

# A Fitted Radiance and Attenuation Model for Realistic Atmospheres: Supplemental Document

ALEXANDER WILKIE\*, Charles University, Czech Republic

PETR VEVODA\*, Charles University, Czech Republic

THOMAS BASHFORD-ROGERS, University of the West of England, United Kingdom

LUKÁŠ HOŠEK, Charles University, Czech Republic

TOMÁŠ ISER, Charles University, Czech Republic

MONIKA KOLÁŘOVÁ, Charles University, Czech Republic

TOBIAS RITTIG, Charles University, Czech Republic

JAROSLAV KŘIVÁNEK, Charles University, Czech Republic

## 1 FITTING THE IN-SCATTERED RADIANCE FUNCTION

### 1.1 Choice of Mathematical Approach

We first extensively experimented with techniques similar to that of the Perez et al. [1993] model and its descendants. The approach of these models is to assume the sky-dome radiance patterns to consist of separable features – e.g. a gradient between the zenith and the horizon plus a radial bright patch around the Sun. In these models, radiance is calculated as a function of ray direction (given as a pair of angles, a “solar” and a “zenith” angle) and a small number of configuration factors: numbers that control the strength of each individual feature, and which are found by means of nonlinear optimisation. This approach works as long as the number of features is kept low: but that obviously limits the range of radiance patterns the model can reliably reproduce.

For our purposes, we need an expression that works for a full sphere instead of just the upper hemisphere: in particular, it has to be able to handle the discontinuity which is present at the horizon in most radiance configurations. It also has to have terms that approximate the features of the sky well, including two phenomena specific to twilight skies: post sunset, the Earth casts a shadow onto the atmosphere, which produces a wedge of darker color at the horizon. Above it, there is a second wedge of brighter pinkish back-scattered light (called “Belt of Venus” or “anti-twilight arch”): Fig. 9 in the paper shows how these features develop as the sun goes beneath the horizon.

Theoretically, a suitable mix of features could be devised by educated guess and trial and error. However, even if we managed to find such features, the fitting process becomes slower with each new parameter in a non-linear fashion, plus more memory consuming and prone to getting stuck in local minima. Extensive experimentation showed the old feature-based approach is simply not suitable for radiance fitting on the full sphere anymore: there are too many features in fully spherical radiance patterns which cannot be cleanly separated. This even applies to models which attempted to separate just the polarisation patterns: [Kreuter and Blumthaler 2013] also only managed to work with the upper hemisphere.

\*Alexander Wilkie and Petr Vevoda share the first authorship of this work.

Authors’ addresses: [wilkie, vevoda]@cgg.mff.cuni.cz, Univerzita Karlova, KSVI MFF, Malostranské náměstí 25, 118 00 Prague 1, Czech Republic.

This is why we opted for an entirely new approach: we obtain the radiance pattern of the sky as a sum of outer products of single variable functions. The functions themselves are free-form, tabulated and were obtained by Canonical Polyadic Decomposition (CPD) [Kolda and Bader 2009], a process very similar to SVD low rank approximation. This approach can be thought of as a specialised compression scheme, however it is also essentially a decomposition of the radiance pattern into an optimal orthogonal set of “features”.

The methods that we describe in the next sections all rely on tensor and matrix decompositions. An alternative choice could have been to use neural networks, similar to [Satılmış et al. 2016][Hold-Geoffroy et al. 2019][Zhang et al. 2019]. However, while learning approaches do have merits, reliability of reconstruction is not one of them. Additionally, they tend to incur a higher runtime overhead than our model.

### 1.2 Input Parametrisation

It is desirable to choose a parametrisation in which the features are as axis-aligned as possible, as that makes the input matrix easy to decompose into separable matrices by CPD / SVD. For a given solar elevation  $\eta$ , the natural parametrisation of a sky-dome model is the view direction, represented as a unit vector  $\vec{v}$ . For better separability we transform it into a set of angles (see Fig. 1):

The first one is the solar angle  $\gamma$ . The solar angle is the angle formed by the view direction and the direction towards the centre of the solar disc. This makes the gradient of the solar glow roughly parallel to the first axis – *roughly* because, as can be seen from reference solutions obtained with libradtran, the solar glow does not actually form a perfect circle around the sun, but tends to extend further to the sides and downward.

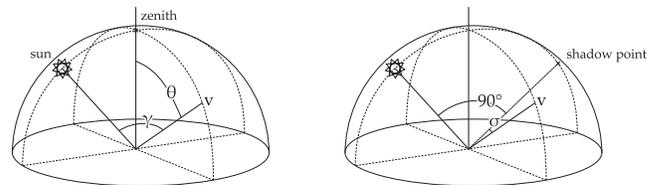


Fig. 1. The angles used in the model.

We also need the model to describe Earth's shadow. So when fitting images of post-sunset skies, we chose the shadow angle  $\sigma$  to be second input angle / axis. This angle is formed by the view ray and a *shadow point* – an imaginary point lying at  $90^\circ$  away from the centre of the solar disc in the direction of the zenith. The shadow line is perpendicular to the solar angle gradient and parallel to the shadow angle gradient, which makes it aligned with both axes.

For regular daytime skies, no shadow is visible, and using the shadow angle as an input parameter provides no benefit since its axis is not parallel or perpendicular to any feature visible on the sky. We instead use the zenith angle  $\theta$  as the second parameter for these configurations. This is the angle formed by the view direction and the direction towards the zenith. This makes the horizon perpendicular to the axis defined by the horizon angle.

The model switches between these two modes at solar elevation  $0^\circ$ , when the zenith point aligns exactly with the shadow point, which makes the transition seamless. To present this approach in a unified manner, we introduce an angle  $\alpha$ , which we for lack of a better name call simply the *zenith/shadow angle*, which is equivalent to the zenith angle for solar elevations greater than  $0^\circ$  and equivalent to the shadow angle otherwise. Note that this makes the horizon not aligned with any axis in post-sunset skies, which makes fitting of the horizon tricky. The image emphasis process described later was developed to make horizon fitting more accurate in these conditions.

### 1.3 High Altitude Angle Correction

Our model consists of a finite number of fitted sky-dome configurations, and intermediate states have to be interpolated in a way that generates plausible sky-dome appearance. It is a key feature of the CPD / SVD separation that this is actually possible - at least for some sky-dome features.

A case that works is the circumsolar glow: there, taking the fitted data for a specific solar elevation (e.g.  $\eta = 20^\circ$ ), and using it to generate a different elevation (e.g.  $\eta = 30^\circ$ ) yields useable results: the  $\vec{v} \rightarrow (\gamma, \alpha)$  re-projection process warps the image correctly, and the solar glow gets moved to the right place.

This unfortunately does not work for the horizon, as that changes in a manner that is too complex for simple re-projection to handle. It only appears as a line at  $\theta = 90^\circ$  for altitude = 0: and as can be seen in Fig. 8 in the paper, it moves downwards for higher observer altitudes, and turns into a curve. If we had data for just two observer altitudes, e.g. 100 m, and 100 km, interpolation between these states would contain two blended horizons, instead of a single one.

We fix this issue via the way the  $\theta$  and  $\sigma$  angles (and thereby also  $\alpha$ ) are calculated. If the view direction  $\vec{v}$  is tangent to the Earth's surface, its  $\theta$  will always be  $90^\circ$ , regardless of altitude.

The un-corrected way of calculating the angles is as follows: the directions towards the solar, zenith and shadow points are given as unit vectors  $\vec{s}$ ,  $\vec{z}$  and  $\vec{u}$ . Assuming that the z-axis points upwards,  $\vec{z} = (0, 0, 1)$ , the angles can be calculated as follows:

$$\begin{aligned} \gamma &= \cos^{-1}(\vec{v} \cdot \vec{s}) \\ \theta &= \cos^{-1}(\vec{v} \cdot \vec{z}) \\ \sigma &= \cos^{-1}(\vec{v} \cdot \vec{u}) \end{aligned} \quad (1)$$

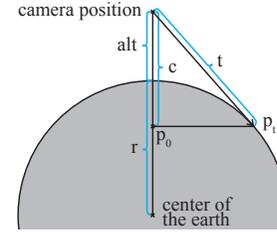


Fig. 2. Horizon correction geometry.  $r$  - Earth radius,  $alt$  - camera altitude,  $c$  - correction length,  $t$  - tangent length,  $p_t$  - point of tangency,  $p_o$  - virtual ray origin.

With an observer altitude above ground, the tangent from the camera origin towards the horizon ( $a$  in Fig. 2) is not perpendicular to the zenith direction  $\vec{z}$ . To correct that, we project the point of tangency  $p_t$  onto the line from Earth's centre to the observer to obtain the *virtual ray origin*,  $p_o$ . We denote  $\vec{v}'$  the direction from  $p_o$  to  $p_t$ . The direction  $\vec{v}'$  can be expressed in terms of the original direction  $\vec{v}$ , camera altitude  $Alt$  and Earth radius  $R$  as:

$$\vec{v}' = \text{normalize}(\vec{v} - \overrightarrow{cor\vec{r}}) \quad (2)$$

with the *correction vector*  $\overrightarrow{cor\vec{r}}$  being defined as:

$$\begin{aligned} \overrightarrow{cor\vec{r}} &= (0, 0, \frac{c}{t}) \\ c &= r + alt - \frac{r^2}{r + alt} \\ t &= \sqrt{(r + alt)^2 - r^2} \end{aligned} \quad (3)$$

This correction to the view direction is applied in the model for the purpose of calculating the zenith and shadow angles.

### 1.4 The Core Function

The function which evaluates the in-scattered radiance is a function of two parameters  $\gamma, \alpha$ . The function is internally represented as an outer product of two single parameter functions:

$$\mathbb{F}(\gamma, \alpha) = \sum_{i=1}^n \mathbb{F}_{solar}^{(i)}(\gamma) \cdot \mathbb{F}_{zenith/shadow}^{(i)}(\alpha) \quad (4)$$

The functions  $\mathbb{F}_{solar}$  and  $\mathbb{F}_{zenith/shadow}$  are tabulated and provided as part of the model. The tabulated functions are obtained by re-projecting the fish-eye image into the  $(\gamma, \alpha)$  space, essentially producing a 2D look-up table of  $\mathbb{F}(\gamma, \alpha)$ , and then decomposing the look-up table into outer vector products using CPD. The process is described in detail in Sec. 1.6.

### 1.5 Improved Horizon Fitting with Image Emphasis

Recall that in order to make the decomposition perform properly, the angle  $\alpha$  was defined so that the horizon is a horizontal line in the re-projected image. This only holds for daytime solar elevations: post sunset the horizon is not axis-aligned anymore, so the composition performs poorly in those cases. To deal with this, we remove the sharp horizon from the images prior to transforming them into

$(\gamma, \alpha)$  space, and only then perform the fitting which then works satisfactorily on a blurred horizon that is not axis aligned. This process, which we call *pre-emphasis*, is reverted when using the model: the *de-emphasis* we perform then returns the sharp horizon transition to its place. We can accurately do this, as the location of the horizon is analytically known.

The pre-emphasis process works as follows: We denote  $I_{orig}$  the original input image and  $I_{orig}(x)$  its value at pixel position  $x$ . The input image  $I_{orig}$  is cut into two parts very close to the horizon line (we chose a value of 30' above the horizon). The top/above horizon part is left intact. The bottom / below horizon part is deleted and infilled using the standard `regionfill` algorithm of Matlab version 2019b, which smoothly interpolates inward. This essentially removes the sharp horizon transition and leaves the bottom part of the image a completely featureless smooth gradient with brightness equivalent to that of the sky-dome area just above the horizon. We denote this newly in-filled image as  $I_{infill}$ . We also note a single value  $g$  denoting the ratio of the average brightness of the original below-horizon area relative to the brightness of the newly in-filled area.

By linearly interpolating between  $I_{orig}$  and  $I_{infill}$ , we create a guide image  $I_{guide}$ . The control value for the interpolation is a function of  $\kappa$  and the pixel position. When  $\kappa$  is high, meaning that the below-horizon part of the image was already bright enough in the original, we mostly leave the original intact, except for the very bottom part near the nadir, which we always replace by the infill, because it's generally always noisy.

$$I_{guide}(x) = \text{lerp}\left(I_{orig}(x), I_{infill}(x), V_c(x)\right)$$

$$V_c(x) = \text{sat}\left(\text{sat}\left(\frac{\theta(x) - \pi/2}{\pi/2}\right) + \text{sat}\left(\frac{\kappa - 0.5}{-0.2}\right)\right) \quad (5)$$

Where:

- $V_c$  is the control value of the linear interpolation.
- `sat` is the *saturate* function that clamps the value to  $[0, 1]$ .
- $\theta(x)$  is the zenith angle value at pixel location  $x$ .

The control value  $V_c$  consists of two terms: the first one makes sure that the original image is always gradually replaced by the in-filled image, starting at  $\theta = 90^\circ$  (at the horizon) and progressing towards  $\theta = 180^\circ$  (the nadir). The second term makes sure that if the bottom part of the original image was too dark, it is replaced completely, the effect being gradually applied depending on  $\kappa$ , starting at  $\kappa = 0.5$  and finishing at full strength at  $\kappa = 0.3$ .

Having the guide image ready, we calculate a pixel-wise ratio between the guide and the original image:

$$I_{ratio}(x) = \frac{I_{orig}(x)}{I_{guide}(x)} \quad (6)$$

The ratio image is then used to define the de-emphasis, a function of zenith angle  $\theta$ :

$$E(z) = \text{mean}_{x; \theta(x)=z} I_{ratio}(x) \quad (7)$$

The function  $E(z)$  is tabulated and becomes part of the model data. Next, the pre-emphasized image is calculated by applying the inverse of the de-emphasis function:

$$I_{preemph}(x) = \frac{I_{orig}(x)}{E(\theta(x))} \quad (8)$$

This pre-emphasized image is then used as an input for CPD.

## 1.6 The Fitting Process

We now have all the components required to perform the fitting. To recapitulate the whole fitting process:

- (1) We start with  $I_{orig}$ . This is the reference rendering of the sky-dome produced by the path-tracer.
- (2) Pre-emphasis is performed on  $I_{orig}$ , yielding a pre-emphasised image  $I_{preemph}$ , and a de-emphasis function  $E$ .
- (3)  $I_{preemph}$  is re-projected into  $(\gamma, \alpha)$  space, essentially producing a two-dimensional look-up table of  $\mathbb{F}(\gamma, \alpha)$ .
- (4) The re-projected image is partially in-filled and filtered. More on that later in this section.
- (5) A CPD decomposition is performed, yielding pairs of one-dimensional tabulated functions  $\mathbb{F}_{solar}^{(i)}$  and  $\mathbb{F}_{zenith/shadow}^{(i)}$ .

The final products of the fitting are:

- The de-emphasis function  $E$
- The tabulated functions  $\mathbb{F}_{solar}^{(i)}$  and  $\mathbb{F}_{zenith/shadow}^{(i)}$

These constitute all the data required to render the sky-dome using the analytical model.

We have chosen the dimensions of the  $(\gamma, \alpha)$  re-projected image to be  $361 \times 361$  (i.e.  $0.5^\circ$ /pixel since the valid values of  $\gamma$  and  $\alpha$  are  $0^\circ - 180^\circ$ ). This image is computed by transforming each pair of  $\gamma, \alpha$  values into the reference rendering followed by bilinear filtering to avoid artefacts. This re-projected image is then decomposed into an outer vector product using the CPD low rank approximation algorithm. Note that in theory, we could extend this process by unwrapping the input into a three or even higher dimensional look-up table, e.g. parametrised by  $(\gamma, \theta, \sigma)$ . CPD is a tensor decomposition algorithm and would deal with the resulting tensor natively. The problem of this approach is that the valid combinations of angles form a 2D manifold inside this 3D space – in other words the tensor is mostly undefined, which makes the decomposition unstable.

Even in 2D, the issue of undefined values requires us to in-fill parts the re-projected image. Not all combinations of angles are valid, e.g. in a sky where the sun is at the horizon there is no direction that would correspond to both  $\gamma$  and  $\alpha$  being 0. The valid combinations form a parallelogram, see Fig. 3. The CPD algorithm deals natively with undefined values, however there is no guarantee what the undefined part is going to look like in the resulting approximation. In Sec. 1.3, we claimed that data from one sky-dome configuration can be re-used e.g. for other solar elevations if it is suitably reprojected. This is true, but a potential issue arises due to the changing shape of the parallelogram of valid combinations: upon re-projection, we might attempt to read from an undefined part of the  $(\gamma, \alpha)$  image.

To fix this, the re-projected image has to be partially infilled. The valid area of the tensor is dilated, and the missing data is again filled using Matlab's `regionfill` algorithm. The amount of dilation is the

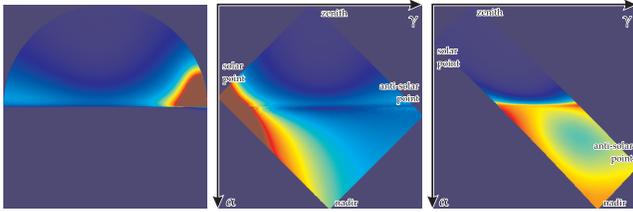


Fig. 3. **Left:** A single spectral channel of the input fisheye image in false color. Solar elevation  $8^\circ$ , observer altitude 15 m, so the image is almost completely zero below the horizon line. **Middle:** The same image transformed to  $(\gamma, \alpha)$  space: the valid combinations of  $\gamma$  and  $\alpha$  form a parallelogram. The below-horizon part of the image is amplified by the pre-emphasis process. **Right:** The  $(\gamma, \alpha)$  remapping of a different sky-dome, solar elevation  $55^\circ$ : this illustrates the changing shape of the parallelogram of valid values.

minimal amount required to cover the intermediate values between the provided fittings (see Table 2).

After in-filling, the tensor also has to be filtered, as CPD decomposition not only retains noise present in the input, but moves it to  $(\gamma, \alpha)$  space: so instead of grain-like Monte Carlo noise, there are ringing artefacts. To be useable, the final fitting has to be completely devoid of any such artefacts: even the slightest unevenness is *very* apparent in renderings, especially as they are usually dissimilar across spectral channels, and show up as rainbow effects. So the reference renderings are filtered (using the `wdenoise` wavelet noise reduction algorithm found in Matlab version 2019b), and after the image is reprojected to a tensor and in-filled, it is again filtered, this time using a gaussian blur with  $\sigma = 1$  both in the solar and zenith/shadow axis (recall that the resolution of the tensor is  $0.5^\circ$  per pixel).

Regarding the decomposition itself, we have chosen to use decomposition rank  $n = 9$ , i.e. the decomposition produces ten sets of vectors. In all images we have tested,  $n = 9$  produces a decomposition that explains  $> 99.5\%$  of the variance in the tensor.

The tensor decomposition produces vectors of length 361 ( $0.5^\circ$  increments). These do not have to be distributed in their entirety in the final model: they essentially represent tabulated functions, and the samples do not have to be placed uniformly. We have chosen to sample  $\mathbb{F}_{solar}$  densely at lower angles (areas directly surrounding the sun) and sparsely around the anti-solar point, giving us satisfactory results at 275 samples. Similarly,  $\mathbb{F}_{zenith/shadow}$  is sampled sparsely around the zenith and nadir and densely around the horizon, giving us 205 samples. The same approach has been used for the de-emphasis function, which is sampled more densely around the horizon, giving us 118 samples.

### 1.7 The evaluation process

The complete radiance function is evaluated as follows: for a given viewing direction the angles  $\gamma$ ,  $\alpha$  and  $\theta$  are computed first. Then the 9 pairs of tabulated functions  $\mathbb{F}_{solar}^{(i)}(\gamma)$ ,  $\mathbb{F}_{zenith/shadow}^{(i)}(\alpha)$  are looked up and combined according to equation 4. Finally, the result is multiplied by looked-up value of the de-emphasis function  $E(\theta)$ .

## 2 FITTING ATMOSPHERIC TRANSMITTANCE

For the atmospheric transmittance, we parameterise the atmosphere by altitude from ground level  $alt$  and distance along the planet surface  $d$ , see Fig. 4. This is motivated by a similar argument to Sec. 1.2 in that it is desirable to keep features as axis-aligned as possible; in our case, features are related to the 1D distribution of atmospheric constituents which is always aligned with the normal of the planet leading to the  $alt, d$  parametrisation. Transmittance is computed at the same set of altitudes as the in-scattered radiance model using ratio tracking at a set of points. These points are non-linearly distributed in the region of the atmosphere which covers the maximum viewable distances from the camera and are distributed to capture aspects of the atmosphere such as inversion layers and the transmittance falloff with distance; however, any other set of points can be used.

When applying the Singular Value Decomposition (SVD) to compress the data, we first non-linearly transform the data via a square root to boost small transmittance values which allows us to use a lower rank approximation than using untransformed data. We experimented with other invertible non-linear transforms, but the square root gave the best results and is fast to invert at runtime. The SVD produces a sorted list of eigenvectors and values  $U\Sigma V^*$  for each altitude. Reconstruction at an altitude  $alt$  uses the first  $R$  bases,  $U_{alt}(R)$ , with associated coefficients  $C_{alt}(R) = \Sigma_{alt}(R)V_{alt}^*(R)$ .  $U_{alt}(R)$  is  $R$  times the number of non-linearly distributed points for a single atmosphere configuration and  $C_{alt}(R)$  is  $R$  times the number of configurations. We use  $R = 12$ , leading to a maximum absolute error over the dataset of 0.014. The supplementary materials provide visual and numerical results for the transmittance component.

At runtime, the transmittance  $\tau(x_1, x_2)$  between two points in the atmosphere  $x_1$  and  $x_2$  at a given wavelength is computed from the reduced rank approximation. First, the basis coefficients corresponding to the atmospheric configuration and wavelength  $C'_R$  are interpolated from the nearest nearest altitudes to  $x_1$ . Then  $x_2$  is projected into the 2D parametrisation, leading to  $alt_{x_2}$  and  $d_{x_2}$ . The inner product between the basis evaluated at  $alt_{x_2}$  and  $d_{x_2}$  and  $C'_R$  are then computed for each altitude:

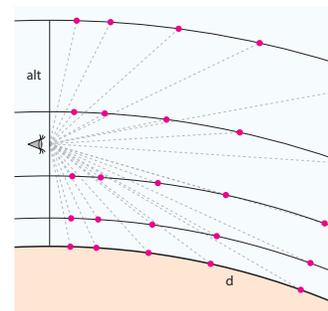


Fig. 4. Illustration of the 2D atmospheric parametrisation showing the non-uniformly distributed set of points (pink dots) where transmittance is calculated for the model.

$$\tau_{alt}(x_1, x_2) = \text{sat} \left( \left( U_{alt}(R)(alt_{x_2}, d_{x_2}) | C'_{alt}(R) \right)^2 \right) \quad (9)$$

This value is interpolated between altitudes resulting in the required transmittance value  $\tau(x_1, x_2)$ .

### 3 VALIDATION AGAINST LIBRADTRAN

The dataset for fitting our analytical model was generated using `atmo_sim`, an atmospheric path tracer which we developed ourselves using some polarisation rendering infrastructure from the ART framework [Wilkie 2018]. Details about why we chose to develop our own system are available in Sec. 4.4 in the main article. In this supplement, we provide a validation of the results computed by `atmo_sim` against those obtained by `libradtran`, a scientific software package for radiative transfer calculations within an atmosphere [Emde et al. 2016].

For the comparisons, we use `libradtran`'s Monte Carlo radiative transfer solver MYSTIC [Mayer 2009], which traces photons at given wavelengths through the atmosphere. We run MYSTIC individually for every single pixel at selected wavelengths 420 nm, 540 nm and 620 nm, which roughly corresponds to blue, green and red, respectively. To get comparable outputs from `atmo_sim`, we render spectral images with 46 bands with 10 nm spacing and extract corresponding single-wavelength data using the `tonemap` tool from the ART toolchain. The comparison images show radiances captured by a panoramic  $180^\circ \times 180^\circ$  camera with  $90^\circ$  towards the zenith (top half) and  $90^\circ$  towards the ground (bottom half).

#### 3.1 Atmosphere Without Aerosols

The first validation was done for a simple atmosphere without any aerosols. The only molecules taken into account are  $N_2$ ,  $O_2$  and  $O_3$  (ozone) and their concentrations correspond to the *U.S. Standard Atmosphere* from `libradtran` (see Sec. 3.2 in the main article).

In `atmo_sim`, we simulate Rayleigh scattering caused by  $N_2$  and  $O_2$ , and only take absorption into account for  $O_3$ . For the  $N_2$  and  $O_2$  Rayleigh scattering cross sections, both `atmo_sim` and `libradtran` use the same formula from Bodhaine et al. [1999, Equation 23]. For the  $O_3$  absorption cross sections, `atmo_sim` uses Gorshlev et al. [2014], as shown in Fig. 4 of the main article.

Validations against MYSTIC (Fig. 6) were performed with a diffuse ground albedo 0.2 for two different observer altitudes (0 km and 10 km) and solar elevations ( $5^\circ$  and  $45^\circ$ ). The chosen altitudes correspond to viewing the sky from the ground (0 km) and from a commercial airplane (with typical flight altitudes around 10 km). The elevation of  $5^\circ$  was chosen to validate the  $O_3$  absorption, as it is most noticeable at low sun elevations. The difference images are computed by dividing the radiances simulated by `atmo_sim` by the radiances from MYSTIC, which shows which areas are brighter and darker, respectively.

The average difference at solar elevation  $45^\circ$  is 1.0025 with an average signal correlation of 0.99978. The error is uniformly distributed over the images with no apparent patterns.

At  $5^\circ$ , the average difference is 1.0002 but with a higher variance than at  $45^\circ$ . The average signal correlation is 0.99960. Notice that `atmo_sim` is darker at higher wavelengths, which is most likely due

to a slightly different  $O_3$  absorption curve, as this does not happen when we remove  $O_3$  from the atmosphere.

Finally, notice that in some comparisons there is a 1 pixel horizontal stripe of large differences between `atmo_sim` and MYSTIC. The stripe is always located at the planet edge, i.e. at the horizon for altitudes of 0 km, or slightly below the horizon for 10 km. These artefacts are most likely caused by sub-pixel sampling and jittering in `atmo_sim`, when for a pixel on the planet edge, some of the photon paths hit the planet and some do not. On the other hand, in our script that evaluates MYSTIC, we always simulate the radiance in the middle of a pixel, hence it is always either above or below the planet edge with no jittering and randomness.

#### 3.2 Atmosphere With Aerosols

We now take the simple atmosphere from Sec. 3.1 and extend it by adding aerosols according to Hess et al. [1998], with concentrations corresponding to a *continental average* atmosphere from `libradtran`. The aerosols include *water-insoluble* (INSO) particles consisting mostly of soil and organic materials, *water-soluble* (WASO) particles at 50% humidity consisting of various sulfates, nitrates and others, and finally *black carbon* (SOOT).

The optical properties of the different aerosols can be computed as discussed in the documentation for the OPAC (Optical Properties of Aerosols and Clouds) software package [Hess et al. 1998], and are also available online<sup>1</sup>. In `libradtran`, the optical properties are computed using Mie theory from the particle size distributions, and using refractive indices from OPAC. In our `atmo_sim` verification, we used the same absorption and scattering cross sections as used in `libradtran`.

However, as discussed in Sec. 4.2.1 of the main paper, `atmo_sim` uses analytical Henyey-Greenstein phase functions [Henyey and Greenstein 1941] instead of precisely sampled phase functions from `libradtran`. The parameters  $g$  of WASO and SOOT aerosols are very close matches to “real” Mie lobes we computed ourselves, while we used more “blurry” asymmetry parameters provided by OPAC for the INSO particles. The effect of this simplification is that the circumsolar region has a considerably less “peaky” distribution of energy right next to the solar disc. This, in turn, makes the resulting function easier to fit, and `atmo_sim` rendering converges faster. However, it is worth noting that the more blurry asymmetry parameters we use are not per se *unrealistic* – they just deviate from what the U.S. Standard Atmosphere datasets should contain, in that the more blurry parameters correspond to different particles being present, instead of the actual INSO ones. The remainder of the atmosphere remains exactly as specified. Validations against MYSTIC (Fig. 7) were again performed in the same way as in Sec. 3.1.

As expected, due to the INSO phase function simplification discussed in the previous paragraph, the largest differences can be seen in the immediate circumsolar region, which is especially noticeable at 0 km and  $45^\circ$ . The inner parts of the solar glow are darker in `atmo_sim`, and the outer parts are lighter. The “real” INSO particles are very strongly forward-scattering, but we wanted to avoid using such an extreme phase function in `atmo_sim`. So the more “blurry” estimate provided by OPAC – possibly because they also had, at

<sup>1</sup><http://cds-espri.ipsl.fr/etherTypo/?id=989>

some point, a reason to avoid the very narrow “real” INSO lobes – came in very handy for our purposes.

To verify that all the observable differences are indeed due to the INSO scattering implementation in `atmo_sim`, we ran another experiment with only INSO aerosols present, where we forced `libradtran` to use the same Henyey-Greenstein phase function with the same asymmetry parameters as in `atmo_sim`. As we can see in Fig. 5, this completely eliminates all the noticeable differences and results in a perfect match across all the wavelengths. Hence, we conclude that the noticeable differences between `atmo_sim` and `libradtran` are caused by us using aerosol phase functions that are only approximated by the Henyey-Greenstein formula, and by using blurred asymmetry parameters for the INSO particle class.

### 3.3 Conclusion

To summarise, we can see that the radiances simulated by our atmospheric path tracer `atmo_sim` which we used for generating our reference dataset are good matches to results obtained with a well-established research-grade atmospheric library `libradtran` and its Monte Carlo radiative transfer solver MYSTIC.

Atmospheres without any aerosols yield almost exactly the same results, which means that Rayleigh scattering and absorption are simulated correctly. The addition of aerosols yields differences caused by the analytical Henyey-Greenstein phase function approximation, which is especially noticeable around the sun as its light spreads more. However, even then, the outer solar glow is only less than 2 times as bright than in `libradtran`, which we consider to be a perfectly valid approximation.

## 4 DATASET STATISTICS

### 4.1 Reference Dataset

The brute force rendered reference has 348 480 images (= 6 visibilities x 30 solar elevations x 22 observer altitudes x 4 ground albedos x 11 wavelengths x 2 stokes vector components). The dataset size is 255 GB in uncompressed form, 96 GB when compressed. It was created by running 3960 render jobs, each of which gives result images for 4 ground albedos times 11 wavelengths and two 2 Stokes components. For the post sunset solar elevations, 200k samples per pixel were used, for the remainder, 100k spp. We needed up to 600 core hours per render job, which was computed on a scientific supercomputing cluster (acknowledgments will be made in the non-anonymous version of the paper).

### 4.2 Fitted dataset

We use a decomposition rank of 9 for the radiance image, and 5 for the considerably smoother polarisation images. There are about 400 coefficients per fitted image. Currently, the size of a full polarisation-capable dataset is 410 MB per visibility: 247 MB radiance data + 127 MB polarisation data + 36 MB transmittance data. Consequently, the dataset for a non-polarising renderer is 283 MB.

## 5 FITTING STATISTICS

Each of the approximately 350k images was fitted separately, which took about 1 core hour per fit.

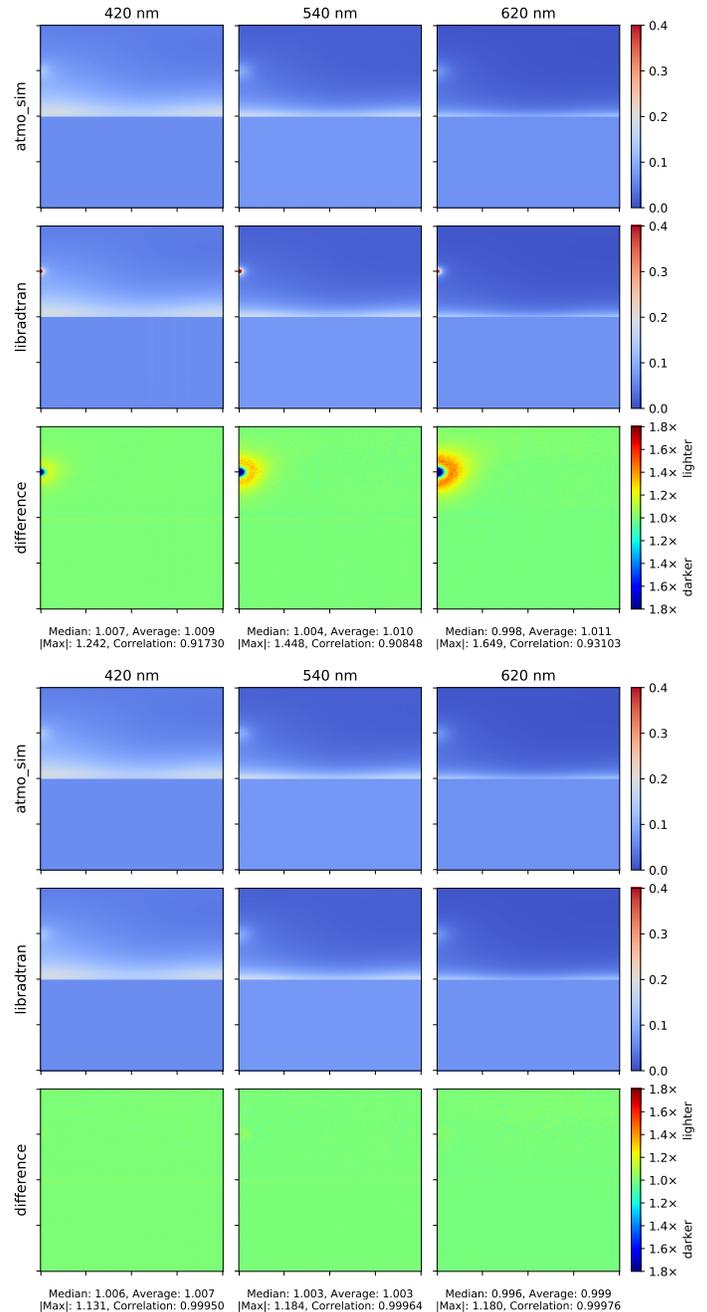


Fig. 5. Comparison of our Henyey-Greenstein phase function against a tabulated phase function for INSO scatterers. The atmosphere only contains molecules and INSO aerosols. **Top:** `atmo_sim` uses INSO Henyey-Greenstein phase function (see text for details), `libradtran` uses their tabulated phase function. Notice that our scattering is less forward, so the sun energy is blurred in a wider area, which makes the inner part darker and outer part brighter. **Bottom:** Both use the same Henyey-Greenstein phase function, the difference disappears. **Notation and scale:** See caption of Fig. 6, where the image notations and scales are explained.

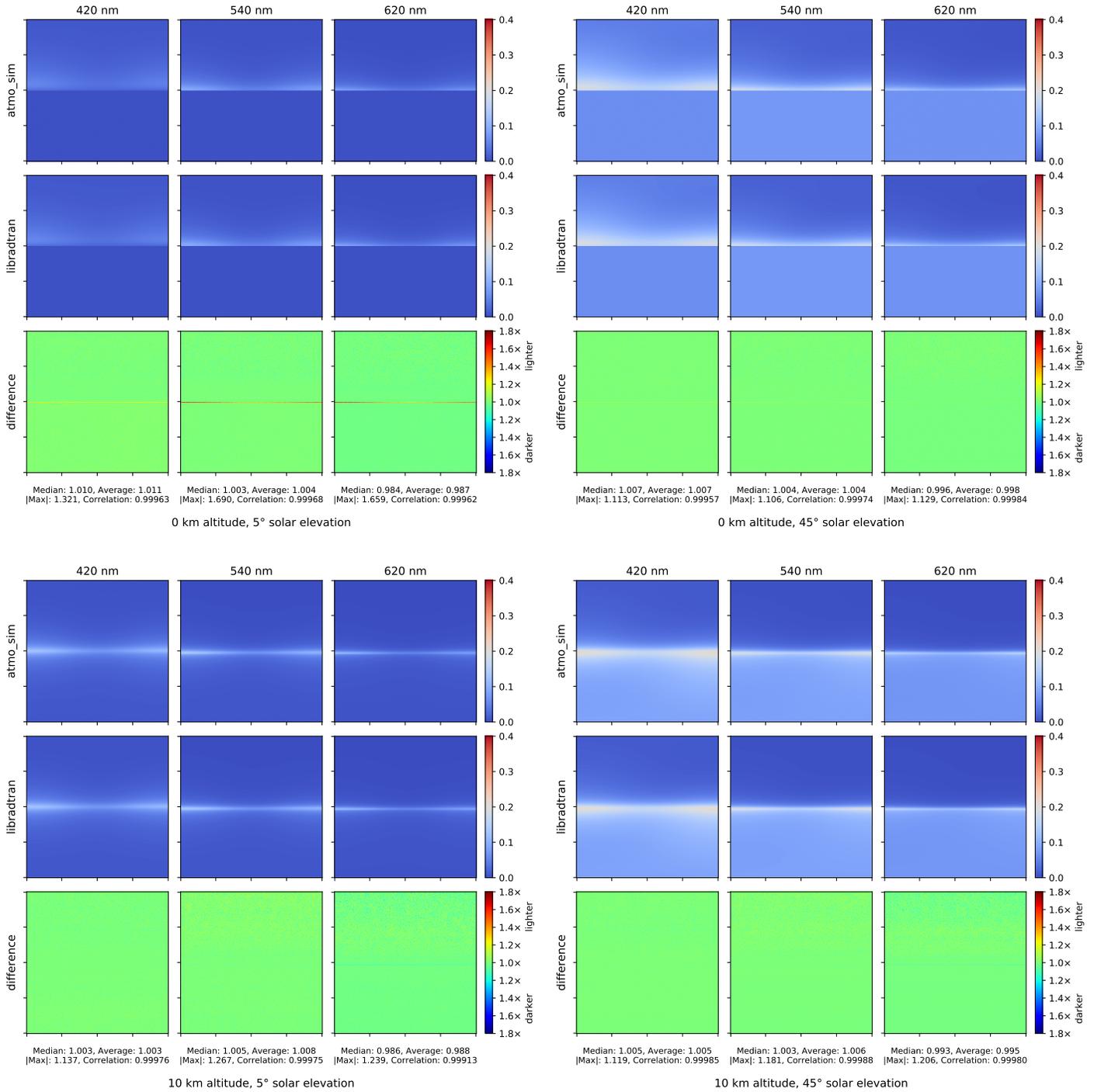


Fig. 6. Atmospheres without any aerosols at 0 km and 10 km altitudes (top, bottom) with solar elevations 5° and 45° (left, right). Notice that the radiances simulated by `atm_sim` are comparable to `libradtran` without any noticeable error patterns. With increasing wavelengths, our atmospheric model produces slightly darker images, which only happens with an O<sub>3</sub> layer and is most likely due to a different absorption cross section. The error lines at horizons are discussed in the text. **Notation and scale:** The top two rows of every comparison show radiances with a color scale in  $W \cdot sr^{-1} \cdot m^{-2}$  per wavelengths 420 nm, 540 nm and 620 nm separately in the three columns. The difference images are acquired by dividing `atm_sim` radiances by `libradtran` radiances per pixel. The color scale is normalised to show brighter and darker areas with an equivalent weight. The median, average and maximum absolute errors are computed directly from the per-pixel divided radiances. The signal correlation coefficients are computed from the radiances flattened to a 1D array.

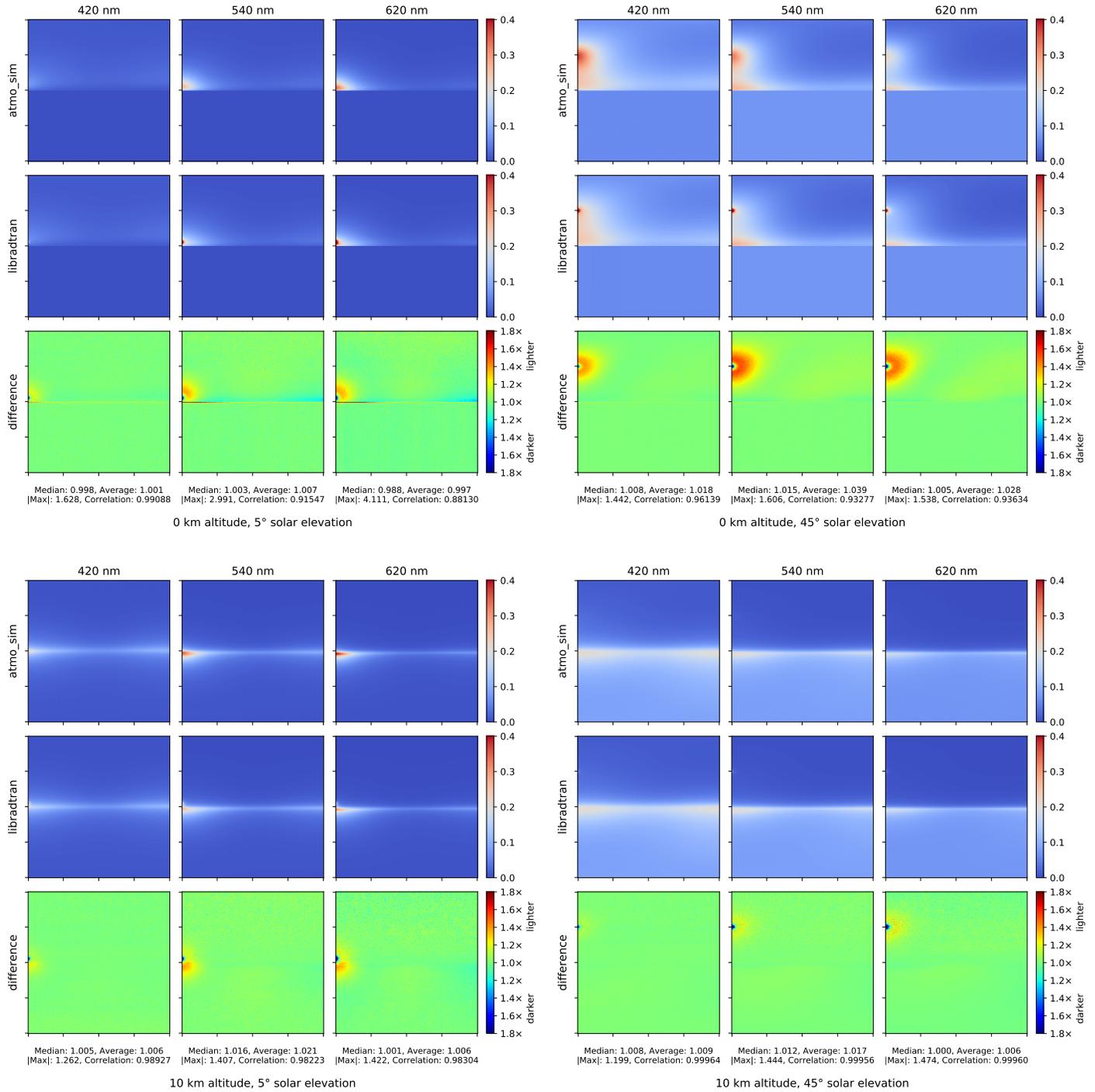


Fig. 7. Similar as Fig. 6, but this time with INSO, WASO and SOOT aerosols from OPAC corresponding to a *continental average* concentration distribution from libradtran. Notice that the color scales are the same as in Figs. 5 and 6, which allows us to see that errors appeared because of differently modelled aerosol phase functions, which we discuss in the text. The errors are more pronounced around the solar disk due to our INSO phase function being less forward scattering. **Notation and scale:** See caption of Fig. 6, where the image notations and scales are explained.



Fig. 8. A low inversion layer on a clear autumn day in Central Europe. These photos were taken seconds apart from on board of a glider that was flying at about 1000m above ground level. Fig. 9 shows the corresponding ground view after the flight. Note that the haze layer with its tops at around 500 m above ground level was homogeneous across the entire region: the strong forward scattering coming from the solar direction, and the comparative transparency in the antisolar direction, moved with the aircraft viewpoint - the haze was not actually denser in the direction of the sun. Above such an inversion layer, the viewing conditions are typically exceptionally clear: the mountaintops in the lowest image are between 100 km and 150 km away. Please note that this is an example of a fairly sharp inversion layer transition at a lower altitude than the OPAC atmospheres we use in our model.

### 5.1 Fit Quality Analysis

One in eight of all images (every other albedo, elevation, altitude) were selected for quality control of the obtained fit. This was done via a comparison between the rendered references and renderings that use the fit. This included a manual check for artefacts, but also a systematic SNR analysis. For all images, the minimum SNR was 14.35, the maximum 34.15, and the mean 28.52. The lowest SNR values were obtained for low solar elevations and high observer altitudes, where there is a narrow bright orange wedge on one side of the horizon: in this setting, the added horizon blur causes the most damage. Samples of the automatically and systematically generated comparison images which were manually viewed to check for artefacts can be found in this supplement as EXRs: a tone-mapped sample is shown in Fig. 10.

We also analysed the “end-to-end” error incurred by the whole process we are using. A summary of our findings is that the following components affect the end result:

- Noise in the reference images: this has some effect, but is limited.
- The direct error incurred by the fit: this is the main source of error.
- Error incurred by dataset compression: so far, this is just the inaccuracies introduced by the conversion from double (all our computations are performed as doubles) to half. This introduces some error but the net effect of it is still negligible.
- Noise in the renderings using the fit: this proved to be negligible as well.

### 5.2 Interpolation Quality Analysis

We also performed a systematic analysis of the error incurred by the interpolation between data points which were provided during the fit. That is, e.g. how far the radiance patterns diverge from the true solution for solar elevations between the ones that were used for the fitting. For a small subset (6) of altitudes in the middle of each elevation interval, 3 images were compared: brute force (B), fit (F), and interpolation (I) from the fits at interval borders. The maximum

of ratio  $\text{rmse}(B, I) / \text{rmse}(B, F)$  was 1.52 (1 at interval borders), which we deemed to be acceptable.



Fig. 9. This photo was taken  $\approx 30$  minutes after the in-flight images shown in Fig. 8. No atmospheric layering is visible from the ground: the inversion layer does not produce any noticeable brighter stripe along the horizon. However, noticeable ground haze for horizontal viewing directions can be observed: the trees and buildings at the edge of the airfield are less than 2 km away. Contrast this with the 100–150 km viewing distances seen towards the mountains in the third image of Fig. 8.

## 6 PLAUSIBILITY OF THE ATMOSPHERIC CONFIGURATION

As discussed in Sec. 4.2 of the paper, we use scatterer profiles provided by OPAC to model our atmosphere: and the vertical profiles defined by OPAC exhibits a distinct lower haze layer. Here, we provide anecdotal real-life imagery of what a similar configuration looks from higher observer altitudes. Fig. 8 and Fig. 9 show photographs taken on a fairly typical clear autumn day in Central Europe during

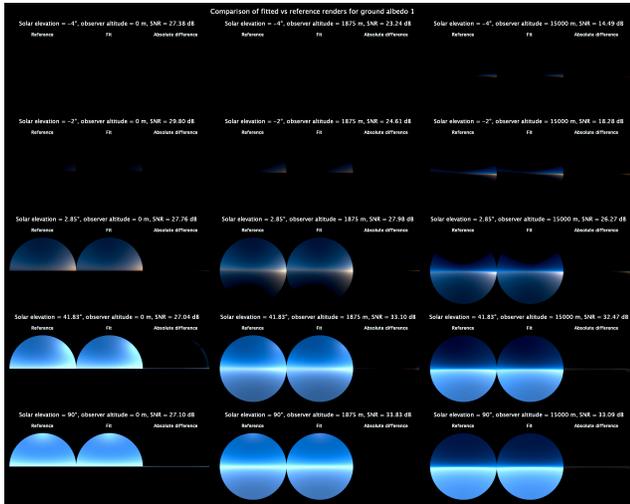


Fig. 10. A sample of the kind of comparison image which were systematically generated to evaluate the model fit. Please disregard the tone mapping artefacts seen in some of the side facing fish eye views: and please refer to the EXR version (which is also supplied as part of the supplemental materials) of this HDR image to get a good impression of its contents.

high pressure weather. In this region, the presence of a marked, hazy inversion layer that can be seen in these images is typical for not just autumn days, but generally high pressure scenarios where the atmospheric layering is so stable that no cloud-forming convection can start. On such days, clear, cloud-free skies can be expected from dawn to dusk: in other words, exactly the conditions that a clear sky model attempts to represent. The only change during such a day is usually a gradual rise of the inversion layer during the course of the day, and a more or less pronounced increase in turbidity: both are due to residual convection within the inversion layer. Compare the photos with the results of our model, e.g. those shown in Fig. 19.

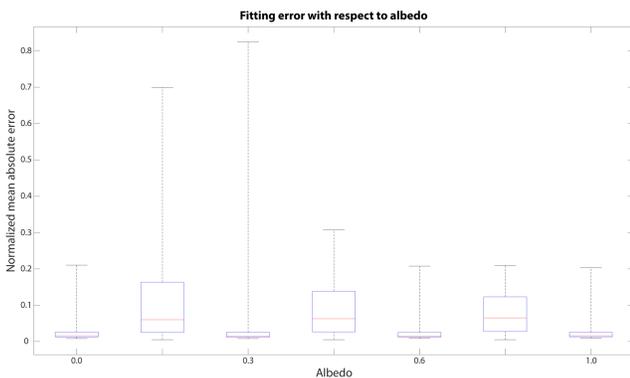


Fig. 11. A box plot of the normalised mean absolute errors for the albedos covered by our model. The logic of the plot follows that discussed in the caption of Fig. 12.

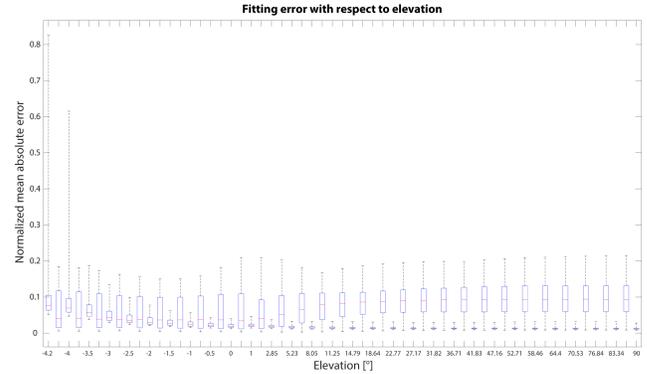


Fig. 12. A box plot of the normalised mean absolute errors for the solar elevations covered by our model. The red line is the median, the blue box goes from the first to the third quartile, and the whiskers are the minimum and maximum values. The labeled values represent solar elevations for which reference images were computed in the brute force dataset, and the errors shown there are between the fit and those images. For these values, it can be observed that as expected, for elevations with brute force reference images, the error increases with decreasing solar elevation due to the higher noise levels in these images.

In between the labeled values, the interpolation error had to be estimated, as intermediate reference images were generally not available. The estimate is the difference between the two neighbouring fits: while this is a loose bound on the true interpolation error, it can best be interpreted as "how wrong could one get if one did not interpolate at all", and not as the actual interpolation error. If this difference-based estimate was low, it would mean there is no point in using the non-trivial image interpolation scheme proposed in the paper, because normal pixel-wise interpolation would already work well. This, in turn, means that the sometimes quite large interpolation error seen here is not automatically a bad thing: if it were too low, the proposed interpolation scheme would be pointless.

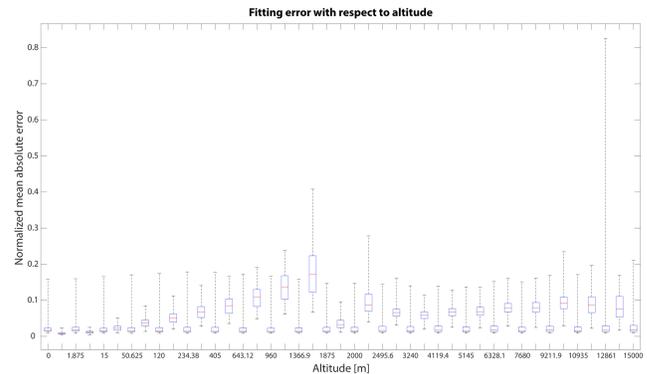


Fig. 13. A box plot of the normalised mean absolute errors for the observer altitudes covered by our model. The logic of the plot follows that discussed in the caption of Fig. 12.

## 7 COMPARISON TO OTHER SKY MODELS

In Sec. 6.5 of the paper we discuss limitations of models based on exponential scatterer distributions and show a sample comparison between one such approach and our OPAC-based model for a

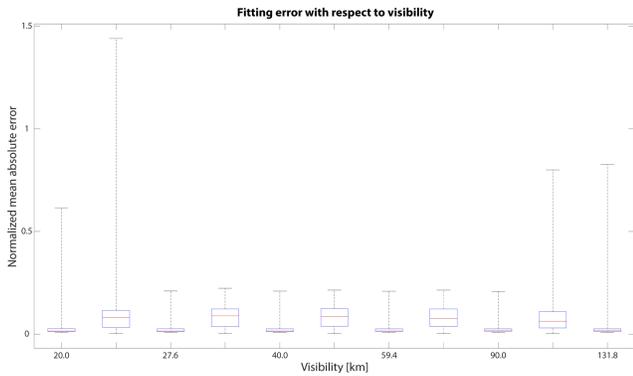


Fig. 14. A box plot of the normalised mean absolute errors for the visibilities covered by our model. The logic of the plot follows that discussed in the caption of Fig. 12.

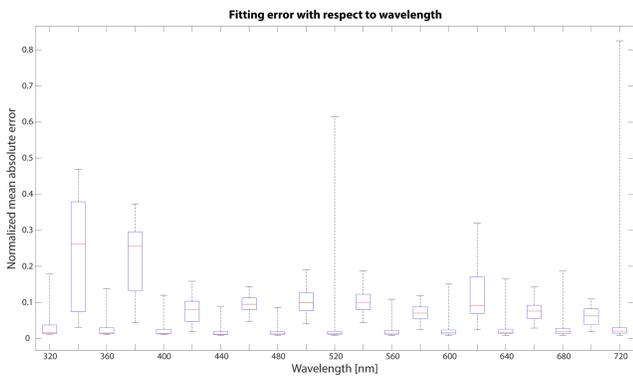


Fig. 15. A box plot of the normalised mean absolute errors for the wavelengths covered by our model. The logic of the plot follows that discussed in the caption of Fig. 12.

hazy atmosphere. Here we provide some additional comparisons. In Fig. 17 we place reference renderings of our atmosphere next to three examples of exponential scatterer distributions in a clearer atmosphere. The first exponential profile (second column) is constructed to match our OPAC-based profile below the inversion layer. As such it results in much hazier atmosphere. On the other hand, the second profile (third column) matching ours above the inversion layer makes the atmosphere much clearer as the inversion layer is completely missing there. Finally, the third exponential profile (fourth column) is designed to give the same ground level visibility and vertical turbidity as our atmosphere. Conclusion in this case is the same as in the paper - the exponential profile gives similar results at lower altitudes but diverges as altitude increases.

To verify this observation we provide a direct comparison to two existing sky models that are based on exponential scatterer distributions: the model by Hillaire [2020] and the one by Bruneton [2016]. For this, we used source code provided by Hillaire which implements both his and Bruneton’s model. We set the input parameters so as the used extinction coefficients were the same as in our model, and the used exponential aerosol profile resulted in the same ground

level visibility and vertical turbidity. As expected, both models then provide output very close to renders of the similarly constructed exponential profile presented in Fig. 17 (the rightmost column). Therefore, in Fig. 18 we can observe a good match between all three models at 2 m. But with increasing observer altitude, differences between our OPAC-based atmosphere and the two exponential ones start to manifest themselves.



Fig. 16. A close-up of one of the column bases of the scene shown in Fig. 14 in the paper and Fig. 20 in this supplement, with an Airbus A380 airliner placed next to it for scale. This intends to give a sense of the massive extent of these scenes: as they show a somewhat sterile “atmospheric debugging” set-up, it is easy to under-estimate how large everything seen in them is.

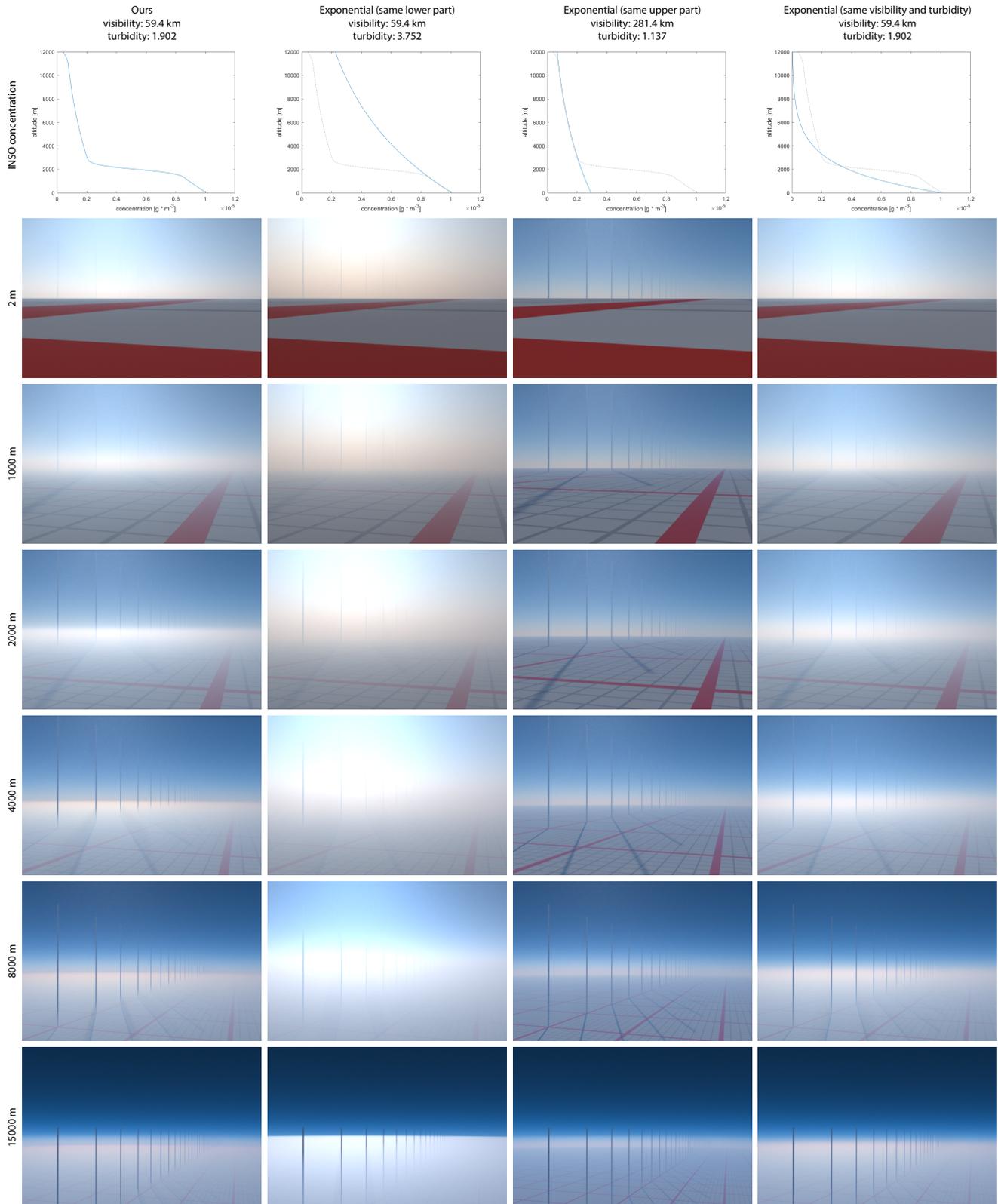


Fig. 17. A comparison of our OPAC-based atmosphere scatterer profile versus three examples of purely exponential ones.

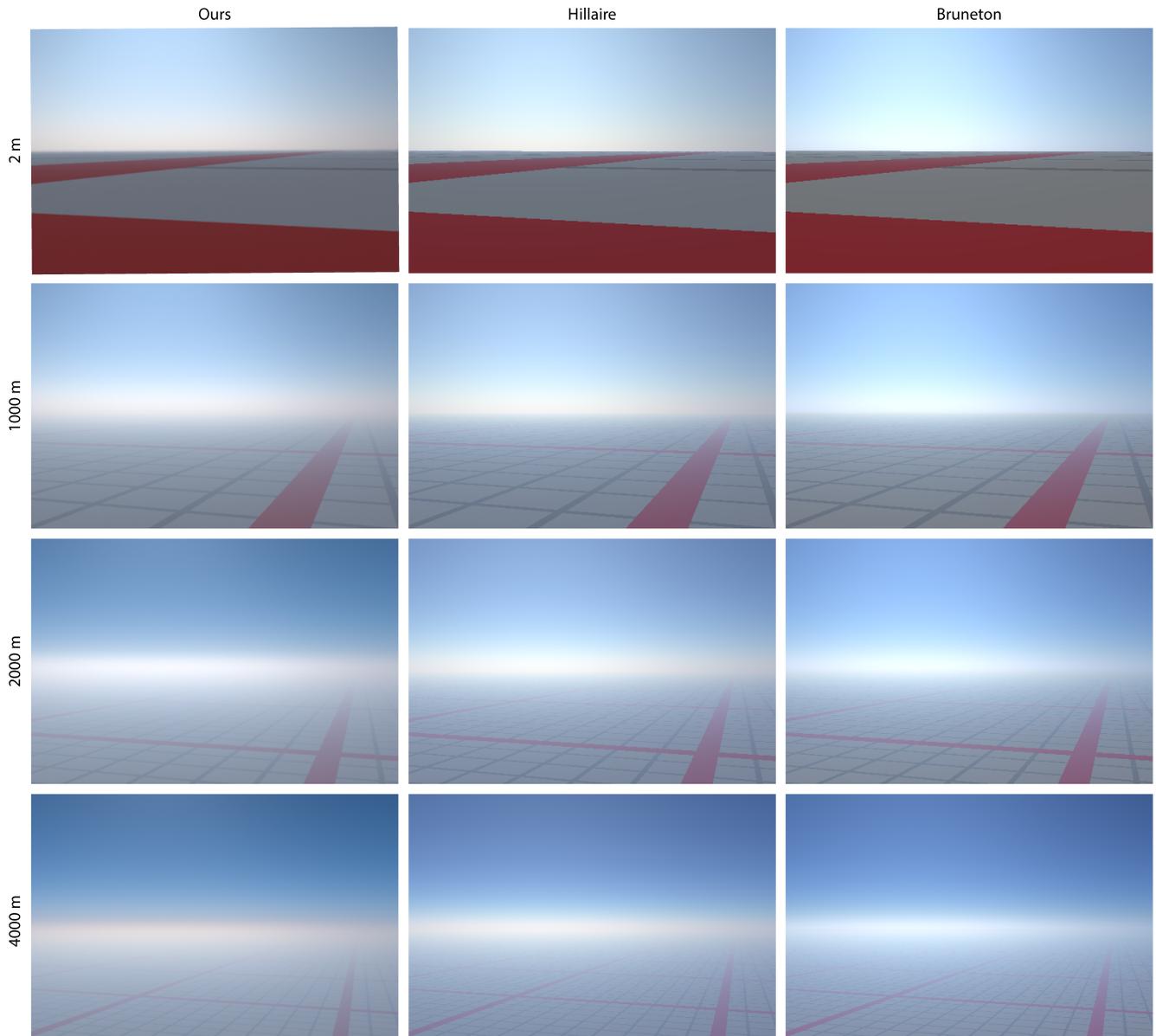


Fig. 18. A comparison of our fitted model with two other related works by Hillaire [2020] and Bruneton [2016].

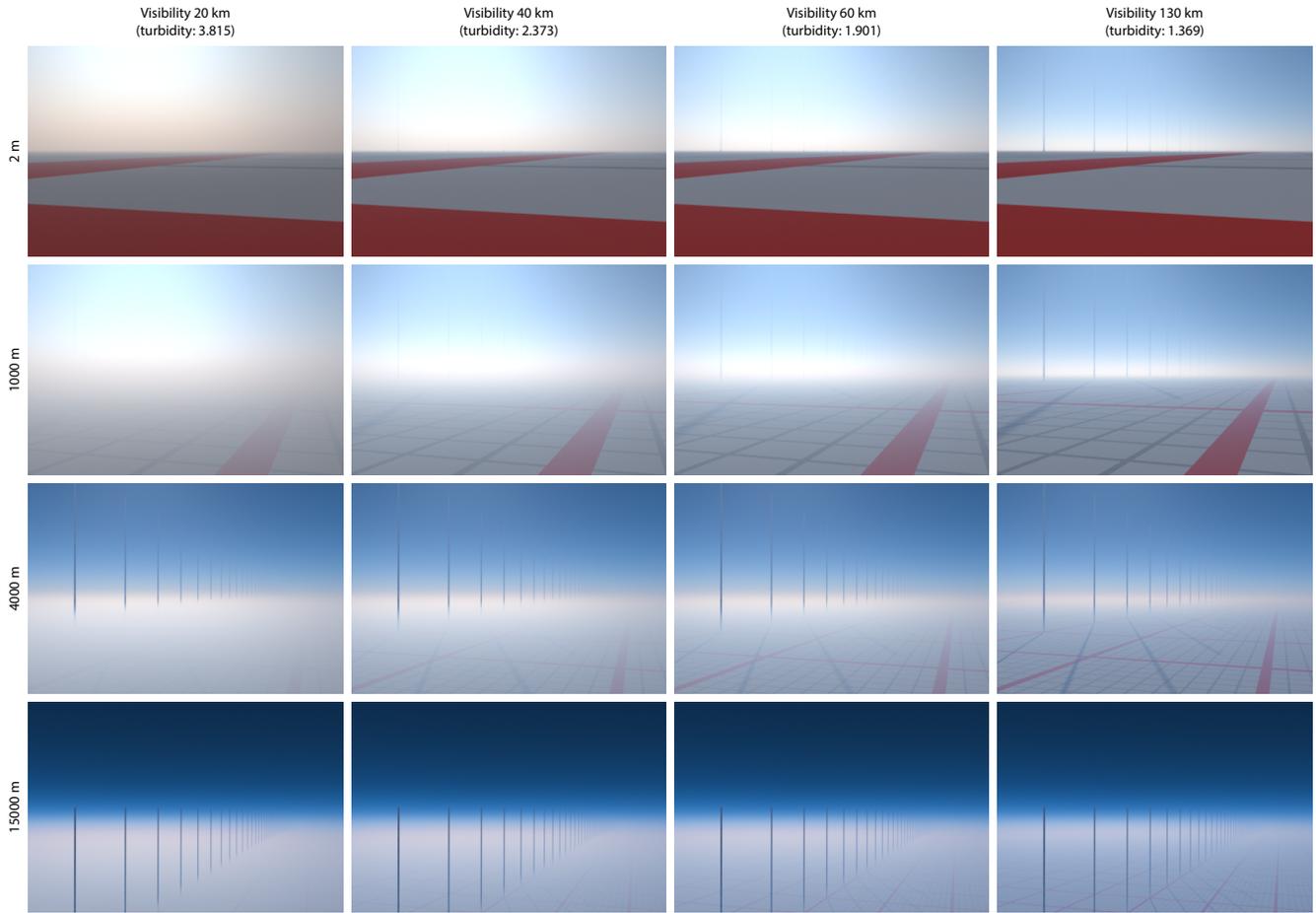


Fig. 19. Our model is capable of rendering ground level visibilities in range between 20km and 131.8km. Here is an example of 4 of them: note how the inversion layer becomes more and more transparent for higher visibility ranges.

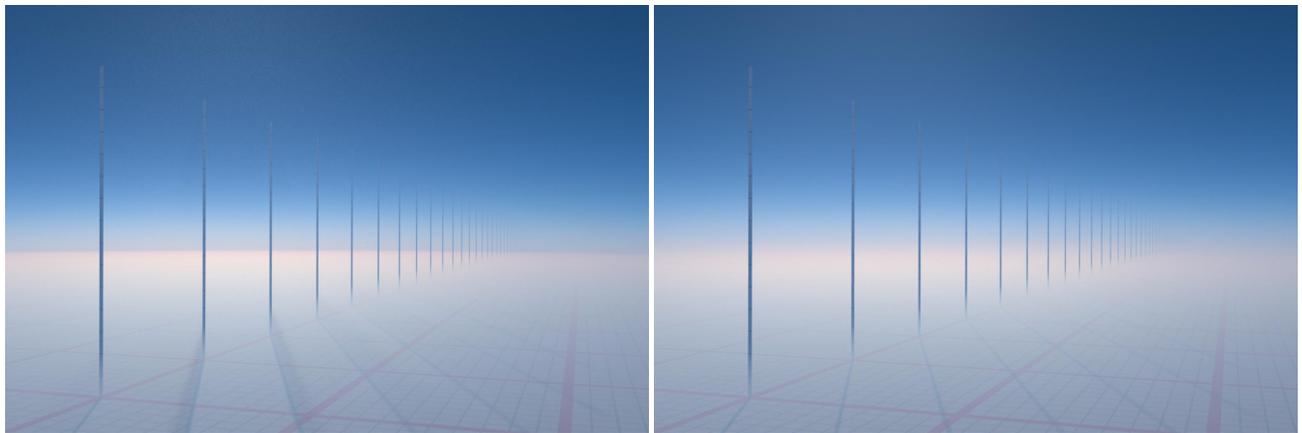


Fig. 20. A viewing geometry for which the limitations of just using a pre-computed sky dome radiance model become noticeable. **Left:** a reference rendering done with `atmo_sim`, observer altitude 8km, looking towards the sun, which is at  $16^\circ$  elevation, and outside the frame. **Right:** the same scene rendered purely by evaluation of the pre-computed model. Note the absence of volumetric shadows in the ground haze layer. However, a good approximation for these shadows could comparatively easily be computed by simple ray marching, which would still be a lot cheaper than a full path tracing solution. Rendering times are 10+ hours for the not fully converged reference, and 4.5 minutes for the model-based rendering.

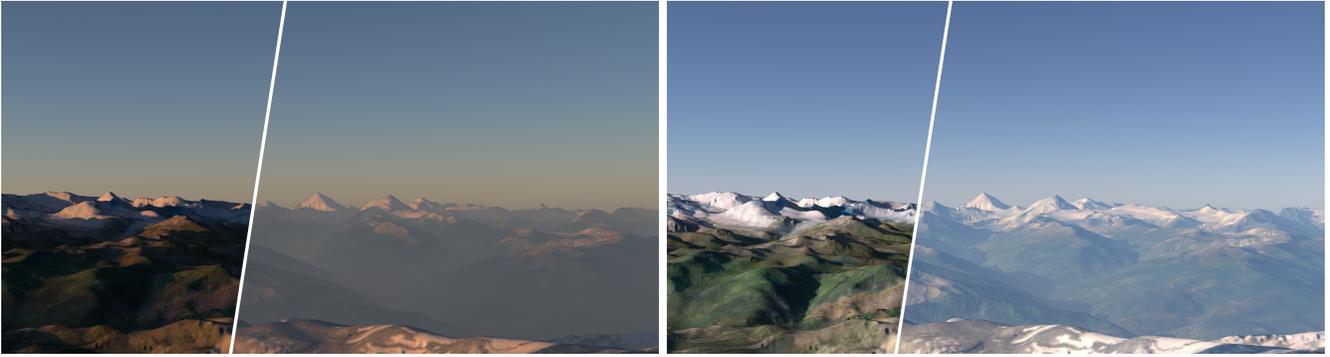


Fig. 21. The two mountain landscape images from the teaser with in-scattered light for finite viewing distances removed.

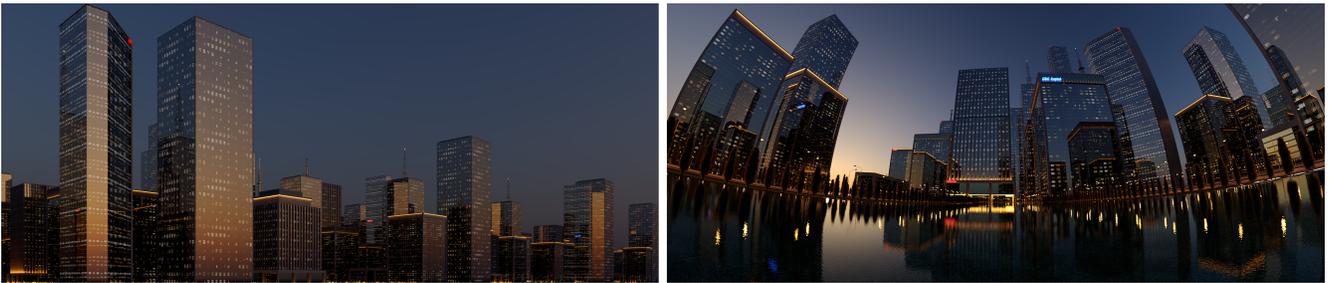


Fig. 22. Two more examples rendered using our model in a production renderer, for post-sunset conditions.

## REFERENCES

- Barry A. Bodhaine, Norman B. Wood, Ellsworth G. Dutton, and James R. Slusser. 1999. On Rayleigh Optical Depth Calculations. *Journal of Atmospheric and Oceanic Technology* 16, 11 (1999), 1854–1861. [https://doi.org/10.1175/1520-0426\(1999\)016<1854:ORODC>2.0.CO;2](https://doi.org/10.1175/1520-0426(1999)016<1854:ORODC>2.0.CO;2) arXiv:[https://doi.org/10.1175/1520-0426\(1999\)016<1854:ORODC>2.0.CO;2](https://doi.org/10.1175/1520-0426(1999)016<1854:ORODC>2.0.CO;2)
- Eric Bruneton. 2016. A qualitative and quantitative evaluation of 8 clear sky models. *IEEE transactions on visualization and computer graphics* 23, 12 (2016), 2641–2655.
- C. Emde, R. Buras-Schnell, A. Kylling, B. Mayer, J. Gasteiger, U. Hamann, J. Kylling, B. Richter, C. Pause, T. Dowling, and L. Bugliaro. 2016. The libRadtran software package for radiative transfer calculations (version 2.0.1). *Geoscientific Model Development* 9, 5 (2016), 1647–1672. <https://doi.org/10.5194/gmd-9-1647-2016>
- V. Gorshelev, A. Serdyuchenko, M. Weber, W. Chehade, and J. P. Burrows. 2014. High spectral resolution ozone absorption cross-sections – Part 1: Measurements, data analysis and comparison with previous measurements around 293 K. *Atmospheric Measurement Techniques* 7, 2 (2014), 609–624. <https://doi.org/10.5194/amt-7-609-2014>
- L. G. Henney and J. L. Greenstein. 1941. Diffuse radiation in the Galaxy. *Astrophysical Journal* 93 (Jan 1941), 70–83. <https://doi.org/10.1086/144246>
- M. Hess, P. Koepke, and I. Schult. 1998. Optical Properties of Aerosols and Clouds: The Software Package OPAC. *Bulletin of the American Meteorological Society* 79, 5 (1998), 831–844. [https://doi.org/10.1175/1520-0477\(1998\)079<0831:OPOAAC>2.0.CO;2](https://doi.org/10.1175/1520-0477(1998)079<0831:OPOAAC>2.0.CO;2) arXiv:[https://doi.org/10.1175/1520-0477\(1998\)079<0831:OPOAAC>2.0.CO;2](https://doi.org/10.1175/1520-0477(1998)079<0831:OPOAAC>2.0.CO;2)
- Sébastien Hillaire. 2020. A Scalable and Production Ready Sky and Atmosphere Rendering Technique. *Comput. Graph. Forum* 39, 4 (2020), 13–22.
- Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. 2019. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6927–6935.
- T. Kolda and B. Bader. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (2009), 455–500.
- A. Kreuter and M. Blumthaler. 2013. Feasibility of polarized all-sky imaging for aerosol characterization. *Atmospheric Measurement Techniques* 6, 7 (2013), 1845–1854. <https://doi.org/10.5194/amt-6-1845-2013>
- B. Mayer. 2009. Radiative transfer in the cloudy atmosphere. *EPJ Web of Conferences* 1 (2009), 75–99. <https://doi.org/10.1140/epjconf/e2009-00912-1>
- R. Perez, R. Seals, and J. Michalsky. 1993. All-weather model for sky luminance distribution—Preliminary configuration and validation. *Solar Energy* 50, 3 (1993), 235–245. [https://doi.org/DOI:10.1016/0038-092X\(93\)90017-1](https://doi.org/DOI:10.1016/0038-092X(93)90017-1)
- Pınar Satılmış, Thomas Bashford-Rogers, Alan Chalmers, and Kurt Debattista. 2016. A machine-learning-driven sky model. *IEEE computer graphics and applications* 37, 1 (2016), 80–91.
- Alexander Wilkie. 2018. The Advanced Rendering Toolkit. <http://cgg.mff.cuni.cz/ART>.
- Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-François Lalonde. 2019. All-weather deep outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10158–10166.