Eurographics Symposium on Rendering (DL-only Track) (2022)
A. Ghosh and L.-Y. Wei (Editors)

SkyGAN: Towards realistic cloud imagery for image based lighting

Martin Mirbauer†1, Tobias Rittig†1, Tomáš Iser1, Jaroslav Křivánek1,2, and Elena Šikudová1

1 Charles University, Faculty of Mathematics and Physics, Czech Republic
2 Chaos Czech a.s., Czech Republic

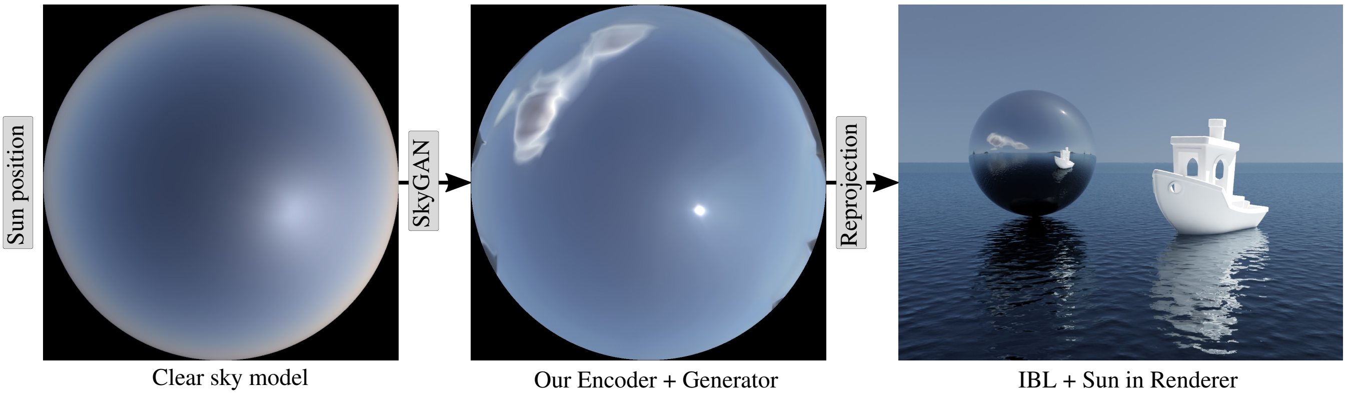

Figure 1: *Our method generates cloudy sky images from a user-chosen sun position that are readily usable as an environment map in any rendering system. We leverage an existing clear sky model to produce the input to our neural network which enhances the sky with clouds, haze and horizons learned from real photographs.*

Abstract

Achieving photorealism when rendering virtual scenes in movies or architecture visualizations often depends on providing a realistic illumination and background. Typically, spherical environment maps serve both as a natural light source from the Sun and the sky, and as a background with clouds and a horizon. In practice, the input is either a static high-resolution HDR photograph manually captured on location in real conditions, or an analytical clear sky model that is dynamic, but cannot model clouds.

Our approach bridges these two limited paradigms: a user can control the sun position and cloud coverage ratio, and generate a realistically looking environment map for these conditions. It is a hybrid data-driven analytical model based on a modified state-of-the-art GAN architecture, which is trained on matching pairs of physically-accurate clear sky radiance and HDR fish-eye photographs of clouds. We demonstrate our results on renders of outdoor scenes under varying time, date, and cloud covers.

CCS Concepts

• ***Computing methodologies*** → ***Rendering; Supervised learning;*** • ***Applied computing*** → *Earth and atmospheric sciences;*

1. Introduction

In photorealistic rendering, scenes are commonly surrounded by environment maps, a concept also referred to as image-based lighting (IBL) [Deb98]. The pictures serve two purposes: First, they provide information about the directional illumination in the scene, and second, the 360° images are used as the visible background. Captured imagery or analytical sky models are used in practice, such as the widely adopted Hošek-Wilkie [HW12; HW13] or the more general model by [WVB*21]. While clear sky models serve excellently for realistic illumination, they can be perceived as too simple as a background. For added realism, VFX artists, architects,

[†] Contributed equally.

and other rendering users are also looking to incorporate clouds into the sky, leaving them with expensive volumetric cloud simulation, or static photographs. Such photographs (termed "HDRIs") need to cover the full dynamic range of the sky and, as such, are expensive to capture: a lot of manual effort and professional gear is required for high-quality results. At the same time, the images are static, meaning limited to the location and weather conditions at the time of capture. The user cannot parametrically change the appearance of the sky, as in the analytical models. That's why finding the appropriate match for requirements such as lighting (mood) or artistic composition of the background scenery is a manual, linear search in a database of pictures [TYS09; CZR22].

We propose a hybrid data-driven generative approach. It is based both on an analytical sky model and a dataset of photographs, and it generates a skydome with realistically looking clouds for any desired sun position and cloud coverage ratio. Our pipeline first uses the analytical model to generate a clear sky image corresponding to the sun position, and feeds it into an encoder-generator network, which generates a corresponding cloudy image. The cloudy image can then be used as a hemispherical environment map in a renderer. The pipeline is trained on high dynamic range (HDR) images, so it produces HDR output crucial for photorealistic rendering.

The state-of-the-art generative adversarial network (GAN) architecture [KAL*21] forms the core of our method, and we propose several domain-specific modifications to it. These mainly serve to condition the generator for the sun position input parameter, but also to support accuracy in the output. As usual, the discriminator loss is in place to produce fake imagery that closely resembles features of the training dataset despite being randomly generated. Additionally, we employ an autoencoder (reconstruction) loss to constrain the generator on the clear sky, for which we leverage an existing accurate model [WVB*21].

In this paper, we take the initial steps towards reaching the ultimate goal: a fully automated model, which generates a corresponding realistic and physically-accurate HDR skydome for a given artist's input (parameters). The results presented in this paper are still limited in quality due to several constraints that we discuss in the text, but these are not fundamental issues of our approach. With this intermediate report we hope to share the hurdles we encountered so far and gather feedback on the proposed solutions.

Our contributions include:

- A directly parameterizable cloudy sky model based on a conditional GAN architecture
- Support for HDR images in the StyleGAN3 codebase
- A method for fitting the [WVB*21] clear sky model to real photographs
- A dataset of 33 000 HDR sky photographs in 30s intervals

2. Related Work

In previous publications, generating skydomes and image-based lighting (IBL) was mainly solved by atmospheric clear sky models or machine learning approaches.

2.1. Atmospheric Models

In high-quality rendering of outdoor scenes, accurate skydome illumination and sky colors can be achieved by using an atmospheric model. One can either perform highly accurate brute-force Monte Carlo simulations based on first principles, evaluating light transport in the atmospheric gasses, or use much faster analytical models that can be directly evaluated with a potentially lower accuracy. We provide a brief overview in this section, but we also refer the reader to [Bru16] for an evaluation of analytical and brute-force models.

Brute-force solutions Probably the most accurate atmospheric simulations are available in the `libRadtran` research package [EBK*16], which can serve as a reference, but is too complex and slow for a direct use in computer graphics. Methods more suited for image rendering, such as [HMS05; BN08; GGJ18], usually include pre-computation steps that later allow more efficient evaluations during the actual path tracing. The main benefit of brute-force simulations is that they are physically accurate for any given sun position. However, path tracing of atmospheres is a very slow process, and furthermore, the models do not directly support rendering of clouds and overcast skies. For that, one would render fully volumetric clouds from a simulation [HMP*20] with expensive volumetric path tracing, or use machine learning [KMM*17] for improved efficiency.

Analytical and empirical solutions Our method is closer to analytical atmospheric models, which are not strictly physically accurate, but still result in realistically looking images with a very high performance. Unlike our method, they are limited to clear cloudless skies. They are usually based on fitting parametric functions to reproduce the actual sky radiance patterns. One of the first widely used models was the Preetham model [PSS99], which was directly based on older brute-force and analytical models. It was later improved in [HW12] to support more accurate sunset and high-turbidity settings, and in [HW13], by adding accurate solar radiance from the solar disk itself. The authors of [LM14] empirically alter the Preetham model to match also the overcast skies. A new model based on tensor decomposition was recently introduced [WVB*21], supporting different observer altitudes, post-sunset conditions, in-scattered radiance and attenuation for finite distances, and polarization. We use the unpolarized ground-level version of this model to produce the matching synthetic clear sky images for our real photographs.

2.2. Machine Learning for IBL

In Computer Vision, there are many methods for generating environment maps using deep-learning tools, especially on the task of lighting estimation. There, the posed problem consists of estimating the spherical scene illumination from narrow field of view images, which can, in turn, be used to render a virtual object into the scene with plausible shadows, reflections, and colors. Input images are conventionally low dynamic range (LDR), while output imagery is always HDR. A comprehensive survey on the topic can be found in [EGH21], and in the following, we will highlight a few methods that overlap with our approach.

In 2017, [HSH*17] proposed a convolutional neural network

(CNN) to fit the parameters of the Hošek-Wilkie model from an exemplar image. Conceptually similarly, [ZSH*19] use a slightly more expressive empirical model (Lalonde-Matthews) [LM14] to improve the quality of overcast skies. As both methods rely on analytical clear sky models, the output imagery does not contain any clouds. [HAL19] try to overcome this limitation by proposing a new data-driven sky model that learns the features of cloudy skies from hemispherical HDR photographs using an autoencoder architecture. Our method follows a similar path, but we still leverage the expressiveness and controllability of a clear sky model to form a hybrid approach, and we aim more on also reconstructing the proper cloud shapes.
Contrary to finding parameters to a fixed model, [SK21] formulate the lighting estimation problem as a task for spherical image extrapolation given a partial observation of the scene. Similar to our architecture, the authors employ a convolutional autoencoder jointly with an adversarial discriminator, and output HDR data. Another major difference of our work to all the methods above is that in Computer Graphics, we aim not only at the plausibility of diffuse and glossy reflections and shadows given a photographed backdrop, but the whole environment map needs to look photorealistic when directly observed.

2.3. GANs and generating cloud images

A generative adversarial network (GAN) [GPM*14] consists of two neural networks – a generator and a discriminator – which compete against each other in producing and detecting fake imagery respectively. This architecture has reached increased popularity and technologically matured over the past years. GANs are used in unsupervised and semi-supervised tasks such as the “image-to-image translation” [ZPIE17] where two classes of images should be converted into each other despite not having a perfect match between individual training samples. The task we are solving is similar but we benefit from having matched image pairs. Our architecture can also be seen as a form of a conditional GAN [MO14; DWX*20] where we enforce the generator to output a matching cloudy sky given a clear sky as input.

GANs are well known for photorealistic results when trained long enough on tens of thousands of images. With adaptive discriminator augmentation (ADA) [KAH*20] it becomes possible to have datasets of even just a few thousand images and still avoid overfitting to a particular training set.
Karras et al. [KAL*21] recently proposed a solution to the longstanding problem of textures “sticking” to the underlying pixel grid. They redesign the generator architecture with respect to fundamental signal processing rules to avoid any sources of aliasing. A detailed analysis of their method is provided in the next section.

Generating cloud images In atmospheric science, GANs are used for short-term forecasting of cloud coverage. Given previous frames of a video sequence, the authors of [ATO*19] predict how the clouds will move in the upcoming frames. They work on fisheye images directly out of the camera, similar to our raw dataset, but we process the projection to be a *stereographic* projection that has known properties. Although our dataset also consists of sequential images, we do not make use of the time dependency yet. For an image-based relighting approach in neural rendering, Yu et al. [YME*20] employ a GAN that fills the image background with realistic sky imagery given a segmentation map. For cloud image segmentation a GAN is used to augment the training dataset and produce ground-truth segmentation maps [MCS21].
There is prior art that is directly related to our approach. We build upon initial works [Hoj19; Špa20] that apply GANs to cloud image generation but are missing the direct control over the sun position and the clear sky supervision. Concurrently to our work, a very similar approach has been published [SMDB22]. They use a U-Net autoencoder architecture to transfer clear sky images of the Hošek-Wilkie sky model together with a cloud segmentation map to realistic cloudy images. This segmentation map on the one hand allows for artistic control of the cloud placement but on the other hand also requires manual input for every picture to achieve realistic distributions. Our method does not allow for spatial control of the clouds, but generates plausible distributions for any input from a random generator.

3. Analysis

We base our method on the StyleGAN3 architecture [KAL*21] which we analyze in this section. The authors provide an interactive visualizer application that loads pre-trained network weights and lets the user tweak the network inputs. Then, one can visually observe the network output at every layer as an image and its spatial frequency analysis. We inspected pre-trained networks provided by the authors that generate human or animal portraits and made three important observations.

First, as described in their paper, the generator architecture is designed to avoid introducing aliasing and other artifacts related to the pixel grid. All signals are composed of a set of 2D basis functions that are randomly generated on network initialization. With each increased resolution, higher frequency content is allowed in, thus effectively refining the signal as it flows through the network. Translation and rotation of the signal are achieved by an affine transformation of the input random vector that influences each layer’s weights as well as the basis functions. In contrast, the discriminator architecture is strictly working on a pixel grid – thus exhibiting any problems that may come with this approach.

Second, the generator spends parts of its capacity to learn textures and parts to generate grid-like coordinate systems that carry semantic meaning. Only at the very last layers, do these two parts get interleaved and the final image is blended from the textures based on the spatial coordinate grid.

Third, along the blending seams, we observe visible “halo” artifacts that hint at remaining aliasing problems within the generator architecture. We first spotted this problem in early iterations of our training where visible seam artifacts were trying to pass as clouds. For well-converged networks such as the ones provided by the authors, this effect is most visible but not limited to high-frequency textures such as hair, fur, or beards where the discriminator has difficulties discerning between the artifact and the intended content. The fact that it is still visible in the long trained networks gives us reason to believe that these artifacts are systematic despite being suppressed by the discriminator over training time.

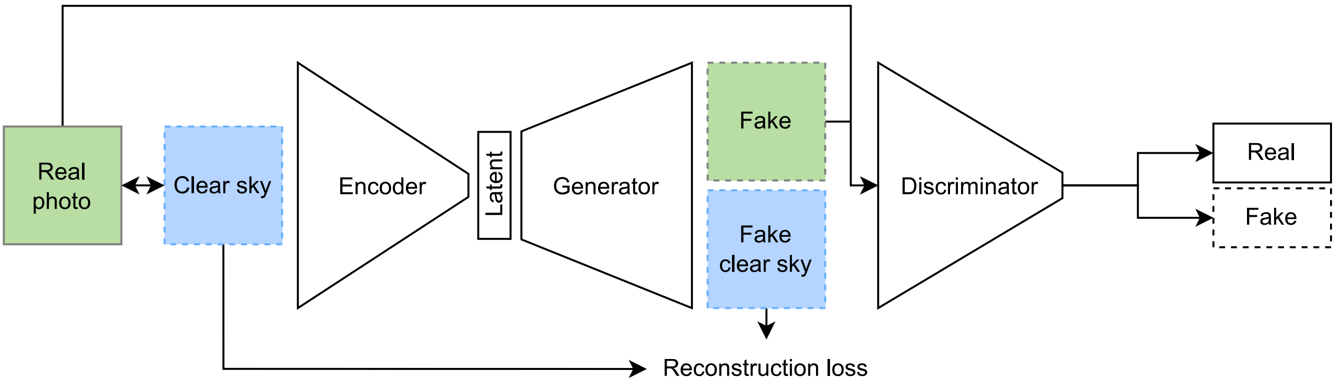

Figure 2: *Our network is trained on matching photos and fitted clear sky images. The clear skies are encoded into the generator's latent space and concatenated with random values. The generator reconstructs from this again a matching pair of the clear and cloudy sky. The quality of the clear sky is judged against the input clear sky, while the generated cloudy sky is judged by the discriminator for its realism.*

4. Method

Our method trains a generator for realistic cloudy sky images from a set of example pictures that we captured. In addition, it combines the training set with analytical clear sky images that correspond in the solar constellation and atmospheric conditions to the real photographs. The method builds on state-of-the-art GAN architectures, for which we propose modifications to take advantage of the matched image pairing and to ensure the outputs are usable in a rendering context.

We show our network architecture in Figure 2. At training time, we feed a clear sky image into an encoder to compress its information to a few numbers. Concatenated together with a random vector, these form the input to a generator that is tasked with producing two images – one clear sky reconstruction and the desired cloudy equivalent. The clear sky image can trivially be compared against the input clear sky image effectively forming an autoencoder loss. An adversarial discriminator network is trained in parallel to the generator and discerns between real and fake imagery effectively challenging the generator to produce more and more realistic looking cloudy skies.

As shown in Figure 1, during inference, the user can input the desired sun position and get a corresponding cloudy sky image. By adjusting the random seed for the latent vector, one can explore different cloud constellations. This works by generating a clear sky image from a state-of-the-art atmospheric model [WVB*21] given the desired sun position. This gets again encoded and combined into the random latent vector of the generator. Finally, the cloudy image from the generator's output can be re-projected to equirectangular projection and then used in a standard rendering pipeline as an environment map. When the images are not only used as a background, but provide the illumination for the scene, a high dynamic range becomes increasingly important. With clipped values, the renderings will exhibit reduced contrast that makes them look flat and unrealistic. For maximum realism, the sun values should not be clipped.

In the following, we will explain our method in more detail while keeping the order of data flow. We start with the dataset, before diving into the network architecture and the training procedure.

4.1. Dataset

The real photographs used during the network training are based on a selected subset of our dataset of HDR skydome photos. The photos were captured on a full-frame camera sensor by aiming an 8 mm circular fisheye lens upward towards the sky. Each hemispherical HDR photo was developed from an exposure stack of five to nine exposures. But even with the shortest possible exposure time, the sun's brightness is clipped on direct observation without the use of an neutral density (ND) filter [SJW*06]. The dataset is based on several locations, mostly in a central European climate and in coastal California. It includes both clear skies and various cloud covers ranging from small isolated clouds to fully overcast skies. The captures were performed in sequence (stop motion), one photo per 15-120 seconds (mostly 30s), sometimes for a very long time ranging from a sunrise to sunset. Around 33,000 HDR images were captured in total, from 54 different days distributed over 6 months (May-November).

It is important to note that for simplicity, our network was only trained on a limited subset of about 5,000 images for the time being. We picked skies with only a light cloud cover, mostly with high-altitude cirrus clouds, and where the sun is directly observable. We discarded most of the sunrise and sunset situations where the sun position could not be detected, as these also correspond to substantially different sky radiance. The light cloud cover corresponds more closely to the input clear sky model which intuitively decreases the difficulty for the generator. We refer the reader to the supplemental material for a preview of the training dataset. The whole dataset will be used in an extended version of this paper alongside which we also plan to release the dataset in full resolution (≥ 8192 px) and including the fitted metadata.

Before passing a photo from the training dataset to the network, it is first transformed from the fisheye projection on the camera sensor to a stereographic projection (in resolution 1024 × 1024), and the region outside the projected circle is masked out.

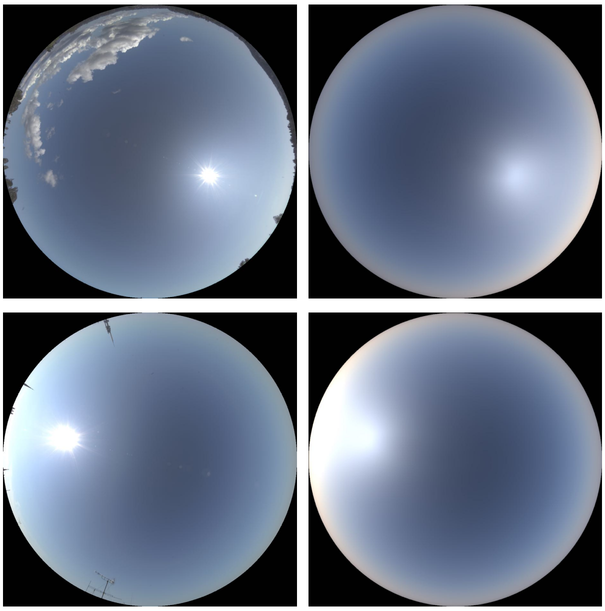

Figure 3: *Two example photographs from our dataset (left) with their fitted corresponding clear skies (right) – all in stereographic projection. The visibility distance mostly affects the aura around the sun, while the ground albedo brightens towards the horizon.*

Fitting a clear sky model The clear sky model by Wilkie et al. [WVB*21] provides incoming light intensity from queried directions and wavelengths, given the 2D sun position, the visibility distance (also called turbidity in older models), ground albedo and the altitude above ground. For simplicity, we use the hemispherical version of the model that is, just like older models, limited to sea-level observer altitude. As our dataset only contains sun positions above the horizon this should be enough information to learn also the lighting of the clouds. For post-sunset conditions one could additionally include clear sky images from the clouds' altitudes to inform the networks about the correct cloud illumination.

For each captured image in our dataset we detect the sun position by fitting an ellipse to the sun disk. The search region can be narrowed down by the expected sun position given the capture location, time and date. Due to calibration mistakes there might be minor mismatches that are corrected by this procedure. If the sun disk detection fails, we filter the image out of our dataset. For one exemplar image per day, we mask non-clear-sky objects like horizon, clouds, sun disk, lens flares and dust on the lens, finding the remaining clear sky model parameters using the BFGS algorithm with finite-difference gradients. The optimization has three steps: we first match the exposure, then run L-BFGS-B to find the model parameters constrained by their range, and finally we jointly fine-tune both the parameters and the exposure. We show results of this procedure in Figure 3. A related fitting procedure is described in [HSH*17, Section 4.2] for the Hošek-Wilkie sky model.

4.2. Architecture

On a high level the architecture is a hybrid between a Generative Adversarial Network (GAN) and an Autoencoder. We prepend the generator with an encoder to form an Autoencoder for clear sky images. At the same time, the generator also outputs cloudy images, which are judged by the adversarial loss of a discriminator network. We also apply modifications to the data processing such that it supports HDR values throughout the pipeline.

Encoder We base the encoder architecture on the architecture of the discriminator in our codebase [KAL*21] as both have a similar overall shape. Both take an image as input and reduce it gradually to a few numbers. Instead of outputting a single variable (real or fake score) like the discriminator does, the encoder is working with a bottleneck width of ten values. We estimate this should be more than enough information to describe the clear sky appearance which is inherently defined by model's four parameters. Ablation experiments have shown that a minimum of two values would be enough to reconstruct reasonably high quality results.

Generator The StyleGAN3 generator has been carefully designed to follow basic signal processing rules and avoid any source of aliasing throughout the layers of the network. We acknowledge this being a complex system whose parameters have been well tweaked to allow for the high quality results shown in their paper. For this reason, we keep modifications to the generator network to a minimum and again only adjust the final output layer. The output is extended to two corresponding images – one clear sky and one cloudy sky. Because the split between the two is only enforced at the last layer, the generator can benefit from the synergies between both images throughout all layers. This is a way of supervising the internals of the generator to produce specific patterns without interfering with the alias-free signal processing. Traditionally one would use losses at different resolutions of the generator to supervise the formation of intermediate patterns. In our case, this helps the generator to produce the unobstructed clear sky for which we have a ground truth and which should be the background of any cloudy sky image. As with the real photographs, we mask the image outside the projected circle of the stereographic projection.

Discriminator The discriminator remains unchanged from the StyleGAN 3 (originally from 2) codebase. We make use of the adaptive discriminator augmentation (ADA) [KAH*20] feature that prevents overfitting of the discriminator on small dataset sizes such as ours. The authors warn to enable only transformations that are valid within the domain of the images (e.g. X-axis flip) or else unwanted transformations (e.g. hue rotation) might leak into the Generator. However, we have not experienced that being an issue in our case despite enabling all transformations.

4.3. HDR values

HDR sky images can exhibit a very high value range while most parts of the sky have reasonable values below 1. Especially the sun, the aura, and any lens flares show a very high local contrast to the surrounding atmosphere. When working with HDR in neural networks it is common practice to transform the values using a compressive function such as a logarithm and then un-transforming it

for final output [EKD*17]. This prevents numerical issues such as exploding gradients while still allowing the network to produce big values albeit at reduced precision. For the training, we processed the images with a log transform and a fixed shift in order to squeeze all intensity values to a semi-open interval with a fixed minimum $[-1,\infty)$. In practice however most values lie in the interval $[-1,1]$ resembling a normal distribution curve with zero mean, while only the sun values reach up to 2. The circular mask that ensures that the values outside the sky hemisphere are exactly zero, corresponds to -1 in the transformed images.
Large value ranges can be processed differently as proposed by [YGH*21] who follow a divide-and-conquer approach to HDR skies by splitting the responsibilities into multiple networks each specialized on specific spatial parts (sun and sky) and thus different value ranges. While it simplifies the training task for individual networks, it also requires a way of merging the results together – typically through another HDR network.

4.4. Training Procedure

We are training our network according to standard StyleGAN 3 procedures for the translationally invariant configuration (StyleGAN3-T) with the resolution 1024 × 1024. We are using a single GPU for training which renders training times in the order of weeks. On top of the discriminator loss for the real images, we have an autoencoder loss in place that supervises the generation of clear sky images. This is a simple L_2 loss between the input image and the reconstructed image from the generator. We weight this loss with a factor of 10^4 higher in order to level it with the magnitude of the discriminator loss.

The circular mask that is applied on top of generated images is really important for the trajectory of a training run. Without a mask, the generator will spend a lot of effort on reproducing the sharp circular boundaries and the horizon, while any sky pattern in the center will have to be a byproduct. Much of the frequency budget in each layer (especially the higher frequencies) is spent to produce the image boundary so that less is available to produce intricate cloud patterns. With a circular mask, any accidental image content on the outside will be ignored by the discriminator thus freeing resources of the generator.

The final quality of the generated images in an adversarial network depends on the training progress of the discriminator. When training from a random initialization most training time is spent bringing the discriminator to a point where it can judge high-quality imagery. One can however benefit from transfer learning and start from network weights that were initially trained for a different (possibly unrelated) dataset. Then, the early discriminator layers already contain good image feature detectors which otherwise have to be learned.

4.5. Rendering

In order to prepare the generator output for the usage in a rendering system, the cloudy images have to be converted into a different projection. Most rendering systems use the equirectangular spherical projection (latitude-longitude) as the input format. Our images are encoded in the stereographic hemispherical projection and are thus only covering the top hemisphere. We re-project the image data with bilinear interpolation into equirectangular images of size 2048 × 1024 using `PTGui` before using them as IBL in `Blender Cycles` to light some scenes.

5. Results

For the results, we trained three variants of our network:

- $\mathcal{B}$ A $\mathcal{B}$aseline which is trained from scratch without our encoder and clear sky reconstruction loss. This corresponds to a standard StyleGAN3-T with support for HDR values.
- $\mathcal{O}$ A version with all $\mathcal{O}$ur modifications enabled. This includes the encoded clear sky images and the reconstruction loss.
- $\mathcal{F}$ A version similar to the above except that it was transfer learned from a pre-trained network that generates human $\mathcal{F}$aces.

Figure 4 depicts a matrix of images for these three training runs. The figure can be read row-wise, from left to right which represents the flow of data through our pipeline. The input clear sky (a) is generated from a picked sun position (part of the dataset) and then fed to the encoder of the applicable networks ($\mathcal{O}$ and $\mathcal{F}$). Mixed together with random noise, the encoded images are fed to the generator who produces two output images: the reconstructed clear sky in column two (b) and the cloudy sky in column three (c). The reconstruction column also has insets showing the signed difference towards the input clear sky with an exposure amplification of 2^2. In the last two columns (d,e) we show examplar renderings of an outdoor scene lit by these cloudy skies. The scene contains a mirror ball and a 100% Lambertian reflective `3DBenchy` boat. The last column (e) additionally adds an explicit sun light source in the renderer that produces a physically correct irradiance. We refer the reader also to the supplementary video, where an animated version of this figure is shown. The animations are linear interpolations in latent space for four different random vectors and a static clear sky input.

Reconstructed Clear Skies The three networks show different behaviour when looking at the reconstructed clear skies in Figure 4b. As expected, a network without any conditioning for clear skies ($\mathcal{B}$) produces arbitrary images in this output slot. The two networks with encoder ($\mathcal{O}$ and $\mathcal{F}$) that have been trained to also output clear skies do reconstruct a meaningful image. The sun position matches in a side-by-side comparison, however the difference image reveals a slight shift in position. This also shows in the animated videos with a temporal instability and the sun position moves around. The clear sky image in $\mathcal{O}$ is brighter than the reference, but seems more stable than $\mathcal{F}$.

Cloudy Skies When viewed from farther away, all three networks produce shapes and patterns that compare with the clouds and horizons in the training dataset. On closer inspection of Figure 4c, the results are however far from photorealistic and look more like clouds from a cartoon.

The clouds in the $\mathcal{B}$aseline show smooth gradients albeit also some visible halo artifacts with sharper corners are visible. The illumination has realistic tendencies with bright scattering clouds in front of the sun and absorbing shadows in the thicker clouds

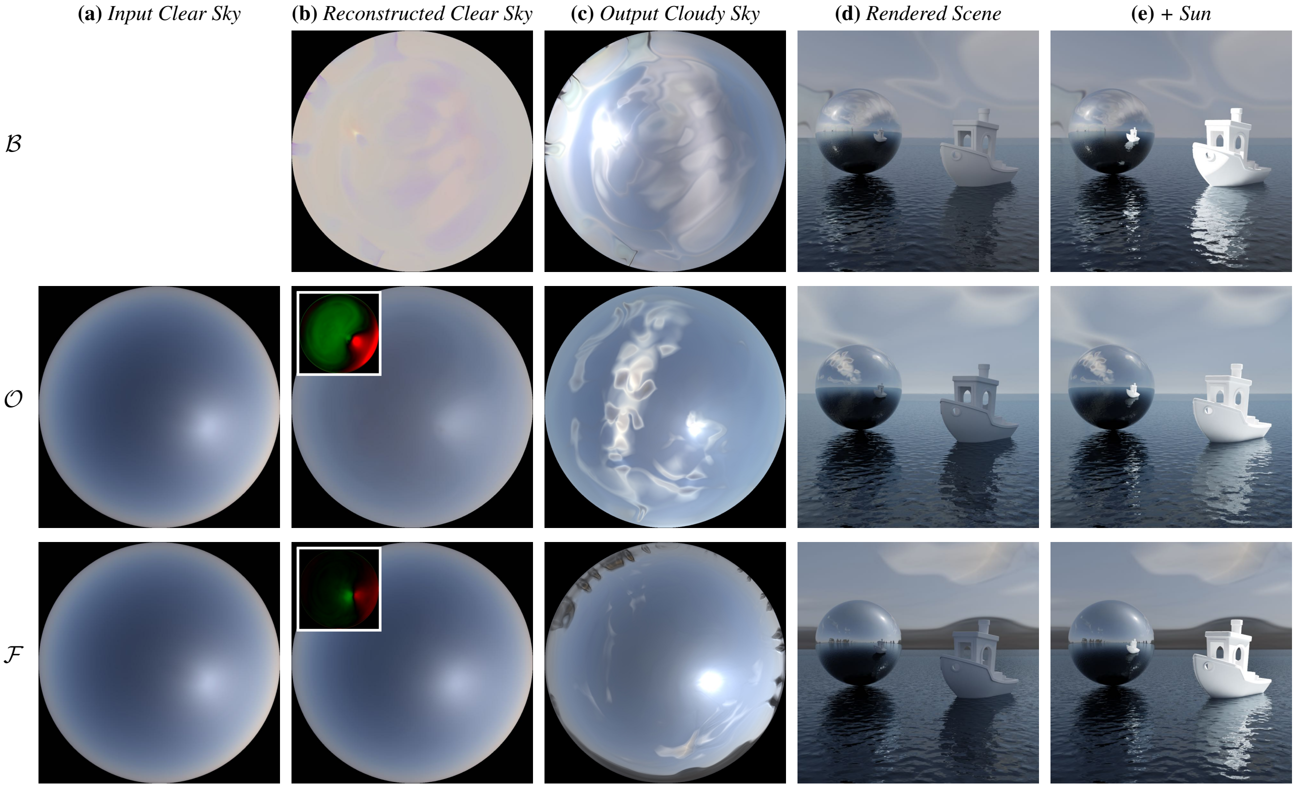

Figure 4: *Three training runs of equal length (1.1 million images): A $\mathcal{B}$aseline (top row), $\mathcal{O}$ur network with encoder and reconstruction loss (middle row), and a version of the latter that was transfer learned from $\mathcal{F}$aces (bottom row). The encoder conditions a roughly matching sun position (a-c) with minor shifts being visible in the difference image (b). The generated sun's brightness is however not enough (d), requiring the addition of an artificial sun light source (e). An animated version of this figure is shown in the supplemental video.*

away from the sun. The antennas on the horizon become part of the sky pattern and directly flow into each other or into a cloud. Note the arbitrary sun position that is not directly controllable. Instead, the network picks a sun position from the random latent vector.

$\mathcal{O}$ur network exhibits rather sharp cloud boundaries and strong halo artifacts that produce a spotty pattern. The cloud textures are less nuanced and look more like images from an earlier point in training time of $\mathcal{B}$. This hints at slower convergence. The sun position does coincide with the input clear sky and the reconstructed clear sky, despite this not being directly enforced by the loss.

The network transfer-learned from human $\mathcal{F}$aces shows a richer horizon that stays disconnected from the sky pattern for the most part. The clouds have a distinct thin appearance, with visible lines crossing trough at random. Halo artifacts are not visible, but the clouds occasionally show a discoloration that has nothing to do with the illumination. The sun position is again correctly matched to the desired input in this example, but for some latent vectors it produces two suns that seem to be mirrored around some non-static axis. This training on the one hand benefits from the pre-trained discriminator, but also suffers from some deeply rooted concepts of the generator such as skin color and face symmetry. We observe that these artifacts are vanishing over training time.

Rendered Scenes When used as-is in a renderer to illuminate the scene (Figure 4d), the HDR cloudy skies look fairly convincing in glossy reflections. A direct observation in the background is however showing the absence of detail. The scenes look rather dark and are lacking contrast which can be explained by the clipped sun values in the training dataset. This motivates the last column (e), where we manually add a sun light source to compensate for the missing energy. There, the boat actually appears in the intended white color.

Controlling the cloud coverage We demonstrate in Figure 5 how our method can also produce images with varying cloud coverage

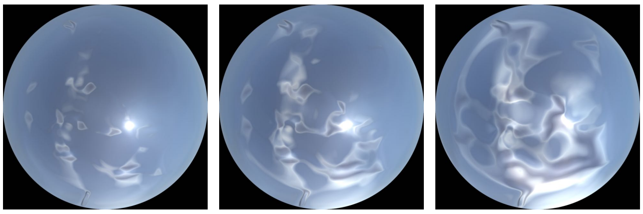

Figure 5: *The cloud coverage can be globally controlled by scaling the magnitude of the random latent vector.*

while keeping the surrounding and sun position constant. This relies on limiting the deviation of the random latent vector from the centroid in latent space and is referred to as the "Truncation Trick" in GAN literature. In our networks, the origin of the latent space produces a clear sky while cloudy images tend to be located farther from the origin.

Quantitative Evaluation Measuring the quality of a generative model is commonly using the Fréchet inception distance (FID), a metric comparing the statistical distribution between a number of generated images and the training dataset. For each datapoint in Figure 6, we generate 5000 images (matching the dataset size) and compute the FID score. The metric is based on a neural network that was pre-trained on ImageNet and thus only supports LDR input. Since our network produces HDR outputs, we clipped the values, disregarding the sun brightness in the computation.

Our $\mathcal{B}$aseline experiment achieved a minimum FID 89.1 in the training time of 1.1 million images. In comparison, Karras et al. [KAL*21] report values of 5 and below for their networks trained on faces. $\mathcal{O}$urs achieved minimum FID of 95.5. While this score is worse, the inclusion of Encoder gives control over the sun position. The transfer-learned $\mathcal{F}$aces experiment reached FID of 76.5, consistently outperforming the other runs. This is expected as the discriminator has a significant head-start in judging finer details which challenges the generator more. The graphs confirm our hypothesis of slower convergence of $\mathcal{O}$ that we have drawn from visual inspection above.

6. Discussion and Future Work

In the previous section, we saw that our method is capable of generating skydomes for given sun positions and cloud coverage ratios, and it can be used in HDR rendering. We now discuss the specifics and limitations of our approach, and how they could be addressed.

Overcast skies and sun position detection We found that training our network works best when the clear sky images and real photographs are matched and synchronized during training, challenging the discriminator even more. This requires detecting the sun position in the photographs such that matching clear sky images can be generated. Unfortunately, automatically detecting the sun position is a challenging task in the presence of occlusion by a thick cloud, or when the sky is completely overcast. Hence for the current experiments, we only used a subset of our dataset with a directly visible or easily detectable sun position, which means that the network cannot generate any overcast skies, or skies with thick clouds obscuring the sun yet.

In the future, a more robust sun position detection technique should be used, e.g., applying temporal constrains to fit a parabola to the sun trajectory in consecutive detected frames. In case the sun disk is hidden behind a cloud in some of the frames, its position can be reliably interpolated from the known positions.

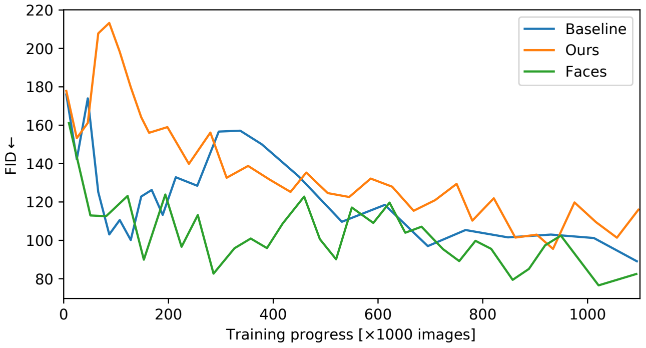

Figure 6: *Convergence plot in terms of the FID metric where lower is better.*

Direct sun radiance The radiance (light energy) coming from the sky can be split to indirect radiance, which arrives from the Sun and is scattered in the atmosphere or clouds, and direct radiance, which is coming directly from the sun disk and is only weakened by the transmission through the atmosphere. While the indirect energy is rather accurately represented by the clear sky model and the dataset photographs, the direct energy is very high and concentrated in a tiny sun disk. This results in two major difficulties. First, since the signal has a very high-frequency and value, it is not trivial for a generative network to accurately generate a sun disk with the correct energy. Second, the energy is so intense that it is impossible to be captured unclipped without using a neutral density (ND) filter in the camera. In this paper, we solved this issue by manually adding the extra solar energy on top of the skydome in the renderings, which is common practice for clipped HDRIs. That makes the final renders look less dull and better represent the actual illumination. In the future, a more robust solution would be to capture a dataset with an ND filter and adaptive exposure times [SJW*06], and possibly generate the sun disk in a separate dedicated network layer. Extra care would have to be taken in case a sun disk is only partially occluded by a cloud.

Stereographic projection distortion The stereographic projection used in our method results in the directions towards the zenith having the highest resolution, while the horizon has the lowest resolution. This is beneficial for learning the details of the cloud shapes above the observer, but is not ideal for rendering flat areas such as sea or oceans that do not have any objects on the horizon covering the low resolution of the generated skydome's horizon. Our early experiments show that the network could be trained with different projections without such prominent distortions, and it could result in fewer problems in the training, so it is an important future research direction.

Convergence It should also be noted that generative adversarial networks (GANs) typically need many iterations to converge. While [KAL*21] trained their network for so long that it saw as many as 25 million training images in total, we only managed to train on 1.1 million in 9 days. From our experiments on lower resolutions, we know that longer trainings result in higher details in the textures, as the network is slowly learning the high-frequency data. Figure 7 confirms this also for the higher resolution when compared to Figure 4. Training our network on an expensive GPU cluster for a long time would likely result in more realistic looking clouds with a less "cartoon-like" appearance.

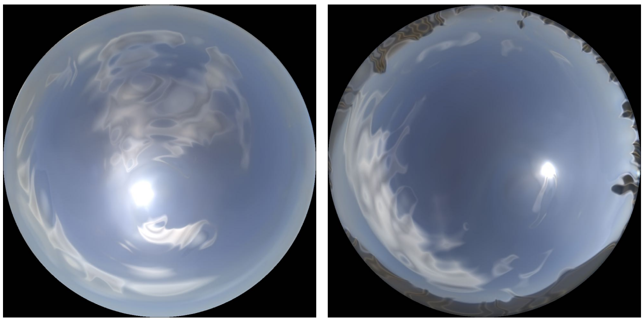

Figure 7: *Results of a more converged training ($\mathcal{O}$ after 2.3 million images) that promises higher fidelity is possible with our method.*

High-resolution tiling The convergence is also related to the actual skydome resolution. While our network was trained on 1024 × 1024 pixels, training on even higher resolutions would benefit the high-resolution renders, but it would be very hard to achieve with a limited GPU memory and time. We believe that tiling several lower-resolution subsets of the skydome could be a solution.

User parameters Our current method allows the users to select a sun position and a cloud coverage ratio. We believe that many additional parameters could be added to allow a finer selection of the generated skydomes, which would be perfect for artists to match the mood of their renderings. For example, we could expose all existing parameters of the clear sky model [WVB*21] such as visibility distance or elevation. We could also condition the generator on outputting certain types of clouds, e.g., thin high-altitude clouds vs. dense thunderstorm clouds. We will also inspect the generator to find latent dimensions that are responsible for certain cloud positions and types, in order to enable spatial control of where exactly the clouds should appear, a concept similar to GauGAN [PLWZ19].

7. Conclusion

We have shown a GAN that produces cloudy skies in stereographic projection. While the goal of photorealism has not been reached due to limited resources, the proposed clear sky encoder approach does help to directly parametrize the sun position. Our network is able to produce the high dynamic range required in rendering, but we see a limitation in our dataset that prohibits our environment maps to light scenes realistically on their own.

With this paper, we are paving the way for a hybrid between analytical and data driven solutions for image-based lighting. Through the combination of parameterizable synthetic and diverse real data, our method leverages the strengths of both classical analytical models and modern data-driven approaches.

Acknowledgments

We thank the reviewers for their extensive feedback. This work was supported by the Grant Schemes at CU, reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935 and the Charles University grant SVV-260588 and by the Grantová Agentura České Republiky with grant number GACR-22-22875S. This work was supported through the Intel oneAPI Center of Excellence at Charles University.

References

[ATO*19] Andrianakos, George, Tsourounis, Dimitrios, Oikonomou, Spiros, et al. "Sky Image forecasting with Generative Adversarial Networks for cloud coverage prediction". *10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, July 1, 2019. doi: 10.1109/IISA.2019.8900774 3.

[BN08] Bruneton, Eric and Neyret, Fabrice. "Precomputed Atmospheric Scattering". *Computer Graphics Forum*. Vol. 27. Issue: 4. Wiley Online Library, 2008, 1079–1086 2.

[Bru16] Bruneton, Eric. "A qualitative and quantitative evaluation of 8 clear sky models". *IEEE Transactions on Visualization and Computer Graphics* 23.12 (2016), 2641–2655 2.

[CZR22] Chalmers, Andrew, Zickler, Todd, and Rhee, Taehyun. "Illumination Browser: An intuitive representation for radiance map databases". *Computers & Graphics* 103 (Apr. 1, 2022), 101–108. doi: 10.1016/j.cag.2022.01.006. url: https://www.sciencedirect.com/science/article/pii/S0097849322000061 2.

[Deb98] Debevec, Paul. "Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography". *ACM Transactions on Graphics*. SIGGRAPH '98. New York, NY, USA: Association for Computing Machinery, July 24, 1998, 189–198. doi: 10.1145/280814.280864 1.

[DWX*20] Ding, Xin, Wang, Yongwei, Xu, Zuheng, et al. "CcGAN: Continuous Conditional Generative Adversarial Networks for Image Generation". Sept. 2020. url: https://openreview.net/forum?id=PrzjugOsDeE 3.

[EBK*16] Emde, C., Buras-Schnell, R., Kylling, A., et al. "The libRadtran software package for radiative transfer calculations (version 2.0.1)". *Geoscientific Model Development* 9.5 (2016), 1647–1672. doi: 10.5194/gmd-9-1647-2016. url: http://www.geosci-model-dev.net/9/1647/2016/ 2.

[EGH21] Einabadi, Farshad, Guillemaut, Jean-Yves, and Hilton, Adrian. "Deep Neural Models for Illumination Estimation and Relighting: A Survey". *Computer Graphics Forum* 40.6 (2021), 315–331. doi: 10.1111/cgf.14283 2.

[EKD*17] Eilertsen, Gabriel, Kronander, Joel, Denes, Gyorgy, et al. "HDR image reconstruction from a single exposure using deep CNNs". *ACM Transactions on Graphics* 36.6 (Nov. 2017), 178:1–178:15. doi: 10.1145/3130800.3130816 6.

[GGJ18] Guimera, David, Gutierrez, Diego, and Jarabo, Adrián. "A Physically-Based Spatio-Temporal Sky Model". *Spanish Computer Graphics Conference (CEIG)*. Ed. by García-Fernández, Ignacio and Ureña, Carlos. The Eurographics Association, 2018. doi: 10.2312/ceig.20181150 2.

[GPM*14] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, et al. "Generative Adversarial Networks". *Advances in Neural Information Processing Systems*. Vol. 27. arXiv: 1406.2661. Curran Associates, Inc., 2014. url: http://arxiv.org/abs/1406.2661 3.

[HAL19] Hold-Geoffroy, Yannick, Athawale, Akshaya, and Lalonde, Jean-François. "Deep Sky Modeling for Single Image Outdoor Lighting Estimation". *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019, 6920–6928. doi: 10.1109/CVPR.2019.00709 3.

[HMP*20] Hädrich, Torsten, Makowski, Miłosz, Pałubicki, Wojtek, et al. "Stormscapes: simulating cloud dynamics in the now". *ACM Transactions on Graphics* 39.6 (Nov. 26, 2020), 175:1–175:16. doi: 10.1145/3414685.3417801 2.

[HMS05] Haber, Jörg, Magnor, Marcus, and Seidel, Hans-Peter. "Physically-based simulation of twilight phenomena". *ACM Transactions on Graphics* 24.4 (Oct. 2005). Place: New York, NY, USA Publisher: ACM, 1353–1373. DOI: 10.1145/1095878.1095884 2.

[Hoj19] Hojdar, Štěpán. "Using neural networks to generate realistic skies". MA thesis. Charles University, Faculty of Mathematics and Physics, 2019 3.

[HSH*17] Hold-Geoffroy, Yannick, Sunkavalli, Kalyan, Hadap, Sunil, et al. "Deep outdoor illumination estimation". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, 2373–2382. DOI: 10.1109/CVPR.2017.255 2, 5.

[HW12] Hošek, Lukas and Wilkie, Alexander. "An analytic model for full spectral sky-dome radiance." *ACM Transactions on Graphics* 31.4 (2012), 95:1–95:9. URL: http://dblp.uni-trier.de/db/journals/tog/tog31.html#HosekW12 1, 2.

[HW13] Hošek, Lukas and Wilkie, Alexander. "Adding a Solar-Radiance Function to the Hošek-Wilkie Skylight Model." *IEEE Computer Graphics and Applications* 33.3 (2013), 44–52. URL: http://dblp.uni-trier.de/db/journals/cga/cga33.html#HosekW13 1, 2.

[KAH*20] Karras, Tero, Aittala, Miika, Hellsten, Janne, et al. "Training Generative Adversarial Networks with Limited Data". (Oct. 2020). arXiv: 2006.06676. URL: http://arxiv.org/abs/2006.06676 3, 5.

[KAL*21] Karras, Tero, Aittala, Miika, Laine, Samuli, et al. "Alias-Free Generative Adversarial Networks". *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, 852–863. URL: http://arxiv.org/abs/2106.12423 2, 3, 5, 8.

[KMM*17] Kallweit, Simon, Müller, Thomas, Mcwilliams, Brian, et al. "Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks". en. *ACM Transactions on Graphics* 36.6 (Nov. 2017), 231:1–231:11. URL: https://doi.org/10.1145/3130800.3130880 2.

[LM14] Lalonde, Jean-François and Matthews, Iain. "Lighting Estimation in Outdoor Image Collections". *2014 2nd International Conference on 3D Vision*. Vol. 1. Dec. 2014, 131–138. DOI: 10.1109/3DV.2014.112 2, 3.

[MCS21] M. Jain, C. Meegan, and S. Dev. "Using Gans to Augment Data for Cloud Image Segmentation Task". *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. July 11, 2021, 3452–3455. DOI: 10.1109/IGARSS47720.2021.9554993 3.

[MO14] Mirza, Mehdi and Osindero, Simon. "Conditional Generative Adversarial Nets". (Nov. 2014). arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784 3.

[PLWZ19] Park, Taesung, Liu, Ming-Yu, Wang, Ting-Chun, and Zhu, Jun-Yan. "Semantic Image Synthesis with Spatially-Adaptive Normalization". (Nov. 2019). URL: http://arxiv.org/abs/1903.07291 9.

[PSS99] Preetham, A. J., Shirley, Peter, and Smits, Brian. "A practical analytic model for daylight". *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM SIGGRAPH. 1999, 91–100. DOI: 10.1145/311535.311545 2.

[SJW*06] Stumpfel, Jessi, Jones, Andrew, Wenger, Andreas, et al. "Direct HDR capture of the sun and sky". *ACM SIGGRAPH 2006 Courses*. July 2006. DOI: 10.1145/1185657.1185687 4, 8.

[SK21] Somanath, Gowri and Kurz, Daniel. "HDR Environment Map Estimation for Real-Time Augmented Reality". *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, 11293–11301. DOI: 10.1109/CVPR46437.2021.01114 3.

[SMDB22] Satilmis, Pinar, Marnerides, Demetris, Debattista, Kurt, and Bashford-Rogers, Thomas. "Deep Synthesis of Cloud Lighting". *IEEE Computer Graphics and Applications* (2022). DOI: 10.1109/MCG.2022.3172846 3.

[Špa20] Špaček, Jan. "Generation of realistic skydome images". MA thesis. Charles University, Faculty of Mathematics and Physics, 2020 3.

[TYS09] Tao, Litian, Yuan, Lu, and Sun, Jian. "SkyFinder: attribute-based sky image search". *ACM Transactions on Graphics* 28.3 (July 27, 2009), 68:1–68:5. DOI: 10.1145/1531326.1531374 2.

[WVB*21] Wilkie, Alexander, Vevoda, Petr, Bashford-Rogers, Thomas, et al. "A fitted radiance and attenuation model for realistic atmospheres". *ACM Transactions on Graphics* 40.4 (July 19, 2021), 135:1–135:14. DOI: 10.1145/3450626.3459758 1, 2, 4, 5, 9.

[YGH*21] Yu, Piaopiao, Guo, Jie, Huang, Fan, et al. "Hierarchical Disentangled Representation Learning for Outdoor Illumination Estimation and Editing". *IEEE International Conference on Computer Vision (ICCV)*. Oct. 2021, 15293–15302. DOI: 10.1109/ICCV48922.2021.01503 6.

[YME*20] Yu, Ye, Meka, Abhimitra, Elgharib, Mohamed, et al. "Self-supervised Outdoor Scene Relighting". *European Conference on Computer Vision (ECCV)*. Aug. 2020, 84–101. DOI: 10.1007/978-3-030-58542-6_6 3.

[ZPIE17] Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". *IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, 2242–2251. DOI: 10.1109/ICCV.2017.244 3.

[ZSH*19] Zhang, Jinsong, Sunkavalli, Kalyan, Hold-Geoffroy, Yannick, et al. "All-Weather Deep Outdoor Lighting Estimation". *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019, 10150–10158. DOI: 10.1109/CVPR.2019.01040 3.