Depth Structure Preserving Scene Image Generation

Article · June 2017

CITATIONS
0

READS
159

5 authors, including:

Bingbing Ni
Shanghai Jiao Tong University
56 PUBLICATIONS **737** CITATIONS

SEE PROFILE

Yichao Yan
Shanghai Jiao Tong University
14 PUBLICATIONS **52** CITATIONS

SEE PROFILE

Depth Structure Preserving Scene Image Generation

Wendong Zhang, Bingbing Ni, Yichao Yan, Jingwei Xu, Xiaokang Yang
Shanghai Jiao Tong University
{diergent,nibingbing,yanyichao,xjwxjw,xkyang}@sjtu.edu.cn

Abstract

Key to automatically generate natural scene images is to properly arrange among various spatial elements, especially in the depth direction. To this end, we introduce a novel depth structure preserving scene image generation network (DSP-GAN), which favors a hierarchical and heterogeneous architecture, for the purpose of depth structure preserving scene generation. The main trunk of the proposed infrastructure is built on a Hawkes point process that models the spatial dependency between different depth layers. Within each layer generative adversarial sub-networks are trained collaboratively to generate realistic scene components, conditioned on the layer information produced by the point process. We experiment our model on a sub-set of SUNdataset with annotated scene images and demonstrate that our models are capable of generating depth-realistic natural scene image.

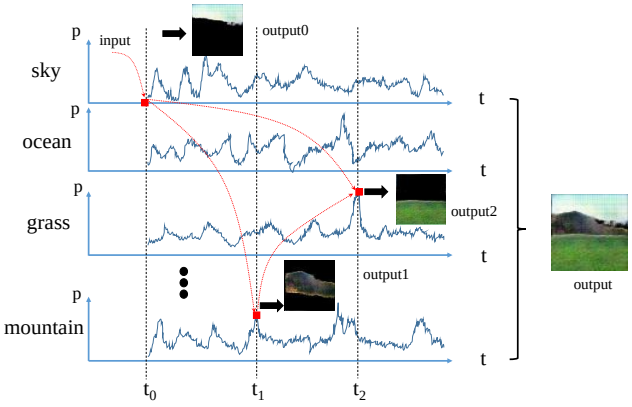

Figure 1. The Hawkes process in image generation in the depth direction. We set 'sky' as the label of first layer at depth t_0 and generate it as the background. Then, we randomly choose two more depth t_1, t_2 where the possibility of different class of layers can be calculated, and the largest one determines what kind of output layer will be generated at depth $t_i, i = 1, 2$. This process is executed sequentially along the depth and generate the entire image.

1. Introduction

Image generation has been a promising topic recently. Among the variant methods, generative adversarial network (GAN) shows enormous potential and becomes the most popular method for image generation. It has been applied to numerous domains such as image synthesis [17], image editing [3], image super-resolution [14], etc. In contrast to previous image generation tasks which mainly focus on numbers, faces or birds, this work is dedicated to generate natural scene images, which has broad application.

The main challenge of generation of scene images is how to well organize the spatial placement of various visual elements (e.g., mountain, river, road, sea, etc.) Spatial relationship between visual elements often refers to spatial proximity information in the image plane (x-y plane) as well as the ordering information in the depth channel. For example, sea and sky are often nearby to each other in the image, and a river often appears closer to the viewer than the mountain behind it. How to well encode spatial relationship between visual elements is the key to generate realistic natural scene images. On one hand, although two-dimensional spatial arrangements could be modeled by Markov random field based methods [30], there lack systematic solutions to well model the dependency information in the depth direction. It is mainly due to the fact that restriction in depth order is more explicit than that in the x-y image plane and should not be violated. Further, often high order dependency in the depth channel needs to be considered, rather than only pair-wise relationship. Namely, one layer of object is not only dependent on the depth layers around it, but also influenced by the complete depth structure of the image. For example, suppose the desert is in the farthest depth, some high trees are in the middle and a river is the nearest element, in this scene, if we only care about pair-wise relation between trees and river, it is alright. However, with the desert appears in the farthest depth, the high trees seems absurd even with a river. These facts make depth order preserving image generation more challenging. On the contrary, depth correctness is even more important for the authenticity of scene photo instead of the degree of the resolution.

Unfortunately, traditional GAN methods do not explic-

itly constrain that the generated image well follows the depth arrangement property for various visual elements. It is thus demanding to develop principled ways for depth order aware natural scene image generation. To this end, we are motivated by the success of point process in a wide range of applications such as market modeling [25], earth quake prediction [15] and social relations modeling [32]. In a point process, two kinds of influences, excitation and inhibition, are proposed to model the dependency between temporal events, which might not be temporally near-by, i.e., events that are not temporally nearby can also have effect on the current event (high-order dependency). Similarly, point process can also be utilized to model the high-order dependency in the depth direction. For example, if we already know that the sky appears as background and the beach is the nearest element to us, we can infer that the sea will be visually in the middle position with high confidence. We therefore propose a novel generation network called depth structure preserving scene image generation network (DSP-GAN).

The proposed network structure is hierarchical and heterogeneous, as illustrated in Figure 1. On top is an asynchronous layer generation module built on point process. In order to model depth dependency among visual elements, the input image is first decomposed into different layers located at different depth in scene to obtain asynchronous event-like visual element sequence. The asynchronous network is then applied to model the dependencies between layers and stochastic layer sequences are simulated for generating novel depth layers. At bottom is a layer-dependent hierarchical generative adversarial network to generate layer-dependent natural image patches/segments/part. In particular, we propose an enhanced version of composite generative adversarial network [12], which uses multiple generators and alpha blending to generate depth-specific image parts. We also propose an end-to-end training procedure for hierarchical GAN training. We have experiments on a subset of SUN2012 dataset [26] which contains natural scene images with complicated spatial structure, and our generated samples show high quality and natural results, along with depth-structure-realistic effects.

The rest of this paper is organized as follows. In section 2, some related works in image generation and Hawkes process are reviewed. In section 3, we explain the details of our work and it is further decomposed into three subsection focus on the application of Hawkes, architecture of our model and the value function we used in our training process. We give our generated images in section 4 and compare them with images generated by other methods and discuss the result in section 5.

2. Related Work

Image generation. Recent approaches proposed for generating realistic images could be mainly categorized into three kinds of models: variational autoencoder (VAE) [5], generative adversarial network (GAN) [6] and auto-regressive model [1].

Auto-regressive models use a product of conditional distributions to regress the joint distribution of the raw pixels in the image based on deep neural networks [24, 7, 18]. However, because of straightly extracting the iner-dependencies between pixels, structures in image are omitted and it is hard to generate more realistic images.

Variational autoencoder (VAE) is a famous approach to unsupervised model complicated distribution and have been applied in many generative models especially image generation. It contains a encoder used to approximate a posterior distribution and a decoder used to reconstruct data from latent variables [5]. Gregor et al. combine the recurrent neural networks with variational autoencoders and introduces attention mechanism to build a sequential generative model [8]. Yan et al. develop a layered generative model based on conditional variational autoencoders [28]. Although different kinds of complicated data could be modeled by VAEs, generating more realistic images is still very hard for it. Generative adversarial network (GAN) [6] is another popular approach for generative model. Many recent works are based on GAN. Some works focus on the architecture of original GAN for better performances [31, 4, 20, 2]. Conditional generative adversarial network [16] constrains the output by adding extra information to the input and many works follow this approach to solve more complicated tasks. Other applications such as image edition [3], image super-resolution [14], style transformation [23, 11] and unsupervised representation learning [21] also shows impressive results.

Hawkes process. Hawkes process [13] is a classical type of self-exciting point processes for continuous-time events modeling. What self-exciting means is that the occurrence of one event will increase the possibility of others for some period of time. Hawkes process has been applied in variant domains such as market modeling [25], earth quake prediction [15], crime modeling [22] and even trailer generation [27]. Additionally, multidimensional Hawkes process [9] is also proposed to tackle the problem of extracting hidden influence network. In computer vision, spatial point processes are introduced for object detection task [19].

In this paper, we take advantages of both Hawkes processes and generative adversarial networks to build a depth structure preserving scene image generation network. We focus more on how to model the spatial dependency between objects in images. Our works are build on [12] which uses recursive structure and generates images part-by-part. Another closely related work is proposed by [29]. It also

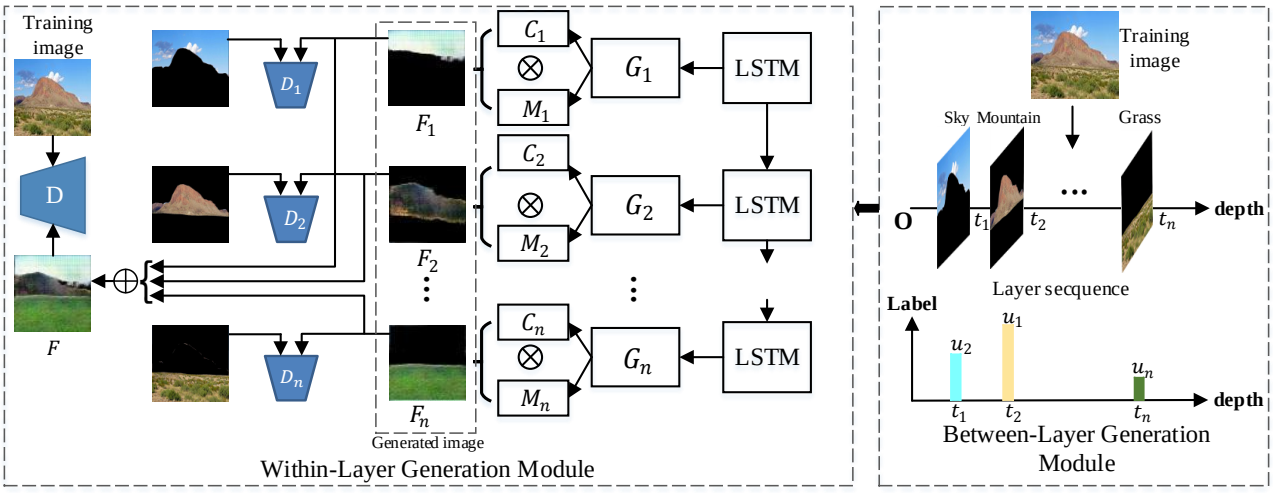

Figure 2. The architecture of the DSP-GAN.

employs a recursive structure but attempt to synthesise image composition by three attributes: appearance, shape, and pose. However, our model differs in following ways: 1) We focus more on how to model the spatial dependency between objects in images and we introduce Hawkes processes to tackle this problem. 2) Our separate sub-networks are trained for generating specific layers and the inputs are information of the structure instead of random noises. 3) We experiment our model with natural scene images which contain more complicated and strict spatial constrains. The results show that our model is capable of generating realistic images with more complicated spatial structure in depth channel.

3. Depth Structure Preserving Scene Image Generation

Most existing image generation methods only employ $2D$ information which can be directly obtained from training data. However, relationship among different layers in depth direction has not been well investigated, especially for those images with hierarchical spatial structures. Although some related works are proposed [28, 29] for generating foreground and background layers independently, these methods cannot be directly applied to more complicated images such as natural scene (i.e., which contain considerable information in depth direction).

To explicitly address this issue, we proposed a DSP-GAN to explicitly model the relationship of different image depth layers and perform layer-wise image generation, which is described in Figure 2. The framework proposed is heterogeneous and hierarchical, which include: 1) a between-layer generation module which is built on Hawkes process to model the influence among layers and generate realistic scene layer structure; and 2) a within-layer generation module which is conditioned on the depth layer structure and further generate layer-wise image contents. Note that although Markov random field based approaches [30] can also model between layer relationship, however, depth structure in scene image are very complex which are far beyond pair-wise (i.e., layers nearby) relationship. In contrast, Hawkes process based approach can well model higher order depth structure and more complex layer-by-layer interaction (i.e., even far away layers). Details of the proposed framework are explained as follows.

3.1. Between-Layer Generation Module

In the depth direction, a natural scene image can be decomposed into different classes of layers, such as sea, sky, mountain, etc. All layers are stacked one-by-one to form the entire image, and there exists strong structural prior for the spatial arrangement of different layers in image. For example, suppose that we observe sea in the image, more likely we will find an island or some rocks in it instead of forest or grass, and usually a beach would appear nearby. In other words, there are mutual influence between layers. For example, layers appear at farther depth will increase the probability of specific layers at nearer depth. In this work, we aim to explicitly model relationship among various layers, i.e., some of the interacting layers are situated far from each other in the depth direction.

Our proposed method is inspired by the Hawkes process [13]. The Hawkes process is originally proposed to model the relationship between continuous-time events. In a Hawkes process, each occurrence of event will excite the

process in the way that the probability of a subsequent occurrence is increased for a period of time after the first one. We adapt the original Hawkes process to our problem based on the underlying similarity between an event sequence and a depth layer sequence. Here depth layer sequence means a sequence of ordered image layers arranged according to their depth values, along with each layer's label (e.g., sky, sea, grass). Formally, we denote $t_i \in [0, 1]$ as the depth of i-th layer in sequence, where $t_i = 0$ means the farthest location and $t_i = 1$ means the nearest w.r.t. the image plane, and $u_i \in 1, 2, \ldots, U$ as the predefined label of the same layer (e.g., 1 represents 'rock'), where U is the total number of layer class. Then for each image, a depth-ordered layer sequence can be represented as $\{(t_i, u_i)_{i=1}^n\}$, where n is the total number of layers in the image.

A multi-dimensional Hawkes process is applied to explicitly model the influence between different layers. Specifically, we employ a U-dimensional point process N_t^u, $u = 1, \ldots, U$, and the conditional intensity function for the u-th dimension can be formulated as:

$$\lambda_u(t) = \mu_u + \sum_{i:t_i<t} a_{uu_i} g(t - t_i), \tag{1}$$

where μ_u is called the "base intensity" for the u-th Hawkes process, and $g(t)$ is the decay kernel for modeling the impact trends. While we use exponential kernel $g(t) = \alpha e^{-\alpha t}$ in this work, other positive kernels can also be embedded in our framework. Coefficient a_{uu_i} shows the mutually-exciting property between the u-th and u_i-th dimension. Suppose $u = 1$ represents the label of "rock" and $u_i = 3$ represents the label of "sea", the value of a_{uu_i} reflects the impact intensity to the layer of "rock" with a layer of "sea" appears before it. A large value of a_{uu_i} means that layers of class u_i are more likely to excite the appearance of layers u at the subsequent depth. In this way, spatial arrangement in the depth direction is well modeled.

With the samples c and conditional intensity functions λ_u, the log-likelihood function of model parameters $\Theta = \{\mathbf{A}, \mu\}$ can be proposed as follows:

$$\mathcal{L}(\mathbf{A}, \mu) = \sum_c (\sum_{i=1}^{n_c} \log \lambda_{u_i^c}(t_i^c) - \sum_{u=1}^{U} \int_0^1 \lambda_u(t) dt), \tag{2}$$

where $\mathbf{A} = (a_{uu_i})$ is the collection of mutually-exciting coefficients called "infectivity matrix", and μ is the collection of base intensities. Both $\mathbf{A}$ and μ can be estimated by maximizing the log-likelihood function above using algorithm ADM4 [32].

For generating new images, layer sequences are sampled from the learned Hawkes process. Given the information of the first layer $\{t_1, u_1\}$, at any depth $t_i, t_i > t_1$, the probability of any class of layer can be calculated according to the intensity functions λ_u. The one with the largest probability value will be chosen as the layer class appearing at depth t_i and the result will influence the samples of layers behind it. This processing can be continued until $t_i = 1$.

3.2. Within-Layer Generation Module

Given the layer sequence sampled from the learned Hawkes process, generating images with proper spatial distribution is still challenging. One straightforward way is to generate each layer separately and combine them together to obtain the entire image. However, the influence between different layers is missed if the generator has no access to previous layers. To solve this problem, a recurrent neural network, e.g., LSTM [10] is proposed to model the underlying dependency in layer sequence.

For a depth layer sequence $\{(t_i, u_i)_{i=1}^n\}$, at layer l, we transform the information of the layer (t_l, u_l) into a vector $\mathbf{x}_l$ by setting the depth t_l in the u_l-th position. Suppose $t_l = 0.5$ and $u_l = 4$, then we set the forth value in vector $\mathbf{x}_l$ as 0.5 and other elements equal to zero. The vector $\mathbf{x}_l$ is further used as the input to a LSTM structure which is updated as follows:

$$\mathbf{i}_l = \sigma(\mathbf{W}_i \mathbf{x}_l + \mathbf{U}_i \mathbf{h}_{l-1} + \mathbf{V}_i \mathbf{c}_{l-1} + \mathbf{b}_i), \tag{3}$$
$$\mathbf{f}_l = \sigma(\mathbf{W}_f \mathbf{x}_l + \mathbf{U}_f \mathbf{h}_{l-1} + \mathbf{V}_f \mathbf{c}_{l-1} + \mathbf{b}_f), \tag{4}$$
$$\mathbf{c}_l = \mathbf{f}_l \cdot \mathbf{c}_{l-1} + \mathbf{i}_l \cdot \tanh(\mathbf{W}_c \mathbf{x}_l + \mathbf{U}_c \mathbf{h}_{l-1} + \mathbf{b}_c), \tag{5}$$
$$\mathbf{o}_l = \sigma(\mathbf{W}_o \mathbf{x}_l + \mathbf{U}_o \mathbf{h}_{l-1} + \mathbf{V}_o \mathbf{c}_l + \mathbf{b}_o), \tag{6}$$
$$\mathbf{h}_l = \mathbf{o}_l \cdot \tanh(\mathbf{c}_l), \tag{7}$$

where σ is the sigmoid function, $\cdot$ denotes the element-wise multiplication operator. W_*, T_* and V_* are the weight matrices, and $\mathbf{b}_*$ are the bias vectors. $\mathbf{i}_l$, $\mathbf{f}_l$, $\mathbf{o}_l$, $\mathbf{g}_l$, $\mathbf{c}_l \in \mathbb{R}^N$ are input gate, forget gate, output gate, input modulation gate and memory cell. The output of hidden unit $\mathbf{h}_l$ at each time step will be used as input for each layer-wise image generator. LSTM structure ensures that the input for layers which will be generated later contains all the information of layers generated before. In our model, it means that the generator is aware of the class and depth of layers which have been generated before it (in the depth direction). In this way, the correlations between layers are well mapped from layer sequence to generation process.

In our model, generators produce both image layer and its corresponding mask layer (for final layer fusion) at the same time. The image layer shows the appearance according to the specific label and the mask layer with the value between 0.0 to 1.0 controls the transparency of image layer at the same timestep on pixel level. These two layers form an intermediate image which is not only a component of the entire image but also used for training the layer-dependent generator which will be explained later. Given the generator $G_1, G_2, \ldots, G_n$, image and mask $C_1, C_2, \ldots, C_n$, $M_1, M_2, \ldots, M_n$, the generation process of intermediate

images $F_1, F_2, \ldots, F_n$ can be described as follows:

$$C_i, M_i = G_i(\mathbf{h}_i), \tag{8}$$

$$F_i = C_i \cdot M_i, \tag{9}$$

$$F = \sum_{i=1}^{n} F_i, \tag{10}$$

where F denotes the final generated image. These elements are also shown in Figure 2.

3.3. Value Function

Based on the work of [12], extra discriminators $D_1, D_2, \ldots, D_n$ are added to each independent generator to form sub-GANs in our model, and the generators are also connected to the final discriminator D, which completes the whole architecture of our generative network.

In our model, each sub-GAN with generator G_i and discriminator D_i is trained with the value function as follows:

$$\begin{aligned} &\min_{G_i} \max_{D_i} (\mathbb{E}_{\mathbf{x}_s \sim p_{data}(\mathbf{x}_s)}[\log D_i(\mathbf{x}_s)] \\ &+\mathbb{E}_{\mathbf{h}_i \sim LSTM_i}[\log(1 - D_i(G_i(\mathbf{h}_i)))]), \end{aligned} \tag{11}$$

where $\mathbf{h}_i$ represents the hidden unit of the LSTM at timestep i, and $\mathbf{x}_s$ is the object pieces segmented according to the labels from the whole image data $\mathbf{x}$. This value function means that each sub-GAN is required to generate meaningful layer according to the label and depth we provide. The whole hierarchical GAN network including the LSTM structure is trained end-to-end.

Moreover, for the whole generative adversarial network with generators $G_1, G_2, \ldots, G_n$, another value function is proposed:

$$\begin{aligned} &\min_{G_1, G_2, \ldots, G_n} \max_{D} (\mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})]) \\ &+\mathbb{E}_{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n \sim LSTM}[\log(1 - D(\sum_i G_i(\mathbf{h}_i)))]), \end{aligned} \tag{12}$$

where $\sum_i G_i(\mathbf{h}_i)$ represents the entire image F. This value function means that we want to modify the final image for better performance after the generation of specific layers. These two value functions are trained alternately in our experiment and the algorithm of our model is illustrated in Algorithm 1.

Additionally, two different loss functions for mask layer generation are also added to the training process. In training of the whole generative network, the loss function is as follows :

$$\mathcal{L}_{M_i} = |p - \sum M_i| + \sum -(M_i - 0.5)^2, \tag{13}$$

where p is a predefined bound for the sum of pixel value in mask layer. While in training of each sub-GAN, we only constrain the value of each pixel in mask layer to be near zero or one:

$$\mathcal{L}_{M_i'} = \sum (M_i - 0.5)^2. \tag{14}$$

Both loss functions are added to the generator loss.

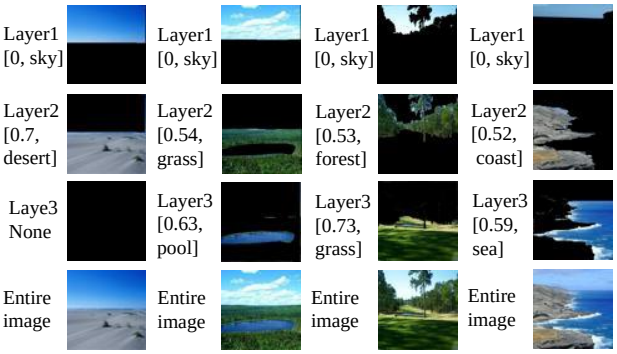

Figure 3. Samples of training data. The values in brackets represent the depth. The first three rows are different layers and the forth row is the entire image.

Algorithm 1 The algorithm of with-in layer generation module

Input: training data $\mathbf{x}$, layer sequences t, u, the number of generators n, mini-batch size m, the number of steps for training sub-GANs k

Initialize the parameters of all discriminator and generators, as well as the LSTM network

for number of training iterations **do**

  for k steps **do**

    Sample m data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \subset \mathbf{x}$ randomly

    Sample m layers sequences $\{\{t_i, u_i\}_1, \{t_i, u_i\}_2, \ldots, \{t_i, u_i\}_m\}$ according to $\mathbf{x}$ and input to LSTM obtain hidden vectors $\{\{\mathbf{h}_i\}_1, \{\mathbf{h}_i\}_2, \ldots, \{\mathbf{h}_i\}_m\}$

    for $i = 1$ to n **do**

      Update the generator G_i and discriminator D_i according to the value function in Equation 11.

    end for

  end for

  Update the generator $G_1, G_2, \ldots, G_m$ and discriminator D according to the value function in Equation 12.

end for

3.4. Training and Testing Details

In training phase, the Hawkes process is learned by optimizing the log-likelihood function Equation 2 with the ADM4 algorithm proposed by [32] based on a multi-dimensional Hawkes process. For the generative network, the input of our model is the layer sequence which contains the depth and labels of specific image. The sequence is further transformed into n 128-dimensional vectors where

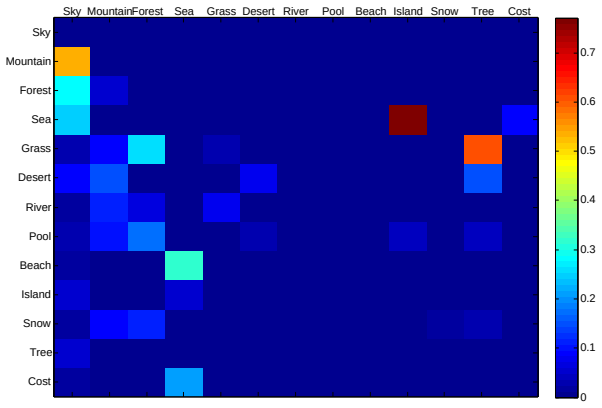

Figure 4. Intensity matrix $\mathbf{A} = (a_{uu_i})$ estimated from the training data.

each vector contains the information of the specific layer and n is the number of layers in the image. The learning rate for training the whole hierarchical GAN network is 0.0002 with RMSPROP optimizer, while for sub-GANs the learning rate is 0.0001. The number of steps for training sub-GANs is 8 and the iteration number of training process is about 8000.

In testing phase, new images are generated based on the layer sequences sampled form the Hawkes process. For convenience, we first use the training data to learn the Hawkes process, and based on the intensity functions we sample from Hawkes process and obtain some layer sequences for testing. The sampling process is organized as follows: for the fist layer, we always set the label u_1 as 'sky' which is a common background in natural scene and the depth t_1=0, while during the generation a small value is added to t_1 to avoid zero input. Then, to thoroughly test the capability of our model, two more depth values $t_2, t_3 \in (0, 1], t_2 < t_3$ are randomly selected for the generation of new layers. After that, according to the intensity functions we have obtained, we calculate the probability of all classes of layers at depth t_2 and the class with the largest probability is chosen for generation, the same process is then performed at depth t_3. Although we only generate 3 layers in our experiments, our method is easy to extend to arbitrary number of layers.

4. Experiments

In this section, we present extensive experimental evaluations and in-depth analysis of the proposed method. We also qualitatively compare our method with some related generation methods.

As there exists no dataset which contains natural scene images with depth information, we manually segment the fully annotated images chosen from SUN2012 dataset [26] into different layers according to the labels provided with

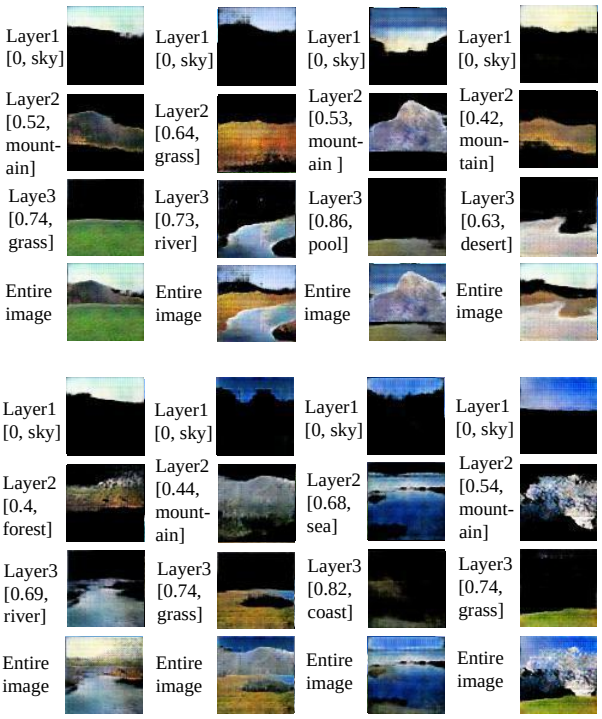

Figure 5. Results of our model. The values in brackets represent the depth. The first three rows are different layers and the forth row is the entire generated image.

the images. About 800 annotated natural scene images are used as training data with totally 13 different labels. We segment all images into two or three layers and resize the images and layers into 64×64 pixels for training our model. Some training data are shown in Figure 3.

4.1. Intensity Matrix in Hawkes Process

Given the layers sequences, we optimize the log-likelihood function in Equation 2 and obtain the intensity matrix $\mathbf{A} = (a_{uu_i})$ which represents the mutual influence between layers with label u and u_i. The matrix is shown in Figure 4.

We can intuitively observe the intensity of the influence between different classes of layers according to the color code. We have two observations. On one hand, the intensity matrix is asymmetrical, which shows obvious constrains on the order that different class of elements appears along depth direction in natural scene. On the other hand, the results shown in the matrix are consistent with our cognitive, such as island usually appear with sea or pool while forest can be observed in variant scene. Therefore, the intersection areas of these classes are highlighted.

These results demonstrate that depth spatial distribution in natural scene is an important factor in image generation and can be well modeled by Hawkes process.

Figure 6. The nearest training data. Images in the left column are generated images by our model.

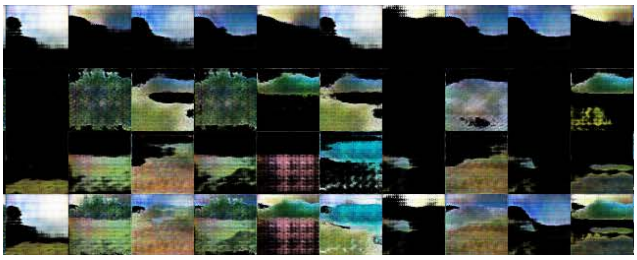

Figure 7. Results of our model without LSTM.

4.2. Generation Results

Some of the generated samples of our model are shown in Figure 5. We have three observations: 1) the specific object is clearly generated in each layer with sharp appearance boundary. 2) the relationship between different layers is successfully maintained during the generation. For example, the mountain layer always appears before the grass and the river always shows on the plain. 3) the entire image remains realistic even with the sharp appearance boundary of each layer. We also show the nearest examples in training data for some of our generated results in Figure 6, which further demonstrates that our generative network is not simply copy of the training data.

The generated layers also show high diversity which is influenced by not only the depth and label, but also the spatial distribution of layers generated before. In other words, even with the similar depth and same label, the generated layer will show huge difference if the layers before it are changed. We also show the nearest examples in training data for our generated results which further demonstrates the effectiveness of our method.

4.3. Component Analysis

We further provide component analysis on our model. We remove LSTM from our model and change the input for generating a specific layer from input vector $\mathbf{x}$ to a noise vector $\mathbf{z}$ sampled from a normal distribution where the mean value and standard deviation are replaced by the label μ and the product of the label μ and depth t. The results are shown in Figure 7. We can obviously see that no correlation exists between different layers and the model falls to generate realistic images, which demonstrates that hierarchical generation plays an important part in maintaining the relationship between different layers.

4.4. Comparison with Other Methods

We compare our results with deep convolutional generative adversarial network (DCGAN) [21] which is a fundamental architecture of GAN and widely used as a basic model in many works. We also show the results of composite generative adversarial network (CGAN) [12] on which we build our work. For both DCGAN and CGAN, the input is the noise $\mathbf{z}$ sampled from a particular noise distribution without any extra information, while we further test the CGAN using the input in our model to evaluate the improvements of our generative network. The results are shown in Figure 8.

From Figure 8, we observe that the samples generated by DCGAN suffer from severe blur effect. The quality of these results is lower to us not only in the whole image level, but also in the layer level. This is because previous methods try to generate the whole image at once and leave out the relationship between different layers. Therefore, it is very hard for DCGAN to generate natural scene image which contains complicated spatial distribution. While for the results of CGAN with noise input shown in 8, we can see that the quality of the generated samples is obvious lower even compared with the results of DCGAN. Some image layers, such as the second and the third layers, have little or no contribution to the entire images. Although this architecture is similar to the model proposed by us which also combines multiple generators to make the entire image part-by-part, no extra information is provided to guide each generator to focus on distinct components. Therefore, CGAN can not be directly applied to images with complicated hierarchical structure. While given the same input of depth and labels, our DSP-GAN successfully produce natural scene images with high quality, as can seen in Figure 8.

5. Conclusions

In this paper, we demonstrated that the depth structure preserving scene image generative framework we proposed, which is motivated by Hawkes process and hierarchical and heterogeneous generative networks, succeeds in generating

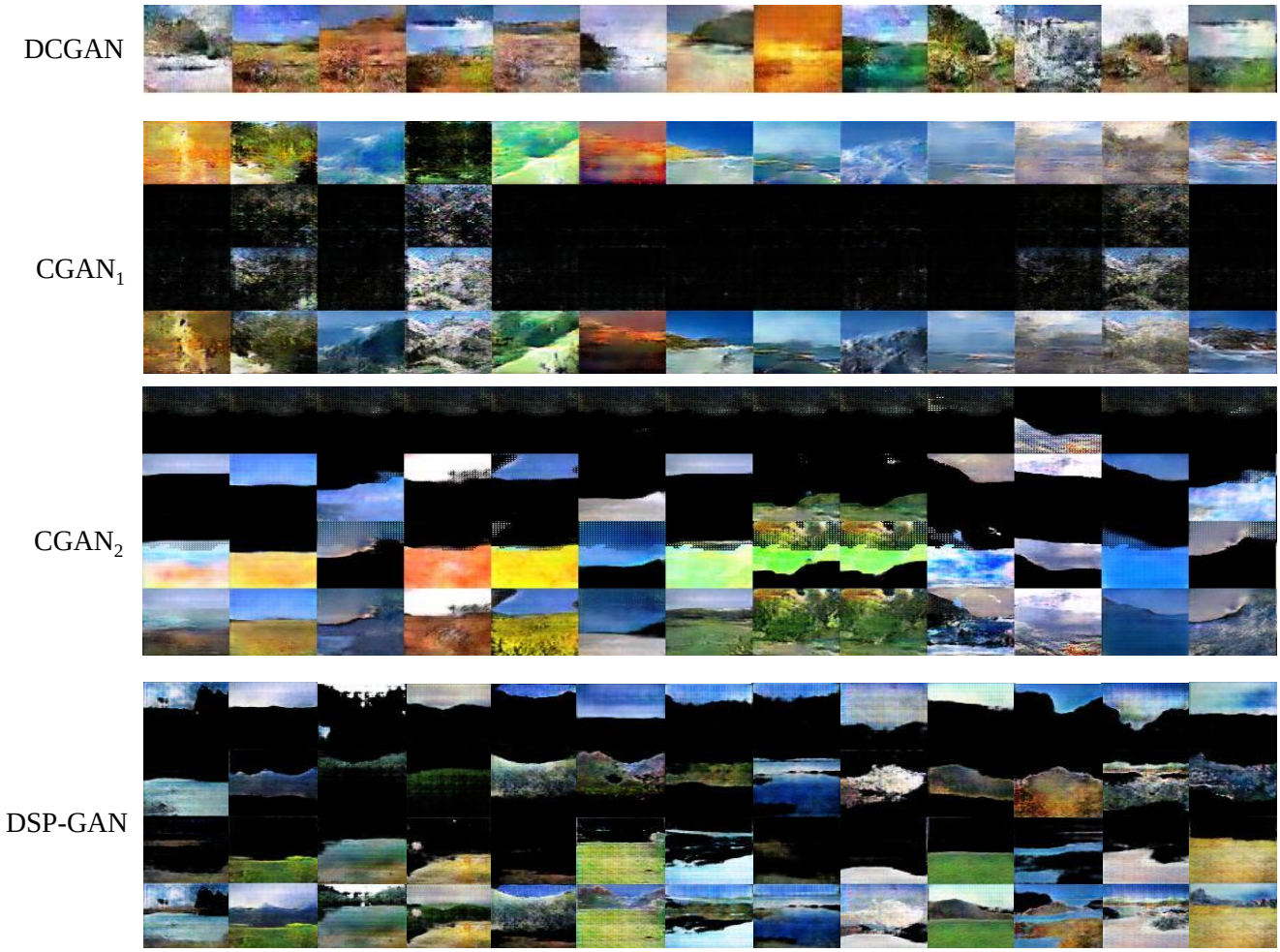

Figure 8. Comparison with other methods. The CGAN1 shows the results which are generated from noise and the CGAN2 shows the results generated from layer sequence.

natural scene images with high complicated spatial distribution.

References

[1] H. Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969. 2

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2

[3] A. Brock, T. Lim, J. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 1, 2

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016. 2

[5] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 2

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[7] A. Graves. Offline arabic handwriting recognition with multidimensional recurrent neural networks. In *Guide to OCR for Arabic scripts*, pages 297–313. Springer, 2012. 2

[8] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2

[9] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, pages 83–90, 1971. 2

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 2

[12] H. Kwak and B.-T. Zhang. Generating images part by part with composite generative adversarial networks. *arXiv preprint arXiv:1607.05387*, 2016. 2, 5, 7

[13] P. J. Laub, T. Taimre, and P. K. Pollett. Hawkes processes. *arXiv preprint arXiv:1507.02822*, 2015. 2, 3

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016. 1, 2

[15] D. Marsan and O. Lengline. Extending earthquakes' reach through cascading. *Science*, 319(5866):1076–1079, 2008. 2

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[17] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016. 1

[18] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[19] T. T. Pham, S. Hamid Rezatofighi, I. Reid, and T.-J. Chin. Efficient point process inference for large-scale object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2016. 2

[20] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017. 2

[21] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 7

[22] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11):115013, 2011. 2

[23] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2

[24] L. Theis and M. Bethge. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems*, pages 1927–1935, 2015. 2

[25] I. M. Toke. " market making" behaviour in an order book model and its impact on the bid-ask spread. *arXiv preprint arXiv:1003.3796*, 2010. 2

[26] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010. 2, 6

[27] H. Xu, Y. Zhen, and H. Zha. Trailer generation via a point process-based visual attractiveness model. In *IJCAI*, pages 2198–2204, 2015. 2

[28] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016. 2, 3

[29] J. Yang, A. Kannan, D. Batra, and D. Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017. 2, 3

[30] S. Yousefi, N. Kehtarnavaz, and A. Gholipour. Synthesis of cervical tissue second harmonic generation images using markov random field modeling. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 6180–6183. IEEE, 2011. 1, 3

[31] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[32] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, volume 31, pages 641–649, 2013. 2, 4, 5