DeepSpace: Mood-based Image Texture Generation for Virtual Reality from Music

Misha Sra, Prashanth Vijayaraghavan, Ognjen (Oggi) Rudovic, Pattie Maes, and Deb Roy
MIT Media Lab, Massachusetts Institute of Technology, Cambridge, MA, USA
{sra,pralav,orudovic,pattie,dkroy}@media.mit.edu

Abstract

Affective virtual spaces are of interest for many VR applications in areas of wellbeing, art, education, and entertainment. Creating content for virtual environments is a laborious task involving multiple skills like 3D modeling, texturing, animation, lighting, and programming. One way to facilitate content creation is to automate sub-processes like assignment of textures and materials within virtual environments. To this end, we introduce the DeepSpace approach that automatically creates and applies image textures to objects in procedurally created 3D scenes. The main novelty of our DeepSpace approach is that it uses music to automatically create kaleidoscopic textures for virtual environments designed to elicit emotional responses in users. Specifically, DeepSpace exploits the modeling power of deep neural networks, which have shown great performance in image generation tasks, to achieve mood-based image generation. Our study results indicate the virtual environments created by DeepSpace elicit positive emotions and achieve high presence scores.

1. Introduction

Generating virtual environments (VEs) is of great interest for many 3D applications, such as games, rehabilitation, and entertainment [39]. Image creation for texture mapping is an integral part of VE design. Images contain a lot of rich abstract semantic information about objects, scenes, activities, and moods. Among many nuanced pieces of information in an image, there has been an increased interest in the image's tone or mood. While humans can perceive and understand images at the affective and cognitive levels [15], affective image analysis usually targets low level visual features such as color, texture, shape, and line. Recent work [5, 50] has explored the importance of understanding the relationship between artistic principles and emotions in an image. However, this is not trivial as it involves bridging the gap between affective content and the user's perception.

The first step in creating a VE for a virtual reality (VR) experience is building 3D models of objects that will be part of the VE. This is usually done using 3D modeling software (e.g., Maya, Cinema 4D). The objects are then imported into a game engine like Unity 3D [1]. This is followed by specifying material properties and textures for all the objects in the scene including the sky and and ground terrain. The textured models along with the terrain and skybox define the shape and look of the VE while the lights and color define its mood. One also needs to consider the aesthetic and affective appearance of the scene as a whole. Despite the availability of a variety of consumer virtual reality (VR) devices with different setups, developing VR applications remains a difficult and time-consuming task requiring mastery of various tools and high-expert skills. One way to facilitate VE creation is to do it automatically from 3D scans of real world environments [42].

A critical step in creating VEs is the image generation process. There has been tremendous progress in modeling and learning image representations [14, 26]. Modeling the image distribution is a challenging task due to statistical dependencies over several pixels in the image. With recent advances in deep neural networks, it is now possible to generate images, conditioned on descriptive labels or tags, that preserve the structure of image data and underlying context. However, it is still challenging to encode high-level affective states like moods in an image generation task. Therefore, learning generative models conditioned on mood and descriptive labels can allow us to create richer representations aimed at building more immersive VEs and, consequently, more dynamic affective experiences.

To address challenges mentioned above, in this work we focus on two main (related) tasks: (i) design of the system pipeline for automatic generation of VEs, and (ii) image generation approach that is mood-informed. These two elements are an integral part of DeepSpace, our novel approach to automatically generate surreal textures and artistic VR experiences using music data. The 3D scene design

[1] https://unity3d.com/

is based on the mood and content extracted from song audio and lyrics data, along with findings from our online user study. Lighting is added manually after the environment is procedurally generated. To generate textures for objects in the 3D scene, we created a Mood-Conditional PixelCNN (MC-PixelCNN), which is build upon the PixelCNN model [45] for image generation tasks. A PixelCNN is a Convolutional Neural Network (CNN) architecture that preserves the spatial resolution of its input through multiple network layers, and outputs a conditional distribution at each pixel location . The DeepSpace pipeline (see Fig. 1) consists of the following steps.

1. The mood of the song is identified by feeding Mel-frequency Cepstral Coefficients (MFCC) features [29] of the audio through a gated recurrent neural network (RNN) [8], noun phrases are extracted from the lyrics of the song using the Stanford part-of-speech tagger [25].
2. A image dataset is created from the results of a Google Image Search using mood-phrase pairs from step 1) as search terms, and the MC-PixelCNN model is trained using the image dataset, mood and phrase triples.
3. Textures are created by feeding the images generated by the MC-PixelCNN into a DeepDream[2] like deep CNN codenamed Inception [44].
4. A genetic algorithm is used to procedurally create a VE where the objects are automatically textured with the output from step 3.

To demonstrate the DeepSpace performance, we generated two VEs corresponding to the two broad mood categories: happy (positive mood) and sad (negative mood), as they are the most frequently felt emotions from music [19, 49, 41]. Additionally, generated images for those two moods were rated higher than, e.g., calm and angry (see Fig. 4).

2. Related Work

2.1. Image Generation

There has been on discriminative models in the past. Discriminative DNNs have shown great performance in various tasks, including image and speech recognition [22, 13], and machine translation [2]. More recently, generative models are starting to gain interest. Some of the earlier literature on generative models has mainly explored variants of Boltzmann machines [1, 38] and deep belief networks [17]. These models are generally powerful, however, they require approximation of the partition functions which can be intractable. Furthermore, they also do not scale well for large datasets. On the other hand, Variational Auto-Encoders (VAE) [21, 34] have received significant attention as generative models.

VAEs can be seen as a neural network with continuous latent variables, where the posterior distribution is approximated using an encoder and the reconstruction of data from the encoded latent representations is stochastically done using decoders. This adds flexibility to generation of new images, modulated with extra information, as we attempt in DeepSpace. In another example, [14] introduced Deep Recurrent Attention Writer (DRAW) neural network for image generation. DRAW networks allow iterative construction of images by combining a novel spatial attention mechanism with a sequential variational auto-encoding framework. One-dimensional LSTMs were used to generate images in a sequential manner. A recent work [26] focuses on generating images from natural language descriptions. The proposed model iteratively draws patches on a canvas, while attending to the relevant words in the description. The authors show that the model improves image generation for unseen captions in the dataset, compared to other existing approaches. Other types of generative models based on adversarial processes and their extensions have been proposed [12, 10]. However, most of these models are unconditional or conditioned on few categorical labels only, and not on descriptive labels or captions that can be words or phrases. Note that none of these works ever attempted image generation using mood and text information together, as done in our proposed DeepSpace.

2.2. Mood-Based Image Generation

There is a lot of work using discriminative models that associates images and emotions [18, 51, 50]. However, not much has been done on combining image generation with human emotions or moods using generative models. In Emonets [20], the authors perform sentiment analysis using multimodal deep learning techniques to predict emotions in videos. Progressively trained and domain transferred deep networks are used in [48] for image sentiment analysis. There has been relatively less attention give to mood based image generation. In DeepSpace, we focus on building generative models using mood and descriptive text labels for the image generation task. Our model explores conditional image generation with a new image density model.

2.3. Virtual Reality

Head mounted displays (HMDs) have gained recent popularity due to improvement in hardware and software and easy availability for consumers. Nature Abstraction[5] is an immersive experience that explores fractals produced in software, rendered using a 360-degree camera, and processed in DeepDream to transform the fractal landscapes into morphing patterns. The visuals in Dreamtime[6] are also

[2] https://en.wikipedia.org/wiki/DeepDream

[5] http://alt-o.com/Nature-Abstraction

[6] http://www.wjsims.com/

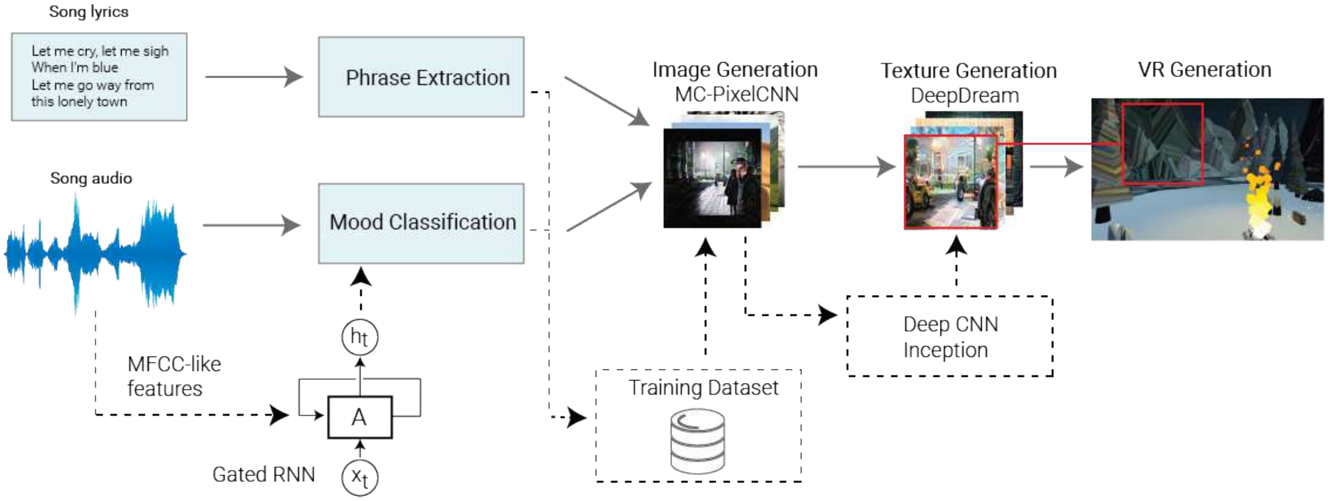

Figure 1. DeepSpace pipeline: mood and noun-phrases (e.g., lonely town) are extracted from song audio and lyrics respectively. The mood-phrase pairs (e.g., sad + lonely town) are used to build a training image dataset for training the MC-PixelCNN model. Images generated by the MC-PixelCNN are converted into surreal textures using a deep CNN with Inception modules. Texture shown in red box is applied to a mountain 3D model in the VE by changing its X/Y tiling values to create the striped pattern (see Sec. 3.4.1). The VE is experienced through an HTC Vive head-mounted display (HMD)[4].

achieved by using DeepDream with 360 degree film making techniques. Several data-driven techniques have been proposed for 3D scene content generation such as scene modeling[6], VE generation [42], and interactive synthesis of virtual worlds [11]. DeepSpace goes a step beyond the 360-degree experiences by allowing users to walk and interact in a psychedelic 3D world for a fully immersive experience. Our work shares the same spirit as the above content-generation methods but it addresses automatic VR scene creation with texture generation and assignment.

3. DeepSpace: The Model

Our goal is to model the distribution over natural images and generate new surreal andartistic images in our attempt to create experiential VEs that try to capture the essence of an emotion. Our model, MC-PixelCNN, a generalization of the PixelCNN[45]. The basic idea of the architecture of PixelCNNs (also PixelRNNs) is to use autoregressive connections to model images pixel by pixel, decomposing the joint image distribution as a product of conditionals. PixelCNNs are faster to train as convolutions are generally parallelizable. Given the large number of pixels in image datasets, this provides an important advantage. In our MC-PixelCNN model, we estimate the likelihood of images conditioned on latent vector embeddings representing descriptive labels and mood of the image, as described in Sec 3.1. The network scans the image in a row-wise fashion taking one pixel into consideration at a particular point in time. For each pixel it predicts the conditional distribution over the possible pixel values given the scanned context We use the standard convolution layers in order to capture a bounded receptive field. This helps in computing features for all pixel positions at once. The MC-PixelCNN model uses multiple convolution layers that preserve the spatial resolution. Masking is employed in the convolutions so that any computation of conditional distribution for the current pixel does not take into account the future context or the unscanned context. An important aspect of our generative model is that it involves conditioning on latent vector representations for descriptive labels and mood of the image to enhance the overall image generation task.

3.1. Descriptive Label & Mood Representation

We learn latent embeddings of the descriptive label and mood using unsupervised learning of label representations mined from text (see Sec. 4). These embeddings prevent costly manual annotation of attributes and can be used to generate images for words that are unseen during training based on nearest neighbors in the embedding space. Two commonly employed approaches are Word2Vec [27] and GloVe [32]. The former consists of a two-layer neural network trained to predict a list of target words based on a given context window of words. The first layer acts as a look up table to retrieve the embedding for any word in the vocabulary, while the second layer predicts the target words using hierarchical softmax or negative sampling. Embeddings are obtained by back-propagating the prediction error gradient over a training set of context windows sampled from the text corpus.

[6] https://www.vive.com

GloVe incorporates co-occurrence statistics of words that frequently appear together within the document. Semantically similar words occur together more frequently than semantically dissimilar words. Based on this, the training objective is set to learn word vectors such that their dot product equals the co-occurrence probability of the words. The GloVe [32] approach has recently been shown to outperform Word2Vec on the word analogy prediction task. We use the GloVe model trained on a common crawl dataset[7] for the representation for words in the descriptive labels and mood. We use one-hot vector representation for mood by transforming it into a vector space that matches the embedding size of the descriptive labels. These latent embeddings capture co-occurence statistics and project them in a semantic space (i.e., semantically similar words) are are used to condition images in our generative model. The one-hot vector is a binary vector $v \in \mathbb{R}^{|L|}$ where $|L|$ refers to the number of all mood labels. The vector contains a "1" in the position of ground-truth mood label and "0" in other label positions.

3.2. Mood Conditioned PixelCNN for Image Generation

Our MC-PixelCNN model extends the PixelCNN model [31, 45]. We model the conditional distribution of natural images given two latent vector representations of descriptive labels and mood as input (see Sec. 4), in addition to image pixels. We model the conditional distribution $p(x|h)$ of images x, given the phrase (P) and mood (M) information represented as the latent vectors h_P and h_M, respectively, as:

$$p(x) = \prod_{i=1}^{n^2} p(x_i|x_1, ..., x_{i-1}, h_P, h_M), \qquad (1)$$

where n is the number of pixels in x. In this model, every pixel depends on all the pixels above and to the left of it. Hence, the joint conditional distribution of pixels in MC-PixelCNN is modeled as a product of conditional distributions defined in Eq.(1). The factorization turns the joint modeling problem into a sequence problem, where one learns to predict the next pixel given all the previously generated pixels. The main difference, however, lies in the fact that in MC-PixelCNN, every conditional distribution is modeled by a standard CNN with masked convolutional filters in order to satisfy the dependency condition (Eq.1), and avoids future context to influence the current computation of the conditional distribution. A stack of such filters is applied over an input image $I \in \mathbb{R}^{N \times N \times 3}$, where height and width of the image is represented by N with 3 color channels (RGB). It is worth noting that there is no pooling layer involved in this process to preserve the spatial resolution. This approach of tractably modeling a joint distribution of pixels in the image as a product of conditional distributions has previously been adopted in autoregressive models such as NADE [23] and fully visible neural networks [3, 30]. However, no conditioning on extra variables was used in our approach.

3.2.1 Gated Activation Units

We use the gated activation units as proposed in PixelCNN decoders [45], which allow us to account for more complex interactions (e.g., latent interactions between pixels or groups of pixels, as well as the mood and descriptive label representations that we learn) via multiplicative units. Formally, we model the conditional distribution $p(x|h)$ of images, given the descriptive label (P) and mood (M) information represented as latent vectors h_P and h_M by computing the following:

$$\begin{aligned} z = tanh(W_{k,f} * x + U_{k,f}^T h_P + V_{k,f}^T h_M) \\ \odot \ \sigma(W_{k,g} * x + U_{k,g}^T h_P + V_{k,g}^T h_M) \end{aligned} \qquad (2)$$

where σ is the sigmoid non-linearity, k is the layer number, h_M is a one-hot encoding that specifies a mood class, and h_P is a descriptive label representation computed by summing the GloVe word-vector representation [24]. Here, f and g denote the filter and the gate, respectively, and $W_{k,f}$ is the convolution filter. $U_{k,f}$ and $V_{k,f}$ are weight parameters associated with vector representation of descriptive labels and mood respectively. Given a high-level image descriptive label represented as a latent vector h_P along with a mood vector h_M, we seek to model the conditional distribution $p(x|h_P, H_M)$ of images suiting a specific mood and descriptive label by adding terms that depend on h_P and h_M to the activations before the nonlinearities. This allows us to seamlessly encode the descriptive label and mood information. Recently, several works have explored *skip connections* to enhance the flow of information during forward and backward propagation. Parameterized skip connections used in Highway Networks (HNs) [43] allow representations learned from previous layers to flow unhindered to later layers, generally known as "information highways". Residual networks [16] simplify HNs by shortcutting with identity functions. This simplification greatly improves training efficiency and enables more direct feature reuse. *Skip connections* are also needed to address the vanishing gradient issue in deep networks. Both residual and parametrized skip connections are used throughout our network to speed up convergence and enable 'deeper' training.

3.2.2 Soft-max layer

To model the conditional distributions over the individual images, we use the softmax layer as recent work [31] shows that a softmax distribution tends to perform well,

[7] http://commoncrawl.org/the-data/

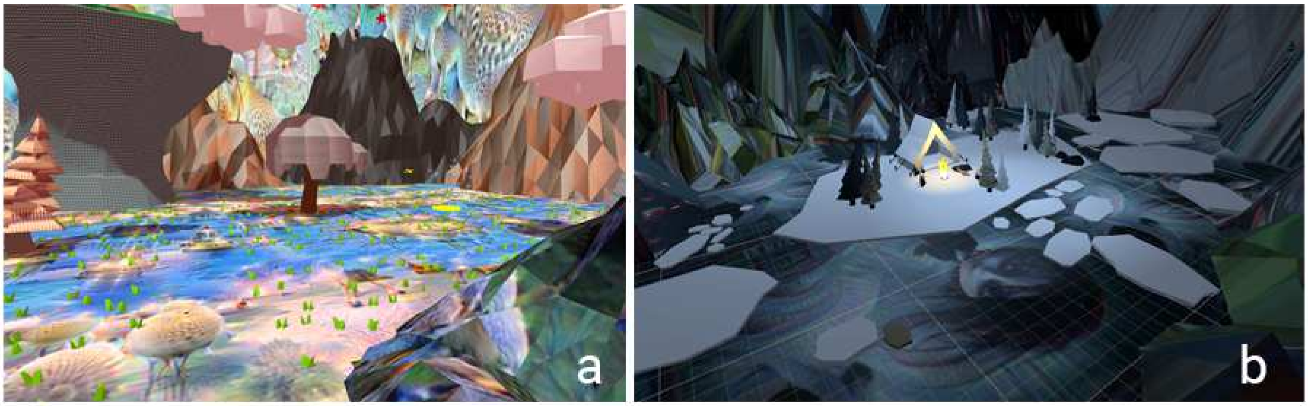

Figure 2. Two virtual environments generated to demonstrate the DeepSpace pipeline. a) VE generated from the song 'The Bird and the Worm by Owl City' which was classified happy. b) VE generated from the song 'Blue Prelude by Nina Simone' which was classified sad. The objects in each scene are based on the noun-phrases extracted from the song lyrics supplemented by findings from our online user study (see Sec 5). The textures are generated by a deep CNN and applied to objects during procedural VE generation.

even though the data is inherently continuous. Each pixel is modeled successively with green channel (G) conditioned on red channel (R) and blue channel conditioned on both red and green channels (R, G). We perform a 256-way prediction for the three color channels (R, G, B) by considering each channel of the feature map individually at every layer. The output of the MC-PixelCNN is $I_{out} \in \mathbb{R}^{N \times N \times 3 \times 256}$. Formally, the probability that a particular pixel position in the image takes a value v for a particular color channel c and a context C (where context C represents the previously scanned image pixels, mood and descriptive label representations) can be represented as:

$$p_c(I_{i,j} = v|C) = \frac{e^{z_v}}{\sum_{k=1}^{K} e^{z_k}} \tag{3}$$

where $I_{i,j}$ represents position (i, j) in the image I, $K = 256$ as each color channel c can take 256 different values, z is the representation obtained from the previous layer.

3.3. Deep Texture Generation

Texture generation, using this deep dream technique is not mandatory but desired for our VEs as it helps add an artistic and non-realistic touch to the images generated by our MC-PixelCNN which inherently tries to mimic real world images. Our goal is to create experiential VEs that allow users to experience the essence of a song. We believe the non-realistic output from this deep dream technique prevents a literal representation of a song's lyrics translated into textures that get applied to 3D objects in the VE.

We use a deep CNN architecture named Inception [28] to extract features from the images generated by the MC-PixelCNN. This architecture relies on increasing the depth and width of the network and was applied originally to improve the overall performance of the image classification task. It involves finding out how an optimal local sparse structure in a convolutional vision network can be approximated by readily available dense components of the network. In our case, it helps us increase the number of units at every layer without any uncontrollable overload of computational complexity. In our experiments, we use an Inception model pre-trained on the full ImageNet dataset [35] and let the network make the decision to select the feature that will be amplified using the pre-trained model parameters. Given an image, we run it iteratively and apply a zooming operation at each iteration to enhance specific features in that image detected at a specific layer. Since higher-level layers extract more sophisticated features, complex structures and objects tend to emerge. Thus, we generate 24 textures from a single input image using this iterative process.

3.4. Virtual Environment Generation

A simple VE is composed of 3D models of objects, environment elements like the terrain and skybox, and lights. Procedural generation techniques are often employed in video game design[8]. We designed the following pipeline for procedurally generating our VEs:

1. We use a Perlin noise [33] based terrain generator that allows us to create a variety of terrains. We use the default skybox object available in Unity 3D.
2. Using the noun-phrases extracted from the lyrics, combined with the findings from our online user study (see Sec. 5) asking participants to provide text descriptors of things they associate with happy and sad places, we select objects from a small database of tagged 3D models to place in the VE. The placement of objects is done

[8] https://en.wikipedia.org/wiki/Procedural_generation

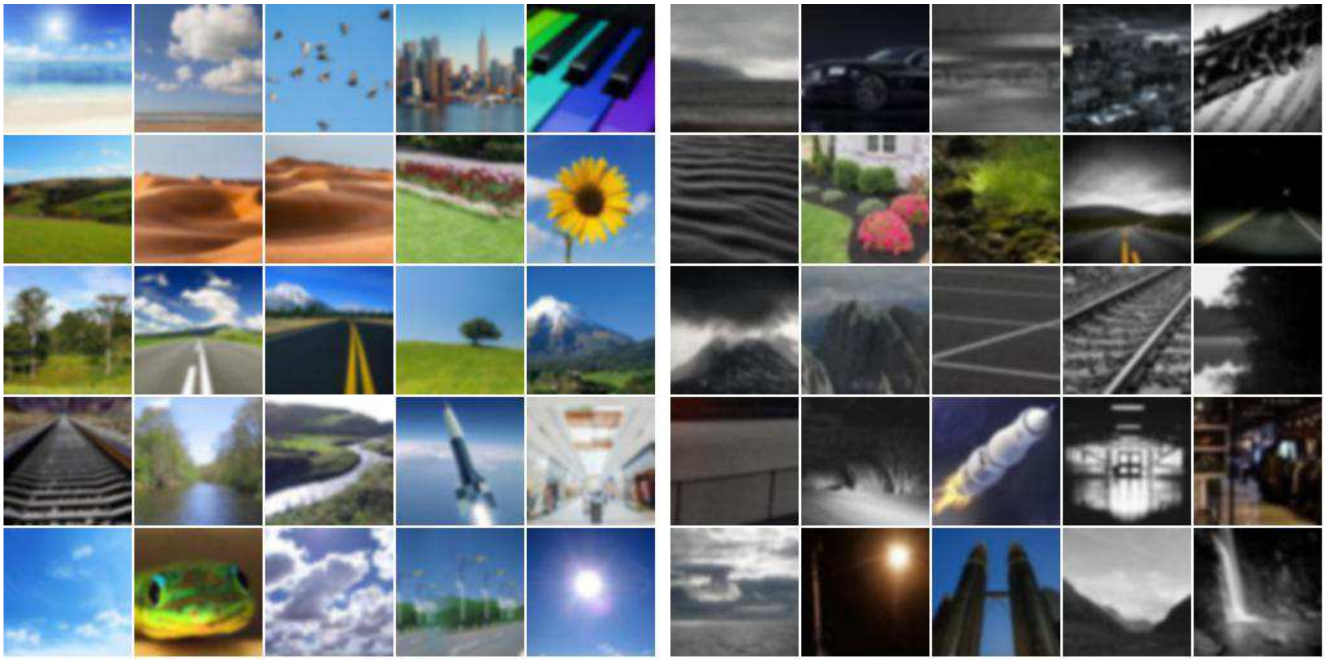

Figure 3. Left: Sample generated images for “happy” mood with descriptive labels: ocean, desert sand, rail track, mountain top, and forest land. Right: Sample generated images for “sad” mood with descriptive labels: towers, waterfalls, desert sand, road trip, and rail track.

using a genetic algorithm [46].

3. Textures resulting from the output of the deep CNN, encapsulating the mood and content of a song, are applied to all objects in the scene.
4. Information about lighting comes from the online user study that involves evaluation by external observers (see Sec. 5). Lights are added manually after scene generation based on user input in the study.

Since our goal was to create artistic VEs, we used low poly 3D models for their desirable blocky appearance. For an aesthetically pleasing arrangement of scene elements, we optimize the positions of 3D objects using a set of designed rules. A genetic algorithm (GA) [46] with elitism is employed to model the optimization function to allow for more sophisticated placement of elements than a simple heuristic/random approach. We create a set of 23 rules that define spatial relationships between sets of environment elements and object elements as shown in Fig.2. The rules take into account orientation relative to the center of the VE, where the user begins the VR experience when they put on the HMD.

3.4.1 Texture Mapping

The visual appearance of 3D models in Unity is controlled by materials made of shaders and textures. We create a new material by randomly selecting one texture from the set of 24 textures that are created by the deep CNN for each input image. While each input image generates 24 textures, the number of input images to the deep CNN depends on the selected song as they are generated from the mood and noun-phrases extracted from the song and fed through the MC-PixelCNN. During material creation we alter the X and Y tiling values from the default 1:1 ratio to create a variety of horizontal and vertical texture patterns. When the tiling is left at 1:1, the texture is visible as a checkered pattern on the object. Some textures are animated at runtime to enhance the psychedelic nature of the VR experience.

4. Dataset

We created our own database of images using moods extracted from song audio and noun-phrases extracted from song lyrics. We use the Million Song Dataset (MSD) [4] that comes as a collection of meta-data such as song names, artists and albums, together with MFCC features and a set of other features like loudness and tempo extracted with the The Echo Nest API[9]. The mood categories are inferred by mapping the LastFM[10] tags associated with the songs in MSD. The mapping is done in a similar fashion as explained in the work by Corona et al [9]. These mapped songs are then used for training a mood classification model from song audio. Using the subset of songs that have been mapped to mood categories, we use the MSD to perform an artist and song title search on Flash Lyrics[11]. Non-english

[9] http://the.echonest.com

[10] http://www.last.fm

[11] https://www.flashlyrics.com/lyrics

song lyrics are filtered out. After filtering out non-English lyrics, we get a total of $\sim 75,000$ songs that are mapped to a mood category and have full song lyrics. We perform phrase extraction on these song lyrics to be used as descriptive labels for our image generation task.

Mood Classification. We feed the MFCC features of the songs into a gated recurrent network (GRU) [8] to predict the mood category from the song audio. We use this approach as it is computationally less expensive than Long Short Term Memory (LSTM) networks and performs better than a standard RNN [7, 8] in target task. At each time step t, the GRU unit takes a row of the MFCC feature segment x_t and a hidden state h_t as input. The internal transition operations of the GRU are defined as $h_t = GRU(x_t, h_{t-1})$. The final hidden state (h_T) is fed to a fully connected layer followed by a softmax layer that outputs one of the four mood categories (happy, sad, angry, calm) [36].

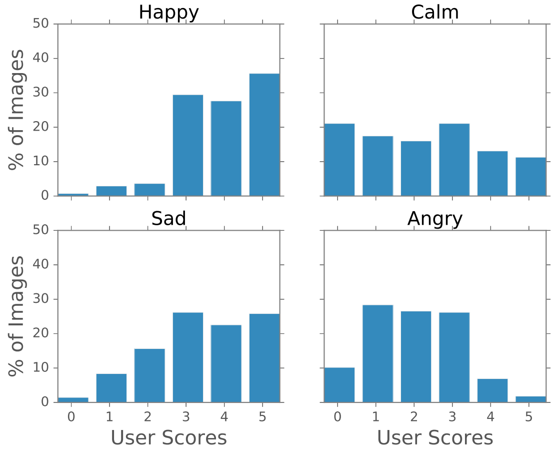

Figure 4. Histogram of users' scores for images associated with the four mood classes used in our mood classification task. The four classes are from Russell's circumplex of affect [36].

Phrase Extraction. Song lyrics are preprocessed by removing stop words, infrequent words, and tokenzing the sentences. The Stanford POS tagger [29] is employed assign parts of speech tags to each token such as noun, verb, and adjective. Based on these tags we get a unigram list of nouns. Given a piece of text and its POS tags, we perform Noun-Phrase or NP-Chunking on the tagged results of the POS. There may exist duplicates between extracted noun-phrases and the unigram list of nouns. We remove the duplicates from the unigram list of nouns because labeled noun phrases are more descriptive that just tagged nouns. The final list of words after de-duplication forms the candidate descriptive labels to train our generative model.

Using the top 5000 words from the final list of noun-phrases suffixed with mood terms like 'happy' or 'sad', we search and download the results from Google Images. Search terms that do not result in a minimum of 100 images per mood and noun-phrase pair are removed. A random manual curation is performed before using the data for training the MC-PixelCNN model. The final result is an image dataset which we further augment during training.

Training MC-PixelNN. We explore class-conditional modeling of the images in our database using our MC-PixelCNN. The descriptive labels obtained from the noun-phrases obtained from the lyrics are converted into fixed size vectors using GloVe vectors. In order to condition on mood, in addition to the descriptive labels, we use one-hot encoding for representing them as a vector. To map the mood into a vector of similar dimension as the descriptive labels vector, we get GloVe vector representations of the mood terms. The collected images along with mood and noun-phrase data are used to train the MC-PixelCNN model. The data is further augmented to reduce overfitting by enlarging the dataset artificially by performing label preserving transformations. We perform image translations and horizontal reflections, executed randomly during the training. on the original images.

5. Evaluation and Results

The Online User Study. Since it is difficult to quantify the performance of the proposed generative approach, we focus on the qualitative analysis. We observe that the generated images have good visual quality for the corresponding descriptive label and mood. Sample images generated for happy and sad moods with different noun-phrases are shown in Fig. 3. The tone of the images shows high contrast between each mood. The images are quite distinct from one another and the corresponding objects and backgrounds are clearly produced. We also note that the images for each descriptive label are diverse and our model is able to generalize and produce new renderings. Our observations are validated by a user study (see Fig. 4) with 55 external observers who were shown 20 generated images for each of the four broad mood categories (happy, sad, angry, calm) in a random order (see Fig. 5). The online user study is also used to collect text data about peoples' associations between moods, place and things. That information is used to decide which objects to add to the virtual worlds, in addition to noun-phrases extracted from song lyrics. The information also helps inform the design of lighting in the VEs. For example, descriptors like bright and sunny were entered for happy moods by several observers while dark and night were entered for sad moods. Thus, the happy scene has bright sunlight while the sad scene is dimly lit, both of which were manually added after scene generation.

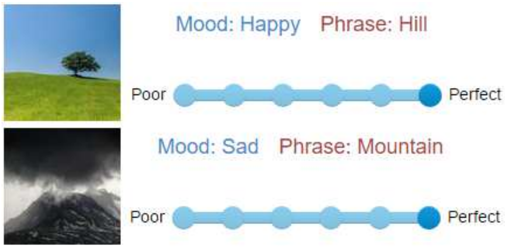

Figure 5. Two generated images for happy/sad moods as shown to participants in the online user study.

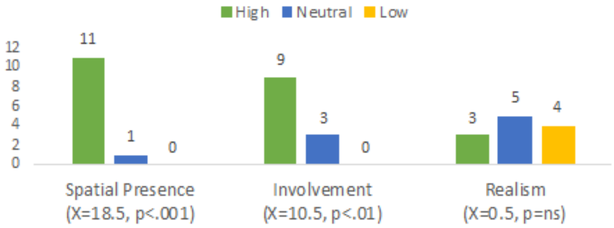

Figure 6. The distribution of participants across three categories from the Presence Questionnaire [47]. Below each category are the chi-square test ($\chi^2_{(2,N=12)}$) results.

Virtual Reality User Study. We conducted a pilot VR study with 2 scenes corresponding to happy and sad moods. 12 volunteers (ages 22-45, mean 34, 8 female) were recruited through email to participate in a the pilot to test the VEs generated using the proposed automatic approach (see Fig.2). A HTC Vive HMD was setup in a 2.5x2.3m tracked space in our lab. The study setup included a happy song (S1) used to generate VE1 and a sad song (S2) used to generate the VE2. 6/12 participants had never tried VR before. After the initial orientation, participants were asked to listen to S1 or S2, followed by a question: "*Thinking about the song you just heard, please describe how it made you feel.*" Then they viewed VE1 or VE2. If they listened to S1 they viewed VE2, and if they listened to S2 they viewed VE1. This was done to reduce the impact of the emotional effect of listening to a song on the user's experience of the VE as we wanted to learn if we had successfully generated affective VEs. After the VR experience, users were asked to answer a similar question as above but related to the VE. Responses to both these questions capture the user's mood perception of the song and of the virtual scene.

The overall rating of presence[12] is derived from the average of ratings for questions from the Presence Questionnaire (PQ) [47] divided into three factors: *Spatial Presence*, which is related to the sense of being in the VE, *Involvement*, which describes the VE's richness, and *Realism*, which is the consistency of information in the VE with the objective world [47]. The reported overall rating of presence across all participants was 5.11/7, SD=.81), with the highest being 7. Across the three factors, the average ratings were high for spatial presence (M=5.87/7, SD=.63) and involvement (M=5.22/7, SD=.51) and medium for realism (M=4.25/7, SD=.91). This indicates that our approach for automatically generating and texturing the VEs resulted in VR experiences where participants were spatially present [40] and highly engaged.

Emotional Responses. While the qualitative responses correlate well with our hypothesis, in order to get a quantitative measure of the text responses, we use the NRC Emolex dataset [37]. We have each user's textual responses to the two questions about how the song and the VE made them feel. In order to map these responses with emotions, we use the NRC Emotion Lexicon (EmoLex) dataset. We tokenize each response and use Emolex to associate it with a distribution over emotions. Since we focus on happy and sad moods in our user study, we compare the "joy" and "sadness" emotions and their association with the user's response and classify if the response is associated with happy or sad moods. We find that 84% of the users felt "joy" listening to the happy song S1 and viewing the happy scene VE1. However, though 84% of the users felt "sadness" listening to the sad song, only 33% of them felt "sad" viewing the sad scene VE2 (see Fig. 2). We attribute this to the fact that both the VEs were highly artistic in nature and were described as "imaginative" and "beautiful."

6. Conclusion

In this work we presented DeepSpace, the first automatic approach to create immersive virtual reality environments from song data that can be experienced in an HMD. The system is able to generate textures that encode mood extracted from song audio, and noun-phrases extracted from song lyrics, and apply them to objects in the 3D scene. The scenes themselves are procedurally generated. Our preliminary study shows that our VEs evoke positive emotions in users while eliciting a high sense of presence, the hallmark of any VR experience. There is much room for improvement as well as opportunity for further development of automated processes for creating affective VEs.

Acknowledgements

The work of O. Rudovic is funded by European Union H2020, Marie Curie Action - Individual Fellowship no. 701236 (EngageMe).

[12] Typically used to quantify the VR performance [40].

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 2

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[3] Y. Bengio and S. Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. In *NIPS*, volume 99, pages 400–406, 1999. 4

[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *ISMIR*, volume 2, page 10, 2011. 6

[5] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232. ACM, 2013. 1

[6] K. Chen, Y. Lai, Y.-X. Wu, R. R. Martin, and S.-M. Hu. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics*, 33(6), 2014. 3

[7] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 7

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2, 7

[9] H. Corona and M. P. O'Mahony. An exploration of mood classification in the million songs dataset. In *12th Sound and Music Computing Conference, Maynooth University, Ireland, 26 July-1 August 2015*. Music Technology Research Group, Department of Computer Science, Maynooth University, 2015. 6

[10] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2

[11] A. Emilien, U. Vimont, M.-P. Cani, P. Poulin, and B. Benes. Worldbrush: Interactive example-based synthesis of procedural virtual worlds. *ACM Transactions on Graphics (TOG)*, 34(4):106, 2015. 3

[12] 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

[13] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013. 2

[14] 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. 1, 2

[15] A. Hanjalic. Extracting moods from pictures and sounds: Towards truly personalized tv. *IEEE Signal Processing Magazine*, 23(2):90–100, 2006. 1

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[17] G. E. Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009. 2

[18] D. Joshi, R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Z. Wang, J. Li, and J. Luo. Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5):94–115, 2011. 2

[19] P. N. Juslin and P. Laukka. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33(3):217–238, 2004. 2

[20] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016. 2

[21] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[23] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In *AISTATS*, volume 1, page 2, 2011. 4

[24] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015. 4

[25] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014. 2

[26] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015. 1, 2

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3

[28] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 20:14, 2015. 5

[29] M. Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007. 2

[30] R. M. Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992. 4

[31] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 4

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014. 3, 4

[33] K. Perlin. Improving noise. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 681–682. ACM, 2002. 5

[34] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015. 2

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[36] J. A. Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003. 7

[37] M. Saif. *NRC Emotion Lexicon*, 2017 (accessed May 19, 2017). 8

[38] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, volume 1, page 3, 2009. 2

[39] M. Slater and M. V. Sanchez-Vives. Enhancing our lives with immersive virtual reality. *Frontiers in Robotics and AI*, 3:74, 2016. 1

[40] M. Slater and S. Wilbur. A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments*, 6(6):603–616, 1997. 8

[41] J. A. Sloboda. Empirical studies of emotional response to music. 1992. 2

[42] M. Sra, S. Garrido-Jurado, C. Schmandt, and P. Maes. Procedurally generated virtual reality from 3d reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 191–200. ACM, 2016. 1, 3

[43] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015. 4

[44] 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*, pages 1–9, 2015. 2

[45] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 2, 3, 4

[46] D. Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994. 6

[47] B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and virtual environments*, 7(3):225–240, 1998. 8

[48] Q. You, J. Luo, H. Jin, and J. Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. *arXiv preprint arXiv:1509.06041*, 2015. 2

[49] M. Zentner, D. Grandjean, and K. R. Scherer. Emotions evoked by the sound of music: characterization, classification, and measurement. *Emotion*, 8(4):494, 2008. 2

[50] S. Zhao, Y. Gao, X. Jiang, H. Yao, T.-S. Chua, and X. Sun. Exploring principles-of-art features for image emotion recognition. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 47–56. ACM, 2014. 1, 2

[51] S. Zhao, H. Yao, X. Jiang, and X. Sun. Predicting discrete probability distribution of image emotions. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2459–2463. IEEE, 2015. 2