Neurocomputing 000 (2018) 1–14

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Perception-driven procedural texture generation from examples

Jun Liu [a], Yanhai Gan [b], Junyu Dong [c,*], Lin Qi [c], Xin Sun [c], Muwei Jian [d], Lina Wang [e], Hui Yu [f]

[a] Science and Information College, Qingdao Agricultural University, 700 Changcheng Road, Qingdao, China
[b] Hisense TransTech Co., Ltd, China, No.17 Donghai West Road, Qingdao, China
[c] Department of Computer Science and Technology, Ocean University of China, 238 Songling Road, Qingdao, China
[d] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China
[e] China Unicom Institute of Software, N0.87 Huaneng Road, Jinan, China
[f] Department of School of Creative Technologies, University of Portsmouth, Eldon Building, Winston Churchill Avenue, Portsmouth PO1 2DJ, UK

ARTICLE INFO

Article history:
Received 11 September 2017
Revised 5 December 2017
Accepted 10 February 2018
Available online xxx

Communicated by Dr Chenguang Yang

Keywords:
Procedural texture
Texture generation
Texture perception
Convolutional neural network
Deep learning
PCA-based Convolutional Network

ABSTRACT

Procedural textures are widely used in computer games and animations for efficiently rendering natural scenes. They are generated using mathematical functions, and users need to tune the model parameters to produce desired texture. However, unless one has a good knowledge of these procedural models, it is difficult to predict which model can produce what types of textures. This paper proposes a framework for generating new procedural textures from examples. The new texture can have the same perceptual attributes as those of the input example or re-defined by the users. To achieve this goal, we first introduce a PCA-based Convolutional Network (PCN) to effectively learn texture features. These PCN features can be used to accurately predict the perceptual scales of the input example and a procedural model that can generate the input. Perceptual scales of the input can be redefined by users and further mapped to a point in the perceptual texture space, which has been established in advance by using a training dataset. Finally, we determine the parameters of the procedural generation model by performing perceptual similarity measurement in the perceptual texture space. Extensive experiments show that our method has produced promising results.

1. Introduction

Procedural textures have been widely used in computer games, animation and many other graphics applications for efficient rendering of natural elements, such as wood, marble, rocks, clouds and other materials [1]. They are typically created by procedural models, which are essentially mathematical functions and implemented using computer algorithms. The advantage of using procedural textures is that they require little storage and computation, and can be generated in real time. However, tuning the parameters of procedural models to produce desired textures is a difficult task even for experienced users. Unless one has a good knowledge of procedural texture models, it is difficult to predict which model can produce what types of textures. In addition, parameters of one model will produce overlapping effects on the output texture appearance. Therefore, it is hard to evaluate the influence of each parameter on the output texture.

For artists, designing a “new” texture that can be used in games or animations normally starts with an example of texture, e.g. a rock surface image downloaded from the internet or generated using procedural models. However, the example texture might not meet user expectations; changes in one or more of its perceptual properties are often required (e.g. the artist might wish the rock surface to look rougher). Fig. 1 illustrates the designing process. Currently, to our best knowledge, no software can provide direct solutions to this user requirement. Most packages only provide functions for manual editing, which is complicated and time-consuming. Even though some powerful texture generators (e.g. Genetica, FilterForge) can create high-quality textures or animated textures from given images, they are not able to make direct modifications to perceptual attributes, e.g. modifying roughness, directionality or regularity of the input texture. It is also the case for finding proper procedural textures, i.e. a “new” procedural texture whose perceptual characteristics are different from the example. This is indeed an even more difficult task, because generation of a procedural texture with different perceptual properties involves finding both proper models and parameter settings. With only an example procedural texture as input, current commercial software cannot determine its generation models and corresponding parameters.

In this paper, we propose a novel approach for generating a new texture with different perceptual properties yet sharing certain similarity to the example image. Fig. 2 shows the framework.

* Corresponding author.
E-mail addresses: dongjunyu@ouc.edu.cn, junyu.dong@163.com (J. Dong).

https://doi.org/10.1016/j.neucom.2018.02.061
0925-2312/

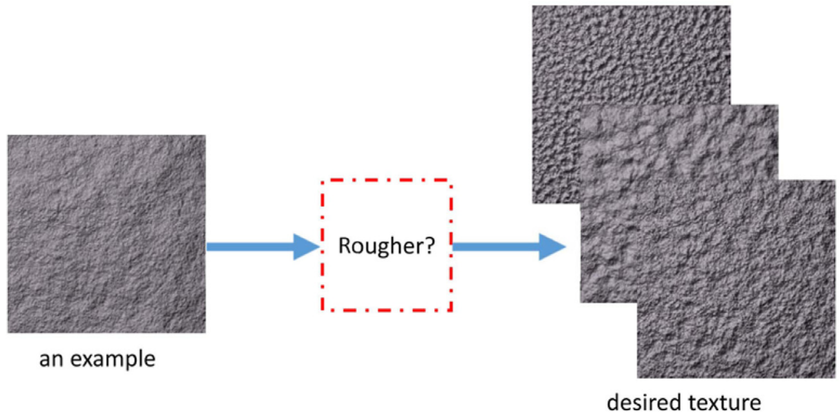

Fig. 1. The procedure of designing a “new” texture based on an example. The rock surface on the left is the example, and the surfaces on the right are the outputs that are a visually similar but look rougher than the example.

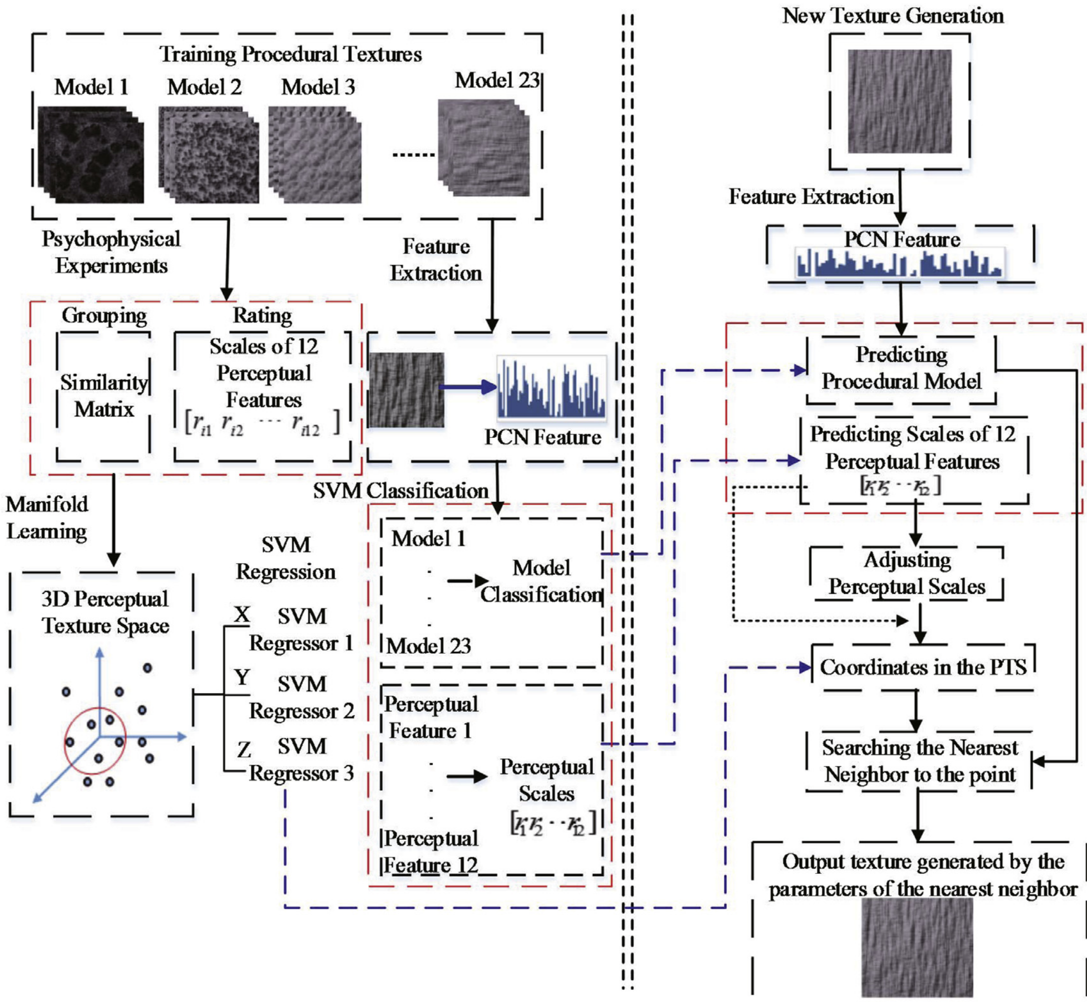

Fig. 2. The framework of the proposed approach. The left part shows the training process and the right part shows the process for generating a new texture.

The input to the system is an example texture, and the system can automatically find a procedural model and determine the parameters to output a new texture. In the proposed approach, users are allowed to adjust one or more perceptual properties of the input texture. Thus, the new texture bears resemblance to the example, while certain perceptual features can be perceived differently from the example. The left block of Fig. 2 shows the training process. First, we use a training dataset introduced in the previous work [2]; the procedural textures in this data set are generated by 23 procedural models and annotated with 9-point Likert scales for twelve perceptual features. We call these annotated values as perceptual scales in this paper, assessing to what extent the features are perceived by subjects. The similarity matrix derived from the grouping experiment is used to construct the perceptual texture space (PTS) [2], while the perceptual scales of the training samples from the rating experiment are used to train regression models.

Please cite this article as: J. Liu et al., Perception-driven procedural texture generation from examples, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.02.061