International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.12 (2015), pp.421-428
http://dx.doi.org/10.14257/ijsip.2015.8.12.39

An Improved SOM Based Surface Texture Synthesis

Jin Zhao and Chen Deyun[*]

Harbin University of Science and Technology, Harbin, China, 150080
chendeyun@hrbust.edu.cn

Abstract

This paper proposed a new method for surface texture synthesis using improved self-organizing maps as the synthesis logic. The method will first loop map the sample image to the target surface until the surface is full filled, then construct a improved self-organizing maps model, in which use sample image as the input level and the target three-dimensional surface as out put level, to adjust the pixel position in target surface, and the mapping area is controlled in the neighborhood of local extremum by reducing search area of variables. The new algorithm not only speed up the synthesis progress, and also enhances the image quality of target surface texture, and compare with the previous algorithm, it is not necessary for the user's intervention. Compared with previous results for the proof of our concepts, we have successfully implemented the experimental results and the proposed algorithm.

Keywords: *3D surface, texture synthesis, self-organizing maps, local extremum*

1. Introduction

In dealing with computer texture mapping, we often encounter such a problem, that is, the texture image source is often a very limited size. Therefore, One of the key issues in texture synthesis technology is how to generate a larger texture with a given smaller texture.

Textures can be two-dimensional, three-dimensional, or even higher-dimensional. Although a large number of studies have been performed successfully on texture synthesis, then most of the works concentrate on two-dimensional texture synthesis, The research on three-dimensional texture, or surface texture synthesis is very scarce. The main reason is that the complexity of the surface texture synthesis is high and it is difficult to solve the problem. Among the existing approaches for surface texture synthesis, how to balance the image quality and efficiency and how to guarantee the completeness of image is a very important topic but not well resolved. The pioneering work in this specific direction, done by Jagnow *et. al.,* (Wei Liyi, Levoy Marc, 2001) especially caught our eyes for its great outcome, But did not solve all the problems, there are two important problems unsolved now. The first question involves the three-dimensional shapes of the target synthesized particles. According to the paper, these shapes could be derived by applying stereology, otherwise you need to provide three-dimensional particles. The application of stereology may be difficult, The application of stereology may be difficult, however, as often times only one two-dimensional image is available for texture synthesis. In this paper, we propose a simple algorithm that could approximately construct the shapes of desired three-dimensional particles through the concept of visual hull use the feature of neural network, assuming that the synthesized three-dimensional particles are iso-tropic, *i.e.,* bearing similar cross sections from every viewing direction. The second question is about the layout of these three dimensional particles. According to the paper, this issue is solved by the improved self-organizing

[*]Corresponding Author

maps, where all the particles were initially put into the volume, then their locations or even orientations could be gradually adjusted to avoid collisions.

The rest of the paper shows the experimental results produced by our system, then concludes our work and hints several potential future research directions.

2. Related Work for Surface Texture Synthesis

At present, there are few publications on the three-dimensional surface texture synthesis. Zalesny and Van Gool (Zhang Jingdan, Zhou Kun, Velho Luiz, 2003) present a multi-view texture model which can synthesise new viewpoints. However, they do not consider different illumination, Which will introduction in this paper. Shum and his colleagues(Shikhmin Michael, 2001) used the CUReT database (Zelinka Steve, Garland Michael., 2006) to develop a method for the generation of bi-directional texture functions (BTFs). They applied a shape-from-shading algorithm to recover surface height and albedo maps of samples assuming Lambertian reflectance. These are used to synthesize a greater height map and image template. The final image is synthesized by the template image and the reference image. Leung and Malik (Kwatra V, Essa I, Schodl A, 2003) proposed the use of three-dimensional textons to synthesise new images under arbitrary viewpoints and illuminations. In later work, Shum *et. al.*, also exploited the idea of 'textons' and coupled this with a modified two-dimensional texture synthesis algorithm (Kohonen T., 1989). The complexity of using the three-dimensional group to calculate the multivariate vector is very high. So they use an effective method to transform the three-dimensional surface texture primitives, *i.e.*, the dot product using three-dimensional texture primitives.

3. Self-Organizing Maps

The SOM is one of the most popular neural networks used in unsupervised learning, which was first described by Kohonen (Kilthau S L, Drew M , Moller T , 2002) as a biologically inspired method to generate useful representations of data objects. An SOM network composed by neurons organized in a lattice. The neurons are connected to the adjacent neurons by a neighboring relationship, which constitutes the topological structure of the map. The network achieves a nonlinear projection from the high-dimensional input space to the low-dimensional lattice of neurons. SOM can be used as a high dimensional data clustering tool to construct a high dimensional space of a topological structure to map the lattice of neurons in such a way that the distance between the input vector and the distance between the input vector is maintained. The derivation of SOM is not complicated. Each neuron is associated with a reference vector. we specify a reference vector for each neuron. We compared the input vector, Looking for which is the most similar reference vector for each neuron. by this way, the reference vector is determined and moved to the input vector.

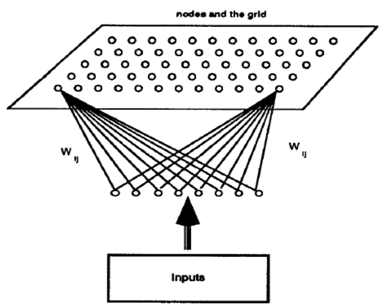

Figure 1. Architecture of Kohonen's Feature Map

Each node is connected to every input dimension. Let's generally describe the basic SOM algorithm:

(1) Each neuton i has a d-dimensional prototype vector $W^{(i)}_{(t)} = [W^{(i)}_{1(t)}, W^{(i)}_{2(t)}, \ldots, W^{(i)}_{N(t)}]$

(2) Give a input set:

$X(t) = \{X_1(t), X_2(t), \ldots, X_N(t)\}$

(3) At each training step, a sample data vector X(c) is randomly chosen from the training set. Distances between x and all the prototype vectors are computed. The best-matching unit(BMU), denoted here by c, is the map unit with prototype closest to x:

$$\left\| X(c) - W^{(s)}{}_{(c)} \right\| = \min_k \left\| X(c) - W^{(k)}{}_{(c)} \right\|$$

(4) Update the prototype vectors: the BMU and its topological neighbors are moved closer to the input vector in the input space, as shown in, The update rule for the prototype vector of unit i is:

$$w^{(k)}{}_i(t+1) = w^{(k)}{}_i(t) + \alpha(t) h_{ci}(t)[X_i(t) - w^{(k)}{}_i(t)]$$

Where t denotes time, $\alpha(t)$ is learning rat and $h_{bi}(t)$ is a neighborhood kernel centered on the winner unit. The kernel can be for example Gaussian:

$$h_{ci}(t) = e^{-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}}$$

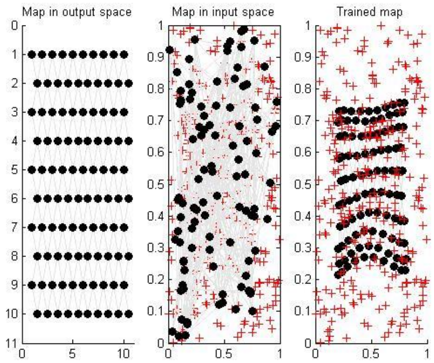

Figure 2. Process of Self-Organizing Maps

For the proposed method, there are the following specifications:

1. Initialization Step:

For the normal SOM algorithm, The output node in the map with a vector of weights, which is assigned at the beginning of the training of small random values. The purpose for that is to distinguish the mapped nodes with normal background nodes. However, for texture synthesis, to get high quality display effect, all the nodes on the surface have to be mapped and the color information should be kept. Our solution is to use the sample image directly draw on the surface until all the pixel on the surface is filled. In general, according to their global feature distribution, surface texture synthesis methods partition image pixels . In Jiang *et. al.*, 's work (Graepel, T., Burger, M., Obermayer, K., 1998), The function of SOM is cluster the image pixels, which based on the spatial features and the color of the image. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ denote the set of feature vectors corresponding to pixels in the sample image. Each item $\mathbf{x}_i$ is a nine-dimensional feature vector whose elements are the *x*-coordinate, *y*-coordinate, R,G and B values of the corresponding pixel and R,G,B values of the L range pixels. We will get the feature vector into the SOM network. After the completion of the training, We classify all the input vectors by topological mapping, so that the input space is partitioned into *k* classes. In other words, we give each image pixel to find the relevant class, the classification results are eliminated in the isolated pixels, and small pieces of processing, so as to get the results of segmentation.

2. How to make sure the average distribution mapping

One issue related to the algorithm above is, with one direction mapping at one time, the node of clustered pixel will gather to only one direction, the output level is what we want. To avoid such issue, directional feature is involved to SOM progress, we first separate the surface to 8 areas, for each epoch, direction is assigned and the random input node only map to the output node in the assigned direction range.

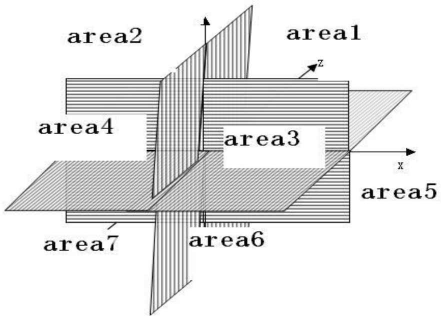

Figure 3. Areas for Output Level

3. Improved self organizing maps:

We use (Kilthau S L , Drew M , Moller T , 2002) proposed improved SOM method as the mapping module, the learning parameter α and the neighborhood radius σ are decreasing functions of time of the same kind and follow the same law.

$$\alpha(t) = \alpha_{\max}\left(\frac{\alpha_{\min}}{\alpha_{\max}}\right)^{\left(\frac{t}{t_{\max}}\right)}$$

$$\sigma(t) = \sigma_{\max}\left(\frac{\sigma_{\min}}{\sigma_{\max}}\right)^{\left(\frac{t}{t_{\max}}\right)}$$

The improved SOM based on the traditional SOM as the main algorithm and abstract the setting of learning parameter and neighborhood radius as a separately small optimality algorithm, simulated annealing is used for the optimization method, which is typically used for large scale problems, where a global minimum is hidden by many local minima. This heuristic maintains the quality of the improved SOM learning process, and maintains its unsupervised features. The adopted approach provides an adaptive learning rate factor α (*t,QE*), steered by the simulated annealing heuristic over the current resolution of the map. Using a SOM trained for epochs, at the end of each learning epoch, the Quantization Error (*QE*) can be identified with the parameter "temperature" T and the evolution of the network can be identified with a perturbation of the system. According to the algorithm, the parameters of the control temperature scheduling algorithm are adjusted automatically; an analogous criterion, the adaptive simulated annealing, was widely analyzed and developed in (Douzono, H., Hara, S., Noguchi, Y., 2000) and (Haese K., 1999). The *QE* of a SOM is defined as the euclidean distance between a data vector and its best matching unit according to:

QE = $QE = \left\| x - m_s(x) \right\|$

$\Delta QE = QE_{new} - QE_{old}$

And check by the condition of $\Delta QE < 0$ or $\exp(-\frac{\Delta QE}{T})$ <rand(0<rand<1), to determine stop the mapping or not.

The improved SOM based surface texture synthesis can be described as the following steps:

1. Initialize the pixel nodes on the surface as $SOM_{c,}$ looping to copy weights from the sample image until SOM_c is full filled, the SOM parameters:α_{MAX}, α_{MIN}, and the epoch counter $p = 1$, direction looping set as 1-8;

2. Start with first learning epoch: $SOM(p) = Training(SOM_{c,}\ direction)$

3. $QE_{MAX} = QE(p)$

4. $\Delta QE(p) = QE_{MAX}$

5. While $\Delta QE(p) \geq \delta$

(1) $p = p + 1$

(2) Run a new learning epoch:
$SOM(p) = T\ raining(SOMcurrent)$

(3) Calculate$QE(p)$

(4) Calculate$\Delta QE(p) = QE(p) - QE(p - 1)$

(5) GEt a random value $0 < rand < 1$

(6) if($e^{-\frac{\Delta QE}{QE}} < rand$ $\|\Delta QE(p-1) > \Delta QE(p)$)

i.$SOM_c = SOM(p)$

ii. $\alpha_{inc}(QE) = \Delta\alpha * \left| 1 - \frac{QE(p)}{QE_{\max}} \right|$

iii. $\alpha_{MAX} = \alpha_{MAX} + \alpha_{inc}(QE)$

iv. $\alpha_{MIN} = \alpha_{MIN} + \alpha_{inc}(QE)$

(7) Else

i. if ($e^{-\frac{\Delta QE}{QE}} > rand$)
Use the initial values of α_{MAX} and α_{MIN}

ii. if ($\Delta QE(p - 1) < \Delta QE(p)$)
$SOM_c = SOM(p - 1)$

6. End of learning after p epochs.

4. Experimant Results and Conclusion

In this section, we use three different square input images to synthesize two solid texture volumes respectively. The 2 sample images are classical texture to test the consistency feature and average distribution for synthesis. we demonstrate the results by our proposed algorithm. All the tests are performed on the computer with a Core-i7 3.2GHz and 8GBytes memory running on the Windows 7 operating system.

Figure 4. Average Distribution Testing

Figure 5. Consistency Testing

We can see that using the approved SOM algorithm, the quality of output surface is good, and achieved both average distribution and consistency.

5. Conclusion and Future Work

We propose a new algorithm for surface texture synthesis. In this algorithm, texture synthesis can be a unsupervised process and the quality of the output texture can be guaranteed. These results can prove the feasibility of our algorithm, but there are some limitations in our system, how to break through these limitations is our future research direction. First, our current assumption is the input sample image is structural picture, otherwise the synthesis may fail because for non-structural image it is loose clustered so the output image may not what we expected. Second, the direction used for pixel mapping may not enough, it should be better to dynamic set the direction based on the histogram information, will investigate more for the issue above.

Acknowledgment

This study is supported y National Nature Science Foundation of China under Grant No. 60173055.

References

[1] W. Liyi and L. Marc, Editors, “Fast texture synthesis using tree-structured vector quantization”, A. Computer Graphics Proceedings, Annual Conference Series ACM SIGGRAPH, New Orleans., **(2001)**, pp. 479-488.

[2] Z. Jingdan, Z. Kun and V. Luiz, “Synthesis of progressively variant textures on arbitrary surfaces”, J. ACM Transactions on Graphics, vol. 3, no. 22, **(2003)**.

[3] S. Michael, Editor, “Synthesizing natural textures”, A. Proceedings of the ACM Symposium on Interactive 3D Graphics, North Carolina, **(2001)**, pp. 217-226.

[4] Z. Steve and G. Michael, “Appearance-space texture synthesis”, J. Copyright restrictions prevent ACM from providing the full text for this work, vol. 2, no. 25, **(2006)**.
[5] V. Kwatra, I. Essa and A. Schodl, “Graphcut textures: Image and video synthesis using graph cuts”, J. ACM Transactions on Graphics, vol. 3, no. 22, **(2003)**.
[6] T. Kohonen, Editor, “Self-organization and associative memory”, J. 3rd ed. Berlin, **(1989)**.
[7] S. L. Kilthau, M. Drew and T. Moller, “Full search content independent block matching based on the fast Fourier transform”, A. Proceedings of IEEE International Conference on Image Processing, Rochester, **(2002**), pp. 669-672.
[8] T. Graepel, M. Burger and K. Obermayer, “Self-organizing maps: generalizations and new optimization techniques”, Neurocomputing 21, **(1998)**, pp. 173-190.
[9] H. Douzono, S. Hara and Y. Noguchi, “A Clustering Method of Chromosome Fluorescence Profiles Using Modified Self Organizing Map Controlled by Simulated Annealing”, IJCNN’00, IEEE-INNS-ENNS, **(2000)**.
[10] W. Xiao-feng, F. Li and S. Lin-quan, “Constraint Resolution Mechanism Based on Feature Modeling”, Journal of Harbin University of Science and Technology, vol. 9, **(2004**), pp. 31-34.