DOI: 10.1111/cgf.14990

COMPUTER GRAPHICS *forum*
Volume 0 (2023), number 0, e14990

SkyGAN: Realistic Cloud Imagery for Image-based Lighting

Martin Mirbauer,[1,*] Tobias Rittig,[1,*] Tomáš Iser,[1] Jaroslav Křivánek[1,2] and Elena Šikudová[1]

[1]Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
{martinm, tobias, tomas, krivanek, sikudova}@cgg.mff.cuni.cz
[2]Chaos Czech a.s., Prague, Czech Republic

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 fisheye photographs of clouds. We demonstrate our results on renders of outdoor scenes under varying time, date and cloud covers. Our source code and a dataset of 39 000 HDR sky images are publicly available at https://github.com/CGGMFF/SkyGAN.

Keywords: modelling; natural phenomena, rendering; image-based rendering, rendering; atmospheric effects

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 HW [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, 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 are required for high-quality results. At the same time, the images are static, meaning they are 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 is 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 on an analytical sky model and a dataset of photographs. 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. Then it feeds the image into an encoder-generator network, which generates a corresponding cloudy image. This 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.

*Both the authors contributed equally.

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 force the generator 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].

This paper serves as a stepping stone towards reaching the ultimate goal: a fully automated model which generates a corresponding realistic and physically accurate HDR skydome in high resolution for a given artist's input (parameters). While our results presented here are not production-ready yet, we support the progress by publishing our source code, including the data-processing tools, and the processed dataset.

Our contributions include the following:

- A directly parameterizable cloudy sky model based on a conditional GAN architecture (Section 4)
- A dataset of 39 000 HDR sky photographs in 30-s intervals (4.1)
- An automatic processing pipeline for HDR, panoramic time-lapse photographs (4.1)
- A method for fitting the [WVB*21] clear sky model to real photographs (4.2)
- Support for HDR images in the StyleGAN3 codebase (4.4)
- An evaluation of a dataset augmentation method (azimuth marginalization) for training with more cloud covers per sun position (4.5)

We have shown an earlier version of this paper at a conference [MRI*22]. Since then, we have significantly enlarged the available dataset size and improved the overall visual quality.

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 the high-quality rendering of outdoor scenes, accurate skydome illumination and sky colours can be achieved using an atmospheric model. One can perform accurate brute-force Monte Carlo simulations based on first principles, evaluating light transport in atmospheric gases. Or one can use faster analytical models that can be directly evaluated with 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 direct use in computer graphics. Methods more suited for image rendering, such as [HMS05, BN08, GGJ18], 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 lengthy process, and moreover, the models do not directly support the rendering of clouds and overcast skies. For that, one would render fully volumetric clouds from a simulation [HMP*20, HHP*21] 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 very high performance. Unlike our method, they are limited to clear, cloudless skies. They are usually based on fitting some 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 also match overcast skies.

The most recent is the Prague Sky Model [WVB*21], which is based on tensor decomposition and supports different observer altitudes, post-sunset conditions, in-scattered radiance and attenuation for finite distances, and polarization. Using the provided transmission function, one can accurately add the solar disk including effects like the red sun at *golden hour*. Subsequent work extends the model to a wider spectral range from ultraviolet (UV) to near-infrared (NIR) wavelengths [VBKW22]. We use the unpolarized ground-level version of this model to produce the matching synthetic clear sky images for our real photographs.

An approach used in production to add clouds to a clear sky uses procedural or pre-computed volumetric density fields. During rendering, these are evaluated using ray marching which is computationally more demanding than simple image-based lighting. The diversity of cloud types is also limited to handcrafted models.

2.2. Machine learning for IBL

In Computer Vision, there are many methods for generating environment maps using deep-learning tools, especially for lighting estimation. There, the posed problem consists of estimating the spherical scene illumination from the narrow field-of-view images, which, in turn, can be used for rendering a virtual object into the scene with plausible shadows, reflections, and colours. 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) for fitting 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. Both methods rely on analytical clear sky models, so the output imagery contains no 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. Simultaneously 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 scene observation. 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*21] 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 long-standing problem of textures 'sticking' to the underlying pixel grid. They redesigned 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 with known properties. Although our dataset also consists of sequential images, we do not exploit the time dependency yet. [SB23] predicts the clouds movement from a single input image. 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 [JMD21].

We instead 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]. The authors use a U-Net encoder-decoder 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. On the other hand, it 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.

Inspired by our conference paper [MRI*22], another U-Net-like network [VMM*23] generates clouds on top of a clear sky image without direct supervision. It uses a patch-based adversarial loss, a perceptual loss comparing generated to a real cloudy image, and a loss that simulates illumination in a simple scene to ensure the correct sun brightness and shadow sharpness. Combined with their unclipped HDR dataset [LAB*16], this loss enables a physically correct sun brightness directly out of the network. Constrained by a clipped HDR dataset, our images require an additional sun light source placed by the artist. Compared to this method, we use a more recent sky model [WVB*21], a conditional GAN architecture based on StyleGAN3 [KAL*21] (potentially avoiding texture sticking/aliasing, and the adversarial loss is not patch-based), and use azimuth marginalization to aid our generation method.

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 users 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 2D basis functions 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 tensor that influences the basis functions. In contrast, the discriminator architecture strictly works on a pixel grid – thus exhibiting any problems that may come with this approach.

Second, the generator spends parts of its capacity learning textures and parts to generate grid-like coordinate systems carrying 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.

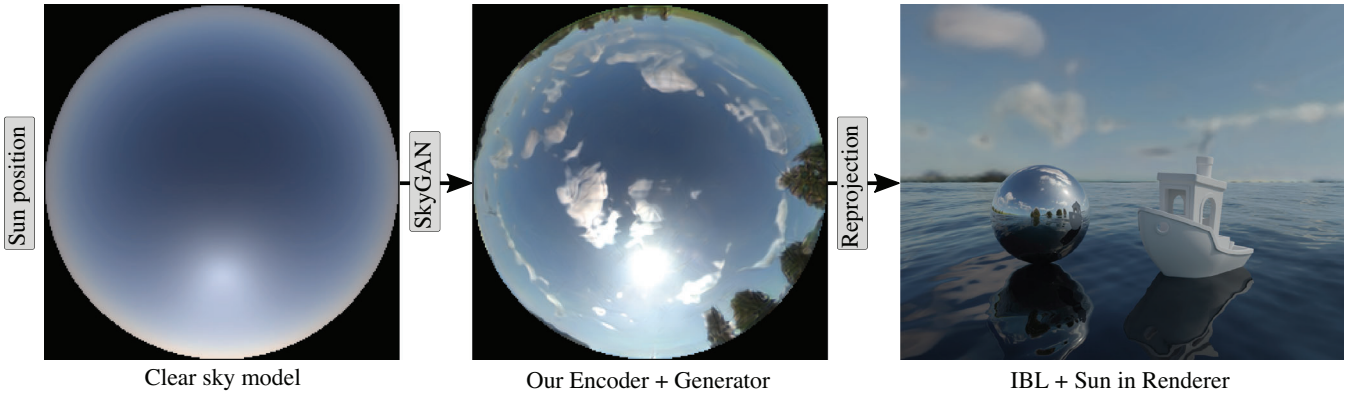


Figure 1: *Our network is trained on matching photos and fitted clear sky images. The clear skies are encoded into the generator's latent space w and concatenated with (mapped) random values z. The generator reconstructs a matching pair of the clear and cloudy sky from this vector. The quality of the clear sky is judged against the input clear sky, while the generated cloudy sky is evaluated by the discriminator for its realism.*

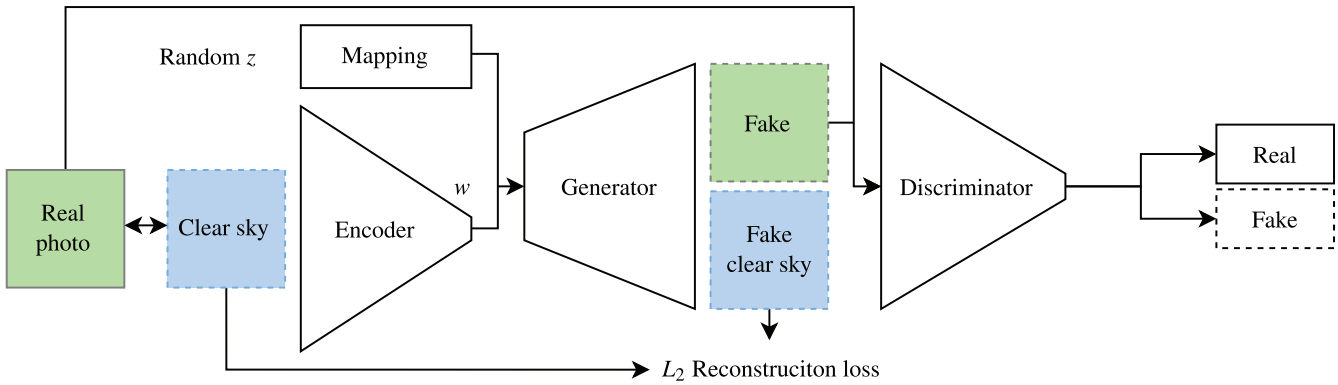


Figure 2: *Our method generates cloudy sky images from a user-chosen sun position that are readily usable as an environment map in any rendering system. We leverage an existing clear sky model to produce the input to our neural network which enhances the sky with clouds, haze and horizons learned from real photographs.*

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 artefact 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.

4. Method

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

We show our network architecture in Figure 1. We feed a clear sky image into an encoder at training time to compress its information to a few numbers. Concatenated with a mapped random vector z, it forms the latent input w to a generator 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 parallel to the generator and discerns between real and fake imagery, effectively challenging the generator to produce gradually more realistic-looking cloudy skies.

As shown in Figure 2, the user can input the desired sun position and get a corresponding cloudy sky image during inference. One can explore different cloud constellations by adjusting the seed for the random generator z. 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 latent vector w of the generator. Finally, the cloudy image from the

generator's output can be re-projected to an equirectangular projection and used in a standard rendering pipeline as an environment map. When the images are not only used as a background but provide illumination for the scene, high dynamic range becomes increasingly important. With clipped values, the renderings will exhibit reduced contrast, making 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. The processed dataset, including the fitted metadata, and all source code, including the processing pipeline and networks, are publicly available [MRI*23].

4.1. Dataset

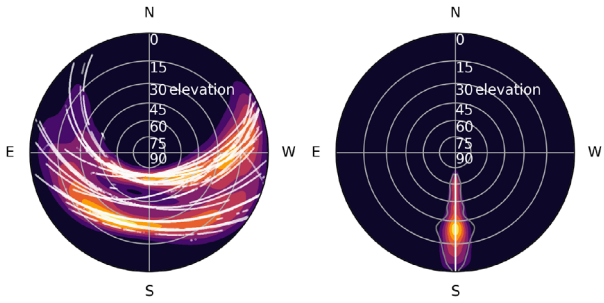
The real photographs used during the network training are based on a novel dataset of HDR skydome photos. The photos were captured on a full-frame camera sensor by aiming an 8 mm circular fisheye lens upward towards the sky. The dataset is based on several locations, mainly in Central Europe and coastal 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 (time-lapse), one photo per 15–120 s (mostly 30 s), sometimes over a long time period from sunrise to sunset. We captured around 39 000 HDR images from 62 different days at 16 locations distributed primarily over 6 months (May–November) with sporadic sessions in winter. Our training used 35 425 images.

Each hemispherical HDR photo was developed from an exposure stack of five to nine exposures. But even with the shortest possible exposure time, the brightness of the sun and its surroundings is clipped on direct observation without the use of a neutral density (ND) filter [SJW*06]. While mounting an ND filter on a hemispherical fisheye is technically possible behind the lens, it would require lens distortion re-calibration and also prolong the exposure times on darker images where the clouds are visible. This would lead to significant motion blur with moderate wind speeds and ghosting artefacts in the stacked HDR image.

Processing photographs. We have developed a semi-automated pipeline for processing of RAW image files into brightness, colour, and geometry-calibrated OpenEXR images. The pipeline performs demosaicing, HDR merging, and panorama re-projection based on each image's EXIF data, using `RawTherapee` [HR05] for RAW processing and `PTGui Pro` [New01] for operations on spherical panoramas. We refer the reader to the supplemental material for details about the training dataset and the processing algorithm.

4.2. 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. Optionally, one can also query the transmission function for finite (e.g. mountains) and infinite (e.g. sun) distances to weight any direct irradiance



(a) *Unmodified* **(b)** *Marginalised azimuth*

Figure 3: *The distribution of sun positions in our dataset, plotted in stereographic projection, similarly to our images. The right plot shows the data augmention we apply (see Section 4.5).*

coming from a direction. 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 mostly contains sun positions above the horizon, this information is enough 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 lighting.

Sun position. For each captured image in our dataset, we detect the sun position by fitting an ellipse to the sun disk. Thanks to the brightness consistency in our dataset, we can use adaptive thresholding with a fixed minimum brightness. We are filling in the gaps where the sun detection failed (e.g. due to occlusion by clouds or the horizon) using a fitted trajectory of the sun's movement over time. The python package `pysolar` [SP07] provides an expected curve for sun positions from location and capture times. This expected trajectory is fitted against the observed sun positions by minimizing the deviation over a time shift of ± 20 min. Thus any calibration errors in camera time setting, tripod levelling, and north orientation are revealed. In the future, we would like to inform the re-projection step of our data pipeline to also compensate for the geometric errors in a subsequent run. Figure 3 displays the density of sun positions over the hemisphere.

Visibility and ground albedo. The remaining two parameters of the clear sky model can be fitted from a clear sky photo using the L-BFGS-B algorithm with finite-difference gradients. After manually masking non-clear-sky objects like the horizon, clouds, the sun disk, lens flares and dust on the lens, the optimization is a three-step procedure. First, the exposure is roughly matched to get the values in the same order of magnitude. Then the optimizer searches for the model parameters constrained by their range, and finally, we jointly fine-tune both the parameters and the exposure. Related fitting procedures are described in [HSH*17, Section 4.2] for the HW sky model and in [VMM*23, Section 3.1] for the LM model. We show the results of this procedure in Figure 4. Because this step involves a manual input, we have not yet performed this fitting for every day in our dataset. As described in the following subsection, the clear sky input acts only as a guiding channel during training and is not directly visible in the final output. This allows us to stay with a fixed visibility and ground albedo despite a slight mismatch

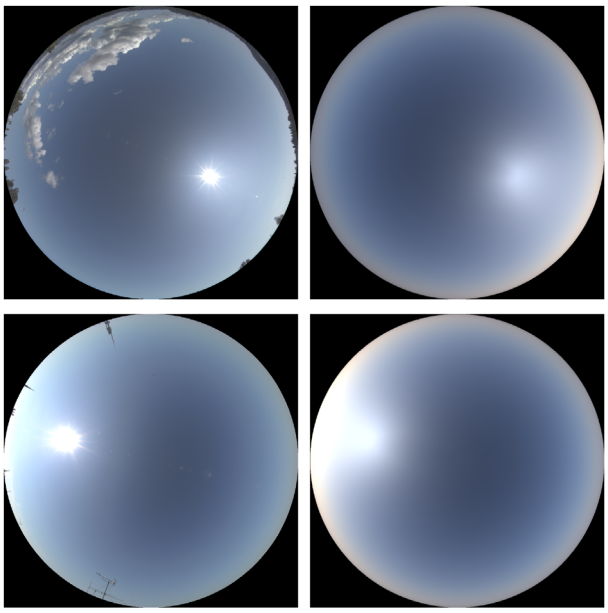


Figure 4: *Two example photographs from our dataset (left) with their fitted corresponding clear skies (right) – all in stereographic projection. The* `visibility distance` *mostly affects the aura around the sun, while the* `ground albedo` *brightens towards the horizon. Our clear sky images only depict the atmospheric scattering component, neglecting the attenuated sun disk and rays caused by aperture on the photograph.*

in atmospheric conditions between real photos and the corresponding clear skies.

4.3. Architecture

On a high level, the architecture, as seen in Figure 1, 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 to support 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) as the discriminator does, the encoder works 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 the model's four parameters. Ablation experiments have shown that a minimum of two values would be enough to reconstruct reasonably high-quality results. On top of the RGB clear sky image, the encoder is passed the angular distance from the sun in a separate channel with the intention of disambiguating post-sunset conditions. The encoded clear sky is injected into the network's latent space w after batch-normalization. An injection into the random vector z would unnecessarily correlate the sun position with the cloud distribution in the mapping network and thus lead to a less predictable user experience.

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 frequency-based, 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. Matching the real photographs, we mask the image outside the projected circle of the stereographic projection.

Discriminator. The discriminator remains unchanged from the StyleGAN3 (originally from StyleGAN2) codebase. We use 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. Our dataset uses a stereographic projection; therefore, we exclude augmentations related to translation and scaling, keeping only arbitrary rotation and colour transformations (brightness, contrast, luma flip, hue rotation and saturation).

4.4. 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.

4.5. Training procedure

We are training our network according to standard StyleGAN3 procedures for the translationally invariant configuration (StyleGAN3-T) with the resolution 256×256. When using eight GPUs for training, a full 30M images training runs in about 5 days.

A common data augmentation technique previously also used by [HSH*17] is to rotate all photographs upon loading such that the sun position lies on a common azimuth angle. This was inspired by a body of research from clear-sky models [PSM93, PSS99, HW12, WVB*21] where the radiance patterns are parameterized independently of azimuth. The sun thus only moves on the line from the zenith, in the centre of the image, to the horizon on the bottom. This marginalizes out the azimuth dimension of the dataset and increases the variety for each elevation angle (see Figure 3b). Effectively this augments the data for each sun position, as more distinct cloud formations with the same elevation angle are available. We call this *azimuth marginalization* and evaluate its performance in Section 5.

The circular mask applied on top of generated images is really important for the trajectory of a training run. Without a mask, the generator spends a lot of effort on reproducing the sharp circular boundaries and the horizon, while any sky pattern in the centre 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. Similarly, it would be helpful to mask foreground objects like trees on the horizon to prevent them from appearing on some generated outputs.

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. However, one can 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.

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.

4.6. Rendering

To prepare the generator output for 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. For azimuth-marginalized networks we first rotate the image to the actually desired azimuth. Then, 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 four 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.
- $\mathcal{B}_A$ The same network architecture as $\mathcal{B}$, but trained on azimuth-marginalized images (see Section 4.5).
- $\mathcal{B}_C$ The $\mathcal{B}$aseline network but including the autoencoder task for the clear sky (see Section 4.3).
- $\mathcal{O}$ A version with all $\mathcal{O}$ur modifications enabled. This includes the encoded clear sky images, the reconstruction loss ($\mathcal{B}_C$) and the training on azimuth-marginalized images ($\mathcal{B}_A$).

Figure 5 depicts a matrix of images for these four variants. 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 ($\mathcal{B}_C$ and $\mathcal{O}$). Concatenated to a random latent-space position w, the encoded images are fed to the generator that 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 2^5. 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-plausible irradiance.

In the supplementary material, we show the HDR images corresponding to Figure 5 as well as additional uncurated samples for each variant.

Reconstructed clear skies. The three networks show different behaviour when looking at the reconstructed clear skies in Figure 5b. As expected, a networks without any conditioning for clear skies ($\mathcal{B}$ and $\mathcal{B}_A$) produce arbitrary images in this output slot. The two networks with encoder and reconstruction loss ($\mathcal{O}$ and $\mathcal{B}_C$) do output a meaningful image. The sun position matches in a side-by-side comparison, however the amplified difference image reveals a slight shift in position. For $\mathcal{B}_C$ there is a shift in azimuth (rotation) visible that likely stems from a bias in sun position density of our training data (see Figure 3a). The error in $\mathcal{O}$ is slightly lower and is limited to the elevation axis (up/down).

Cloudy skies. When viewed from farther away, all four networks produce shapes and patterns that compare with the clouds and horizons in the training dataset. On closer inspection of Figure 5c, the

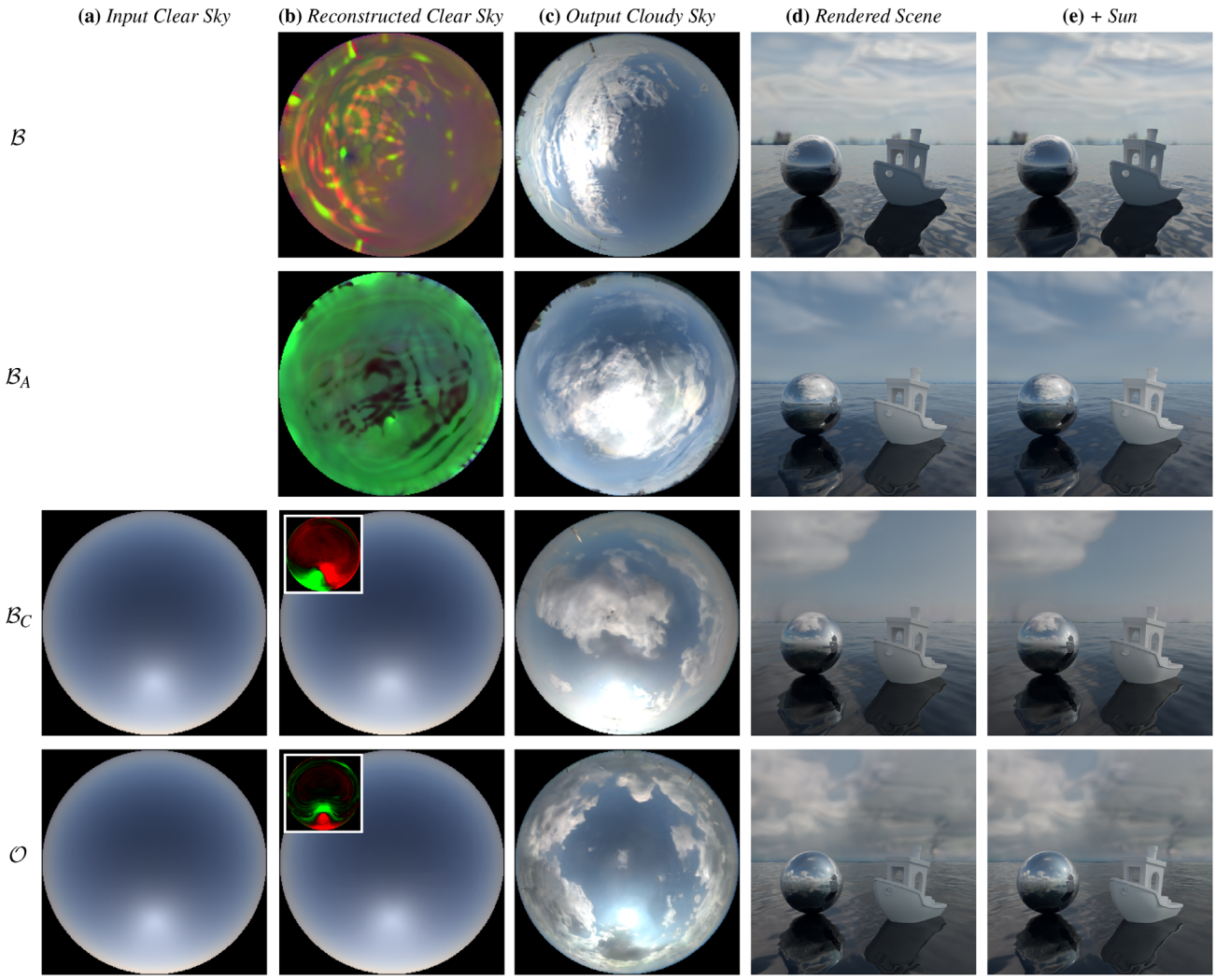


Figure 5: *Ablation study over four training runs: A Baseline (top row) and Our network (bottom row) with encoder and azimuth marginalization. In between, $\mathcal{B}_A$ isolates the azimuth marginalisation feature and $\mathcal{B}_C$ the clear sky encoding. The encoder conditions a roughly matching 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) whose brightness was artistically chosen to match the cloud cover.*

results are however clearly distinguishable from real photographs. Most noticeable are banding and halo artifacts around cloud edges. The horizon is slightly blurred or even half-transparent. Occasionally the horizon structures "flow" into the nearby clouds. These artifacts can be explained by the choice of GAN architecture. As the StyleGAN3 generator works in Fourier space, it reconstructs the images from a set of base frequencies. Starting with the low-frequency structure it adds higher frequencies over (training) time. Sharp cloud boundaries are thus harder to represent often resulting in under- and overshooting (halos) before and after the edges. Judging the visible artifacts (e.g. ringing around the horizon), $\mathcal{B}$ seems to be worst, whereas $\mathcal{O}$ has less apparent problems. $\mathcal{B}_A$ and $\mathcal{B}_C$ rank somewhere between them. This visual impression is also supported by the quantitative evaluation (Figure 7). The sun positions for the $\mathcal{B}$ and $\mathcal{B}_A$ networks are effectively random, as they were not conditioned for this parameter by the encoded clear sky. For the networks with the encoder, the sun position roughly coincides with the one in the clear sky. However, later on during training the sun starts to decouple and shift slightly. We see room for improvement here, with an explicit (adversarial) loss between the clear and cloudy sky to avoid such divergence.

Rendered scenes. When used as-is in a renderer to illuminate the scene (Figure 5d), the HDR cloudy skies look fairly convincing in glossy reflections. A direct observation in the background is however showing the absence of detail and the low resolution of 256×256. The scenes look rather dark and lack contrast. While the generated HDR values match the scale of the training dataset, the clipped sun values are omitting vital energy for photorealistic rendering. 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 colour.

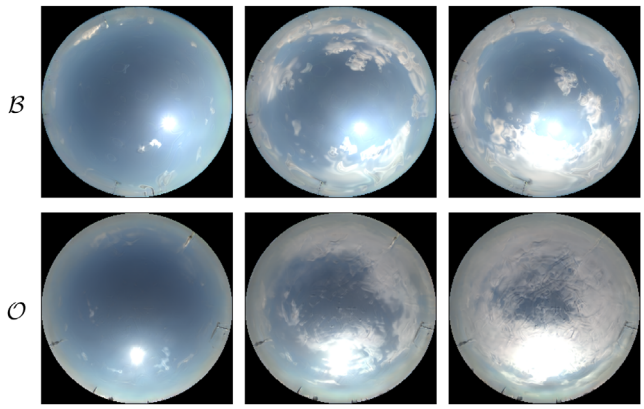


Figure 6: *The cloud coverage can be globally controlled by scaling the magnitude of the random latent vector w. Note how the sun moves for $\mathcal{B}$, as its position is entangled with the clouds in the latent space.*

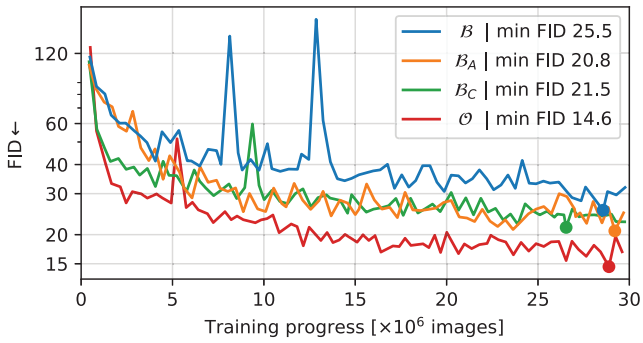


Figure 7: *Convergence plot in terms of the FID metric where lower is better. The best FID value is highlighted for each run.*

Controlling the cloud coverage. We demonstrate in Figure 6 how our method can also produce images with varying cloud coverage 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 accomplished 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 7, we generate 35 424 images (the used dataset size rounded to a multiple of the generation batch size) and compute the FID score. The metric is based on a neural network that was pretrained 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 25.5 in the training time of 30 million images. In comparison, Karras et al. [KAL*21] report values of 5 and below for their networks trained on faces. Note that the absolute FID values are not directly comparable, because their dataset is two times larger, and is more or less aligned: facial features are present at roughly the same positions with eyes exactly matching. In contrast, cloud covers are more or less *random*. $\mathcal{O}$urs achieved a minimum FID of 14.6 and the individual features ($\mathcal{B}_A$ and $\mathcal{B}_C$) lie in between with 20.8 and 21.5 respectively. This aligns with the impression of the visual inspection above. We therefore conclude that both our proposed features have a positive effect on the training convergence and image quality.

6. Discussion and Future Work

In the previous section, we saw that our method can generate 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.

Direct sun radiance. The radiance (light energy) coming from the sky can be split into indirect radiance, which arrives from the Sun and is scattered in the atmosphere or clouds, and direct radiance, which comes 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, LAB*16] and possibly generate the sun disk in a separate dedicated network layer. The encoded clear sky could also be augmented with the correctly attenuated sun disk using the transmission component of the Prague sky model [WVB*21]. Extra care would have to be taken in case a sun disk is only partially occluded by a cloud. Another improvement could be to train the networks in LDR brackets which would avoid potential numerical issues and distribute the impact of the losses more evenly across the value range.

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 the sea 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, resulting in fewer problems in training, so it is an important future research direction.

Resolution. Our network was trained on 256 × 256 pixels, but for sharp backgrounds in high-resolution renders a resolution of 8196

or more would be required. This seems utopic given that resolutions of 512 or 1024 already showed significantly lower quality results, because of the GPU memory (48 GB) limiting the network capacity. We see two possible solutions: First, the tiling of several lower-resolution cutouts of the skydome or second, the usage of up-sampling networks that add detail and sharpness. Additionally, it may be worth adapting our approach to use a visual transformer-based image generator like VQGAN [ERO21].

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 to output certain types of clouds, for example thin high-altitude clouds vs dense thunderstorm clouds. We will also inspect the generator to find latent dimensions responsible for certain cloud positions and types to enable spatial control of where precisely the clouds should appear, a concept similar to GauGAN [PLWZ19].

Consistency. Our dataset contains time-lapse photographs which could be exploited to train a time-consistent generator. We have observed an implicit organization of the latent space along the time dimension, but it currently is not user-accessible nor is this consistency explicitly trained for.

When traversing the latent space, the sun position is not always steady but moves with the clouds. Future work could address this consistency of sun position by passing the discriminator pairs of clear and cloudy images and let it judge the consistency between the two. In addition, the discriminator could be tasked with reconstructing the sun position as further conditioning.

7. Conclusion

We have shown a GAN that produces cloudy skies usable for rendering of 3D scenes. The proposed clear sky encoder approach helps to directly parametrize the sun position while our data augmentation technique of marginalising the azimuth increases the variety in cloud shapes for each elevation. 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. We also hope for future work to increase the resolution in order to prepare our method for production usage.

With this paper, we open 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.

Acknowledgements

We thank the reviewers for their extensive feedback on the manuscript and Karolína Švarcová for checking the dataset. 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. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). This work was supported through the Intel oneAPI Center of Excellence at Charles University.

References

[ATO*19] Andrianakos G., Tsourounis D., Oikonomou S., Kastaniotis D., Economou G., Kazantzidis A.: Sky image forecasting with generative adversarial networks for cloud coverage prediction. *10th International Conference on Information, Intelligence, Systems and Applications (IISA)* (Patras, Greece, July 2019), IEEE, p. 7. https://doi.org/10.1109/IISA.2019.8900774.

[BN08] Bruneton E., Neyret F.: Precomputed atmospheric scattering. *Computer Graphics Forum 27* (2008), 1079–1086.

[Bru16] Bruneton E.: A qualitative and quantitative evaluation of 8 clear sky models. *IEEE Transactions on Visualization and Computer Graphics 23*, 12 (2016), 2641–2655.

[CZR22] Chalmers A., Zickler T., Rhee T.: Illumination browser: An intuitive representation for radiance map databases. *Computers & Graphics 103* (Apr. 1, 2022), 101–108.

[Deb98] Debevec P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, July 1998), SIGGRAPH '98, Association for Computing Machinery, pp. 189–198. https://doi.org/10.1145/280814.280864.

[DWX*21] Ding X., Wang Y., Xu Z., Welch W. J., Wang Z. J.: CcGAN: Continuous Conditional Generative Adversarial Networks for Image Generation. Oral Presentation. Jan. 2021. https://openreview.net/forum?id=PrzjugOsDeE.

[EBK*16] Emde C., Buras-Schnell R., Kylling A., Mayer B., Gasteiger J., Hamann U., Kylling J., Richter B., Pause C., Dowling T., Bugliaro L.: The libRadtran software package for radiative transfer calculations (version 2.0.1). *Geoscientific Model Development 9*, 5 (2016), 1647–1672.

[EGH21] Einabadi F., Guillemaut J.-Y., Hilton A.: Deep neural models for illumination estimation and relighting: A survey. *Computer Graphics Forum 40*, 6 (2021), 315–331.

[EKD*17] Eilertsen G., Kronander J., Denes G., Mantiuk R. K., Unger J.: HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics 36*, 6 (Nov. 2017), 178:1–178:15.

[ERO21] Esser P., Rombach R., Ommer B.: Taming transformers for high-resolution image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), IEEE, pp. 12868–12878. https://doi.org/10.1109/CVPR46437.2021.01268.

[GGJ18] Guimera D., Gutierrez D., Jarabo A.: A physically-based spatio-temporal sky model. *Spanish Computer Graphics Conference (CEIG)* (Madrid, Spain, 2018), The Eurographics Association, pp. 29–37. https://doi.org/10.2312/ceig.20181150.

[GPM*14] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial networks. *Advances in Neural Information Processing Systems* (2014), vol. *27*, Curran Associates, Inc. arXiv: 1406.2661. http://arxiv.org/abs/1406.2661.

[HAL19] Hold-Geoffroy Y., Athawale A., Lalonde J.-F.: Deep sky modeling for single image outdoor lighting estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, June 2019), pp. 6920–6928. https://doi.org/10.1109/CVPR.2019.00709.

[HHP*21] Herrera J. A. A., Hädrich T., Pałubicki W., Banuti D. T., Pirk S., Michels D. L.: Weatherscapes: Nowcasting heat transfer and water continuity. *ACM Transactions on Graphics (TOG) 40*, 6 (Dec. 2021), 1–19. ISSN: 0730-0301.

[HMP*20] Hädrich T., Makowski M., Pałubicki W., Banuti D. T., Pirk S., Michels D. L.: Stormscapes: Simulating cloud dynamics in the now. *ACM Transactions on Graphics 39*, 6 (Nov. 2020), 175:1–175:16. ISSN: 0730-0301.

[HMS05] Haber J., Magnor M., Seidel H.-P.: Physically-based simulation of twilight phenomena. *ACM Transactions on Graphics 24*, 4 (Oct. 2005), 1353–1373. New York, NY, USA: ACM. https://doi.org/10.1145/1095878.1095884.

[Hoj19] Hojdar t.: Using Neural Networks to Generate Realistic Skies. MA thesis, Charles University, Faculty of Mathematics and Physics, 2019. http://hdl.handle.net/20.500.11956/109131.

[HR05] Horváth G., RawTherapee Contributors: Rawtherapee, 2005. https://www.rawtherapee.com/.

[HSH*17] Hold-Geoffroy Y., Sunkavalli K., Hadap S., Gambaretto E., Lalonde J.-F.: Deep outdoor illumination estimation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, Hawaii, July 2017), pp. 2373–2382. ISSN: 1063-6919. https://doi.org/10.1109/CVPR.2017.255.

[HW12] Hošek L., Wilkie A.: An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics 31*, 4 (July 2012), 95:1–95:9. ISSN: 0730-0301.

[HW13] Hošek L., Wilkie A.: Adding a solar-radiance function to the Hošek-Wilkie skylight model. *IEEE Computer Graphics and Applications 33*, 3 (May 2013), 44–52. ISSN: 1558-1756.

[JMD21] Jain M., Meegan C., Dev S.: Using gans to augment data for cloud image segmentation task. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS* (July 11, 2021), pp. 3452–3455. https://doi.org/10.1109/IGARSS47720.2021.9554993.

[KAH*20] Karras T., Aittala M., Hellsten J., Laine S., Lehtinen J., Aila T.: Training generative adversarial networks with limited data. (Oct. 2020). arXiv: 2006.06676. URL: http://arxiv.org/abs/2006.06676.

[KAL*21] Karras T., Aittala M., Laine S., Härkönen E., Hellsten J., Lehtinen J., Aila T.: Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* (2021), vol. *34*, Curran Associates, Inc., pp. 852–863. https://proceedings.neurips.cc/paper_files/paper/2021/file/076ccd93ad68be51f23707988e934906-Paper.pdf.

[KMM*17] Kallweit S., Müller T., Mcwilliams B., Gross M., Novák J.: Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics 36*, 6 (Nov. 2017), 231:1–231:11.

[LAB*16] Lalonde J.-F., Asselin L.-P., Becirovski J., Hold-Geoffroy Y., Garon M., Gardner M.-A., Zhang J.: The laval HDR sky database. http://sky.hdrdb.com, 2016.

[LM14] Lalonde J.-F., Matthews I.: Lighting estimation in outdoor image collections. *2014 2nd International Conference on 3D Vision* (Tokyo, Japan, Dec. 2014), vol. *1*, IEEE Computer Society, pp. 131–138. ISSN: 1550-6185. https://doi.org/10.1109/3DV.2014.112.

[MO14] Mirza M., Osindero S.: Conditional generative adversarial nets. (Nov. 2014). arXiv: 1411.1784. http://arxiv.org/abs/1411.1784.

[MRI*22] Mirbauer M., Rittig T., Iser T., Kriváneк J., Šikudová E.: SkyGAN: Towards realistic cloud imagery for image based lighting. *Eurographics Symposium on Rendering* (2022), The Eurographics Association. https://doi.org/10.2312/sr.20221151.

[MRI*23] Mirbauer M., Rittig T., Iser T., Křivánek J., Šikudová E.: CGGMFF/SkyGAN, 2023. https://github.com/CGGMFF/SkyGAN.

[New01] New House Internet Services B.V.: PTGui: Photo stitching software, 2001. https://ptgui.com/.

[PLWZ19] Park T., Liu M.-Y., Wang T.-C., Zhu J.-Y.: Semantic image synthesis with spatially-adaptive normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, Nov. 2019), pp. 2332–2341. https://doi.org/10.1109/CVPR.2019.00244.

[PSM93] Perez R., Seals R., Michalsky J.: All-weather model for sky luminance distribution—Preliminary configuration and validation. *Solar Energy 50*, 3 (Mar. 1993), 235–245. ISSN: 0038-092X.

[PSS99] Preetham A. J., Shirley P., Smits B.: A practical analytic model for daylight. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (Los Angeles, CA, USA, 1999), ACM SIGGRAPH, pp. 91–100. https://doi.org/10.1145/311535.311545.

[SB23] Satilmis P., Bashford-Rogers T.: Deep dynamic cloud lighting. *arXiv preprint arXiv:2304.09317* (2023).

[SJW*06] Stumpfel J., Jones A., Wenger A., Tchou C., Hawkins T., Debevec P.: Direct HDR capture of the sun and sky. *ACM SIGGRAPH 2006 Courses* (Boston Massachusetts, July 2006), SIGGRAPH '06, Association for Computing Machinery, pp. 5–es. https://doi.org/10.1145/1185657.1185687.

[SK21] Somanath G., Kurz D.: HDR environment map estimation for real-time augmented reality. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 11293–11301. https://doi.org/10.1109/CVPR46437.2021.01114.

[SMDB22] Satilmis P., Marnerides D., Debattista K., Bashford-Rogers T.: Deep synthesis of cloud lighting. *IEEE Computer Graphics and Applications* (2022). https://doi.org/10.1109/MCG.2022.3172846.

[SP07] Stafford B., Pysolar Contributors: Pysolar, 2007. https://github.com/pingswept/pysolar.

[Špa20] Špaček J.: Generation of Realistic Skydome Images. MA thesis, Charles University, Faculty of Mathematics and Physics, 2020. URL: http://hdl.handle.net/20.500.11956/109131.

[TYS09] Tao L., Yuan L., Sun J.: SkyFinder: attribute-based sky image search. *ACM Transactions on Graphics 28*, 3 (July 27 2009), 68:1–68:5.

[VBKW22] Vévoda P., Bashford-Rogers T., Kolářová M., Wilkie A.: A wide spectral range sky radiance model. *Computer Graphics Forum 41*, 7 (2022), 291–298. ISSN: 1467-8659.

[VMM*23] Valença L., Maquignaz I., Moazen H., Madan R., Hold-Geoffroy Y., Lalonde J.-F.: Lm-gan: A photorealistic all-weather parametric sky model. (2023). arXiv:2302.00087.

[WVB*21] Wilkie A., Vevoda P., Bashford-Rogers T., Hošek L., Iser T., Kolářová M., Rittig T., Křivánek J.: A fitted radiance and attenuation model for realistic atmospheres. *ACM Transactions on Graphics 40*, 4 (July 2021), 135:1–135:14. ISSN: 0730-0301.

[YGH*21] Yu P., Guo J., Huang F., Zhou C., Che H., Ling X., Guo Y.: Hierarchical disentangled representation learning for outdoor illumination estimation and editing. *IEEE International Conference on Computer Vision (ICCV)* (Oct. 2021), pp. 15293–15302. https://doi.org/10.1109/ICCV48922.2021.01503.

[YME*20] Yu Y., Meka A., Elgharib M., Seidel H.-P., Theobalt C., Smith W. A. P.: Self-supervised outdoor scene relighting. *European Conference on Computer Vision (ECCV)* (Aug. 2020), pp. 84–101. https://doi.org/10.1007/978-3-030-58542-6_6.

[ZPIE17] Zhu J.-Y., Park T., Isola P., Efros A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, Oct. 2017), pp. 2242–2251. ISSN: 2380-7504. https://doi.org/10.1109/ICCV.2017.244.

[ZSH*19] Zhang J., Sunkavalli K., Hold-Geoffroy Y., Hadap S., Eisenman J., Lalonde J.-F.: All-weather deep outdoor lighting estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, June 2019), pp. 10150–10158. ISSN: 2575-7075. https://doi.org/10.1109/CVPR.2019.01040.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information