Ilan, K. Keenoy, E. Yaari, and M. Levene, "User rankings of search engine results," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 9, pp. 1254–1266, 2007.

[18] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 11, pp. 1624–1636, 2011.

[19] J. Filip and M. Haindl, "Bidirectional texture function modeling: A state of the art survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 1921–1940, 2009.

[20] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[21] X. D. M. E. B. Eng, "Perceptual texture similarity estimation," Ph.D. dissertation, Citeseer, 2014.

[22] J. Lou, L. Qi, J. Dong, H. Yu, and G. Zhong, "Learning perceptual texture similarity and relative attributes from computational features," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 2540–2546.

[23] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[24] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[25] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[26] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[27] J. Sivic, A. Zisserman *et al.*, "Video google: A text retrieval approach to object matching in videos." in *iccv*, vol. 2, no. 1470, 2003, pp. 1470–1477.

[28] H. William, "Press. numerical recipes 3rd edition: The art of scientific computing," 2007.

[29] A. Lasram, S. Lefebvre, and C. Damez, "Procedural texture preview," in *Computer Graphics Forum*, vol. 31, no. 2pt2. Wiley Online Library, 2012, pp. 413–420.

[30] A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker, "A survey of procedural noise functions," in *Computer Graphics Forum*, vol. 29, no. 8. Wiley Online Library, 2010, pp. 2579–2600.