CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Portsmouth University Research Portal (Pure)

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 147 (2019) 233–239
Procedia Computer Science
www.elsevier.com/locate/procedia

2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018

BTF data Generation based on Deep Learning

Xiaohua Zhang[a], Junyu Dong[a], Yanhai Gan[a], Hui Yu[b], Lin Qi[a,*]

[a] *Department of Information Science and Engineering, Ocean University of China, Qingdao, China*
[b] *School of Creative Technologies, University of Portsmouth, Portsmouth, UK*

Abstract

Many applications, such as computer-aided design and game rendering, need to reproduce realistic material appearance in complex light environment and different visual conditions. The authenticity of the three-dimensional object or the scene is heavily depended on the representation and rendering of textures, where the Bidirectional Texture Function (BTF) is one of the most widely-used texture models. In this paper, we proposed a neural network to learn the representation of the BTF data for predicting new texture images under novel conditions. The proposed method was tested on a public BTF dataset and was shown to produce satisfactory synthetic results.

© 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.

Keywords: Bidirectional Texture Function (BTF), image generation, Neural Networks, image compression, Adversarial Training

1. Introduction

Realistic visualization of materials and object surfaces is one of the major challenges in computer graphics. Traditionally, the micro-structure responsible for the reflectance behavior of material is simulated using relatively simple BRDF models [1]. However, the BRDF model deals well with homogeneous materials. In the real world, many surfaces are consisted of complex geometric structures, which introduce more complex optical characteristics such as shadows, mutual reflections, whereas simple BRDF models cannot represent. To solve this problem, Dana et al. [2] proposed the BTF model, which describes texture images in terms of positions, lighting and viewing directions. Therefore, it is a 6D function:

$$f(x, y, \theta_i, \varphi_i, \theta_v, \varphi_v) \tag{1}$$

where (x, y) denotes the pixel location, θ is the elevation angle and φ is the azimuth angle, the subscript i and v specify illumination and view respectively.

* Corresponding author. Tel.: +86-532-66781729.
E-mail address: qilin@ouc.edu.cn

1877-0509 © 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.
10.1016/j.procs.2019.01.241

The research on BTF focuses on data collection, compressed and modeling. Jiri Filip [3] proposed a BTF acquisition setup based on a simple, economical mechanical gantry. Boon University [4] proposed a method to divert industrial SVBRDF acquisition equipment to full BTF capture devices using sparse reconnaissance technology. In BTF data compression, Apparent BRDF (ABRDF) [5] decomposition or dedicated reflection model was used [6] [7] as well as various linear factorization techniques such as SVD, PCA [8] or tensor factorization [9]. BTF modeling includes sampling methods, spatial enlargement of BTF reflectance models, statistical models and hybrid methods. Guangzheng Fei [10] proposed a BTF generation algorithm based on the 3D surface mesh and BTF calculation on fabric surfaces using Ray tracing. Michael [11] proposed a method for creating comprehensive training samples for material classification. They use synthetic BTF data to overcome the time-consuming problem in manual acquisition and annotation. There is few relevant research on BTF data generation using deep neural networks.

The Convolutional Neural Network (CNN) has shown great superiority in the latest studies [12] [13]. Goodfellow et al. proposed a generative adversarial framework (GAN) [14] and showed excellent results in image generation tasks. To generate more realistic images, Wang factorized the image generation process and proposed a joint model consisting of style and structure generative adversarial networks [15]. It is a promising practice to exploit joint convolutional neural networks and adversarial training schemes for generating high-quality images. Deep Convolutional Generative Adversarial Networks (DCGAN) [16] is a better improvement of GAN. It dramatically improves the stability of GAN training and the quality of the results.

In this paper we proposed a deep convolutional generative adversarial network (DCGAN) to learn the appearance of BTF. Our model combined the existing Conditional Generation Adversarial Network (CGAN) [17] and the perception driven texture generation model (PDTG) [18]. It demonstrates the ability to learn BTF data under various illumination and viewing directions. The model can synthesize new BTF images given novel illumination and viewing directions.

2. BTF data generation

In this section, we first introduce the architecture of the proposed model for BTF data-driven generation. Then we provide details on the network design and initialization.

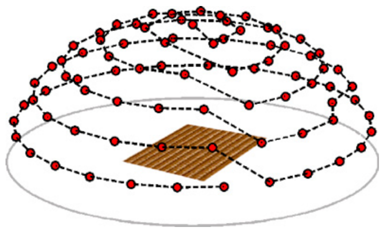


Fig. 1. An example of light and camera locations above the sample during the measurement UBO 2003 Dataset [19].

2.1. BTF Data Set

We used UBO 2003 BTF Dataset [19]. In this dataset, each sample was captured at pre-defined azimuth and elevation angles for illumination and viewing directions, as shown in Fig.1. An angular sampling of these BTF measurements covers a whole hemisphere uniformly above the sample. The 81 illumination/view directions result in 6561 images for one sample.

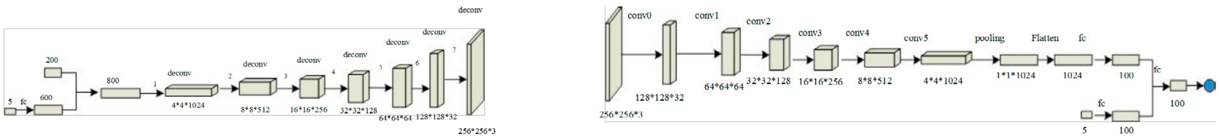
In our experiment, we selected three materials from UBO2003 Datasets [19]. Each image is labeled by a vector of five elements: viewing direction, illumination direction and material category. The direction is represented by the elevation angle and the azimuth angle. In detail, the direction categories are defined as:

- θ_i : 0°,15°,30°,45°,60°,75°
- θ_v : 0°,15°,30°,45°,60°,75°
- φ_i : Variety of direction between 0° and 360°
- φ_v : Variety of direction between 0° and 360°

- Material category (a): 10,20,30 (10 represents the material of pulli, 20 represents the material of corduroy, 30 represents the material of wool)

2.2. Structure of the Designed Network Model

We designed a deep joint model by using the light direction, observation direction and material type as conditional constraints. After the feature extraction and combination, we make predictions under the new illumination and viewing directions. SSIM [21] was used to evaluate results. As shown in Fig.2, the overall architecture includes two parts: the generative model and the discriminant model. The generative model is responsible for conditional BTF data texture generation,and the discriminant model is used to distinguish the generated BTF data texture from the training sample. In the discriminant model, direction features are mapped to a 100-dimensional space using the full connection layer. The two parts play an important role in judging whether a sample pair conforms to the true joint probability distribution. (see Fig.3.) We need the discriminator to figure out the union distribution of the condition and samples, and we use the conditional label information to constrain the generator which generate desired BTF data. We use G, D to represent the generative and discriminant model.



(a).generated model (b).discrimination model

Fig. 2. Network structure diagram of a model for the generative and the discriminant

Then the loss is constrained by D, the definition is:

$$d_{loss} = -\frac{1}{n}\sum_{i=1}^{n}(m_i \ln(D(x_i, y_i)) + (1 - m_i)\ln(1 - D(x_i, y_i))) \tag{2}$$

where x_i represents a training example, $m_i$is one or zero, y_i is the corresponding feature vector, and n is the number of training examples. The quadratic loss for h is defined as:

$$h_{loss} = \frac{1}{2n}\sum_{i=1}^{n}(f(y_i) - x_i)^2 \tag{3}$$

where $f(y_i)$ is estimated value and x_i is target value. The loss of G is constrained by D and H:

$$g_{loss} = m_{loss} + \alpha * p_{loss} \tag{4}$$

$$m_{loss} = -\frac{1}{n}\sum_{i=1}^{n}\ln(D(G(y_i, z_i),\ y_i)) \tag{5}$$

$$p_{loss} = \frac{1}{2n}\sum_{i=1}^{n}(f(y_i, z_i) - x_i)^2 \tag{6}$$

Where α is a tradeoff parameter, z_i is a random noise vector, G and D are trained in an adversarial scheme. H is another loss function, In this manner, the discriminator makes the generator produce realistic BTF data textures.

2.3. Network Design Details

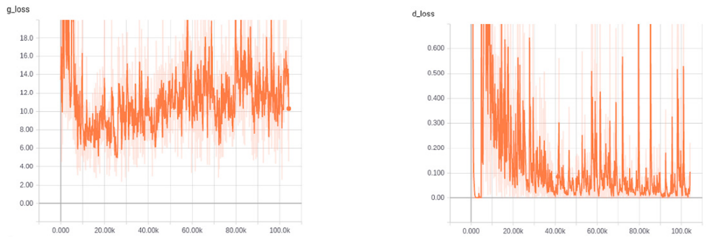
To emphasize the importance of the features of the BTF data under illumination and view direction for texture generation, we stretch the feature vector to 600 dimensions via a fully connected layer. The random noise vector is uniformly drawn from a 200-dimensional space, ranging from -1 to 1. The 200 dimensions random noise vector can be significantly varied to generate diverse textures given certain illumination and views information. If we change each aspect of the random noise vector with a step of 0.1, we can obtain different vectors. Let z represent the random variable. Then its variance is $Var[z] = 5^{200}$ The features vectors of obtained illumination and observation direction are scaled between -0.9 and 0.9. We use the following equation for scaling:

$$\overrightarrow{h} = min(max(h - E(h))/\sigma(h), -3, 3) * 0.3 \tag{7}$$

The purpose is to facilitate data processing and accelerate convergence. In this work, we set up a vector of 800 dimensions after data processing. Note that the variance of the random noise is three times larger than that of the stretched feature vectors. We tried to fuse kernels of different size to generate realistic textures with more details and global information, and found the $5 * 5$ kernel for convolution or transposed convolution produces better results.

2.4. Training

The architecture was trained by learning its weights using backpropagation with the training examples from the UBO2003 Datasets [19]. The dataset was randomly divided into a training set and a test set. We randomly select 80% of the data set as a training set. In our model, the generative model and the discriminant model are alternately optimized. After each optimization of the discriminant model, the generative model is optimized twice in succession. Then, we choose the ADAM [20] method of optimization. The tradeoff parameter was set as 10. Finally, we ran 100000 optimization iterations. The training process is illustrated in Fig.3



(a) is the generative model training loss curve (b) is the discriminant model training loss.

Fig. 3. Training loss curves of the designed model.

3. Experiments

3.1. Quantitative Evaluation

We decided to use the structural similarity (SSIM) [21] to evaluate results. Table.1 shows the result of evaluation method for the generated data. The SSIM is shown in Table.1. We compute the SSIM as suggested by Wang [21] while taking the mean value over all pixel values, as follows:

$$SSIM(x, y) = [l(x, y)]^{\alpha}.[c(x, y)]^{\beta}.[s(x, y)^{\gamma}] \tag{8}$$

While I is denoting the luminance, c is the contrast and s is a structural term. The SSIM covers values from -1 to 1, while a value of 1 describes two identical images. x and y are two image signals. In order to simplify the expression, we set $\alpha = \beta = \gamma = 1$ in this paper. People like to compare with the Variational Auto-encoder (VAE) [22] and GAN about image generation. We used CVAE to compare with our method. The results are shown in the following Table.1. The selection of training set in CVAE is consistent with our method. The values in the table indicate that the learning ability of different materials is different in the same model. Because the test set contains a lot of data and the values under each direction of the predicted material are different. We got lots of SSIM, and we can only take the average as a reference.

The Variational Auto-encoder (VAE) is an important generative model that was proposed by Diederik P. Kingma and Max Welling in 2013 [22] [23]. VAE is easier to train and has explicit distribution assumptions (Gaussian

Table 1. SSIM evaluation Table

	Pulli	Corduroy	Wool
CVAE	0.1052	0.2078	0.2645
Our method	0.3805	0.3104	0.4281

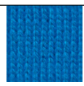
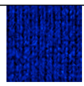
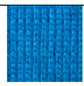
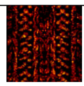
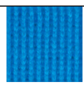
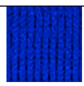
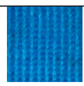
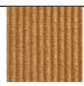
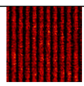
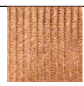
distributions) for explicit representation and observation, while our method is better able to capture the distribution of observations without any observation of the distribution. So we decided to make a comparative experiment on the method of adopting CVAE and our method. The predicting result and the value of the SSIM are shown in the Table.2. Images in the same row have the same direction feature. The experimental results show that our method has better performance in generating BTF data.

a.CVAE (SSIM=0.1888) b. our method(SSIM=0.6085) c. ground truth

Fig. 4. Zoom in on the generated images in Table 2 and compare their appearance textures.

We selected an example to enlarge in Table.2 (45 60 45 120 a=30), and its result is shown in Fig.4. We found that our method is better than the CVAE. The result of CVAE is fuzzy and the texture's appearance is different from ground truth. Its color is deeper due to the Gaussian distribution when encoding. But our method can generate clearer images and learn better the texture's appearance. Because our method can capture pixel correlations. We add H loss function to the generated model. It can solve the joint probability distribution between texture images and feature vectors, which is responsible for supervising the generation of texture image in accordance with the conditions of the generated model. It only focuses on whether the generated image is real, regardless of whether it corresponds. Therefore, our method is clearer and more vivid. But our method is also flawed, it only has a small range of changes in resolution. The texture generated during darker or brighter light is not ideal, which makes the SSIM value of the data is lower.

Table 2. Generating images based only the existing direction information and their value of SSIM.

Direction	GT	CVAE	Our	SSIM(CVAE)	SSIM(Our)
(a=30)45,60,45,120				0.1888	0.6085
(a=10)0,0,15,120				0.0538	0.3697
(a=30)15,240,45,260				0.2330	0.5732
(a=20)0,0,15,120				0.1782	0.3811

3.2. Qualitative Evaluation

We designed three experiments to evaluate the qualitative performance of our predicted results. Firstly, we fed directional feature vectors with different random noise to the generative model(Fig.6). It shows the appearance of the BTF data under specific illumination and viewing direction. Secondly, we manually edited some directional feature vectors and used them to generate textures(Fig.5). It should be emphasized that the new directional features were based on between 0 and 360 degrees, whereas the others were kept the same as the existing ones. We only provide six results due to the limited space. Finally, we manually edited randomly direction features to compare CVAE with our method(Table.3). It shows that the CVAE cannot learn well for the appearance of the texture generation. Our method can randomly generate a lot of data in new directions.

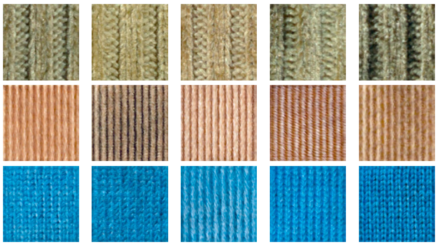


Fig. 5. BTF textures generated from new direction features. These images are generated from new perspectives without database. Images in the same column have the same direction feature. Every column feature vectors are (a 0 35 60 245), (a 15 45 120 60), (a 30 60 180 265), (a 45 70 30 120), (a 60 45 85 330). a is one of 10, 20 or 30

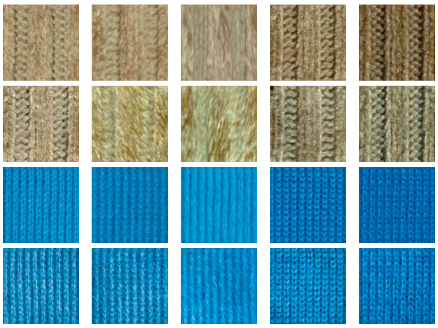
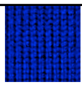
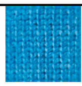
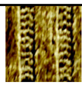
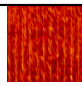
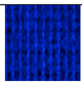
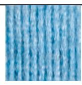
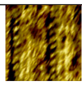
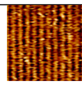
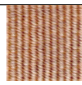
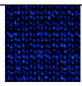
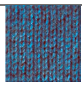
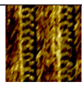
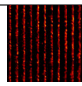


Fig. 6. BTF textures generated from existing feature vectors. Images in the same column have the same feature. The odd rows are ground truth, and the even rows are predicted images. The direction feature vectors are (a 30 150 45 100), (a 30 270 60 234), (a 30 330 75 90), (a 45 300 30 180), (a 60 18 15 60). a is one of 20 or 30.

Table 3. Image generation of new feature vectors that aren't within the dataset.

Direction	Wool (a=30)		Pulli (a=10)		Corduroy (a=20)	
	CVAE	Our	CVAE	Our	CVAE	Our
20,75,120,256,a						
65,120,240,324,a						
85,200,60,75,a						

4. Conclusion

We proposed a model to predict BTF data under novel conditions. We used the illumination direction, viewing direction and material type as conditional constraints to train the network. If we arbitrarily change a feature attribute, the other related features will not change. By comparative experiments, we found that the CVAE is worse than our method. Our method can generate clearer images and learn better about the appearance of the texture based on conditional constraints. But there are still problems in the network, such as how to improve the accuracy of predicting data. The CVAE can make the model to learn the distribution of materials with bright or dark light by Gaussian distribution. In the future, we will try to combine the two methods to improve the accuracy of the model.

5. Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) (No.61501417) and the Ph. D. Program Foundation of Ministry of Education of China (No.20120132110018)

References

[1] F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis, Geometrical considerations and nomenclature for reflectance, 1977, nBS Monograph 160, U.S. Department of Commerce, National Bureau of Standards

[2] Dana, K.J., Nayar, S.K., van Ginneken, B., Koenderink, J.J.: Reflectance and texture of real-world surfaces. In: CVPR, pp. 151-157. *IEEE Comput.* Soc., Los Alamitos(1997)

[3] Jiri Filip, Radomir Vvra, and Mikulas Krupicka; Rapid Material Appearance Acquisition Using Consumer Hardware;*Sensors* 2014,14(10), 19785-19805

[4] *In Proceedings of Proceedings of the 13th International Joint Conference on Computer Vision*, Imaging and Computer Graphics Theory and Applications, Volume 1: GRAPP, (2018)

[5] WONG T T,HENG P A,OR S H,et al. Image-based rendering with controllable illumination [C] *//Proceeding of the Eurographics Workshop on Rendering Techniques.* St. Etienne: Springer, 1997: 13-22

[6] McCool, M., Ang, J., Ahmad, A.: Homomorphic Factorization of BRDFs for high performance rendering. In: Fiume, E. (ed.) *ACM SIGGRAPH* 2001, pp. 185194. ACM, New York (2001).

[7] Ma, W.C., Chao, S.H., Chen, B.Y., Chang, C.F., Ouhyoung, M., Nishita, T.: An efficient representation of complex materials for real-time rendering. In: Lau, R.W.H., Baciu, G. (eds.)VRST, pp. 150153. ACM, New York 2004.

[8] KGuthe, M., Mller, G., Schneider, M., Klein, R.: BTF-CIELab: a perceptual difference measure for quality assessment and compression of BTFs. Comput. Graph. Forum 28(1), 101113 (2009)

[9] Vasilescu, M.A.O., Terzopoulos, D.: Tensortextures: Multilinear Image-Based Rendering. *ACM SIGGRAPH* 2004. ACM, Los Angeles (2004)

[10] Guangzheng Fei, Chu Qiu, MinYong Shi, Procedural Bi-directional Texture Function Synthesis of Woven Fabrics, *Journal of Computer Aided Design Computer Graphics*, Vol**17**, No 10, Oct 2005.

[11] Michael Weinmann, Juergen Gall, Reinhard Klein, Material Classification based on Training Data Synthesized Using a BTF Database, *Lecture Notes in Computer Science*, 8691 LNCS, PART **3**,156-171(2014)

[12] Kevin Jarrett, Koray Kavukcuoglu, Yann Lecun, et al., What is the best multi-stage architecture for object recognition?, in*2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 21462153.

[13] Alex Krizhevsky Ilya Sutskever and Geoffrey E Hinton Imagenet classification with deep convolutional neural networks Advances in *Neural Information Processing Systems*, vol. **25**, no. 2, pp. 2012, 2012.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, Generative adversarial nets, in *Advances in Neural Information Processing Systems*, (2014), pp. 26722680.

[15] Xiaolong Wang and Abhinav Gupta, Generative image modeling using style and structure adversarial networks, arXiv preprint arXiv:1603.05631, (2016)

[16] Radford, Alec, L.Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *Computer Science 2015*

[17] Mehdi Mirza and Simon Osindero,Conditional generativer adversarial nets, arXiv preprint arXiv:1411.1784, (2014)

[18] Yanhai Gan, Huifang Chi, Ying Gao, Jun Liu, Guoqiang Zhong, Junyu Dong. Perception Driven Texture Generation. *ICME*, 2017.

[19] Sattler, M., Sarlette, R., Klein, R.: Efficient and realistic visualization of cloth. In: *Eurographics Symposium on Rendering* 2003, pp. 167178 (2003)

[20] Diederik Kingma and Jimmy Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, vol. **13**, no. 4, April 2004, pp. 600612

[22] Kingma D P, Welling M. Auto-Encoding Variational Bayes[J]. stat, 2014, 1050: 10.

[23] DOERSCH C. Tutorial on Variational Autoencoders[J]. stat, 2016, 1050: 13.