Check for updates

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2023MS003797

A Stereo Camera Simulator for Large-Eddy Simulations of Continental Shallow Cumulus Clouds Based on Three-Dimensional Path-Tracing

Yannick Burchart[1], Christoph Beekmans[2], and Roel Neggers[1]

[1]Institute for Geophysics and Meteorology, University of Cologne, Cologne, Germany, [2]Meteorological Institute, University of Bonn, Bonn, Germany

Key Points:

- Pairs of hemispheric ray-tracing renderings of shallow cumulus fields in large-eddy simulations are used to reconstruct cloud boundaries
- The method robustly reproduces close to 70% of the visible cloud boundaries
- The results provide proof of principle for its use as a stereo camera simulator for evaluating convective clouds in high-resolution models

Correspondence to:
Y. Burchart,
yburchar@uni-koeln.de

Citation:
Burchart, Y., Beekmans, C., & Neggers, R. (2024). A stereo camera simulator for large-eddy simulations of continental shallow cumulus clouds based on three-dimensional path-tracing. *Journal of Advances in Modeling Earth Systems*, *16*, e2023MS003797. https://doi.org/10.1029/2023MS003797

Received 6 JUN 2023
Accepted 15 JAN 2024

Author Contributions:
Conceptualization: Yannick Burchart, Roel Neggers
Data curation: Yannick Burchart
Formal analysis: Yannick Burchart
Funding acquisition: Roel Neggers
Investigation: Yannick Burchart, Roel Neggers
Methodology: Yannick Burchart, Christoph Beekmans, Roel Neggers
Project administration: Roel Neggers
Resources: Yannick Burchart, Roel Neggers
Software: Yannick Burchart, Christoph Beekmans, Roel Neggers
Supervision: Roel Neggers
Validation: Yannick Burchart, Roel Neggers
Visualization: Yannick Burchart
Writing – original draft: Yannick Burchart

© 2024 The Authors. Journal of Advances in Modeling Earth Systems published by Wiley Periodicals LLC on behalf of American Geophysical Union.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Abstract The complex spatial and temporal structure of cumulus clouds complicates their representation in weather and climate models. Classic meteorological instrumentation struggles to fully capture these features. Networks of multiple high-resolution hemispheric cameras are increasingly used to fill this data gap, and provide information on this missing multi-dimensional spatial information. In this study, a path-tracing algorithm is used to generate virtual camera images of resolved clouds in large-eddy simulations (LES). These images are then used as a camera network simulator, allowing reconstructions of three-dimensional cloud edges from the model output. Because the actual LES cloud field is fully known, the combined path-tracing and reconstruction method can be statistically analyzed. The method is applied to LES realizations of summertime shallow cumulus at the Jülich Observatory for Cloud Evolution (JOYCE), Germany, which also routinely operates a camera network. We find that the path-tracing method allows accurate reconstruction of up to 70% of the visible cloud edges. Additional sensitivity tests show that the method is robust for changes in its hyperparameters. The sensitivity to cloud optical thickness is also investigated, finding a cloud boundary placement error of approximately 182 m. This error can be considered typical for cloud boundary reconstruction using real stereo camera imagery. The results provide proof of principle for future use of the method for evaluating LES clouds against camera network imagery, and for further optimizing the configuration of such camera networks.

Plain Language Summary We propose an algorithm to make camera images from clouds as simulated on large and fast computers. The algorithm relies on so-called path- or ray-tracing techniques as often applied in modern gaming or computer-generated imagery in movies. This method not only yields beautiful cloud visualizations, it also enables the emulation of scientific camera networks such as the one currently operational at the Jülich Observatory of Cloud Evolution (JOYCE) in west Germany. From the images, three-dimensional structures of the simulated cloud fields can be reconstructed. This technique allows the scientific evaluation of the model clouds against images from real camera networks. This would mean a significant step forward from the use of common vertically pointing meteorological instruments such as lidar or radar. However, before such an effort is undertaken, the accuracy of the method still needs to be determined. For this purpose, the cloud edges, or outer hulls, are reconstructed for selected days in summer. These reconstructed edges are then compared to the original clouds in the simulation. We find that the method satisfactorily reproduces the high-resolution geometry of clouds, providing proof of principle for this approach and opening a new way of comparing high-resolution meteorological models with camera measurements.

1. Introduction

Cumulus clouds (Cu) play a crucial role in Earth's weather and climate, by globally impacting radiative transfer and vertically transporting heat, moisture and momentum. This makes their accurate representation in general circulation models of crucial importance (Cesana & Del Genio, 2021; Myers et al., 2021; Neggers, Neelin, & Stevens, 2007; Tiedtke, 1989). The complex spatial geometry of Cu is a key factor in most of these impacts. Until recently, most observations of Cu geometry relied on one-dimensional, vertically pointing remote sensing instruments such as lidars or radars (Fielding et al., 2020; Schulz et al., 2021; Zhan et al., 2021). While satellite imagery can also provide information on Cu geometry, these data are usually limited to vertical projections, they still miss the fine-scale structure on the meter scale, and often lack the temporal resolution to study cloud evolution (Fielding et al., 2020; Schulz et al., 2021). Large-Eddy Simulations (LES) can be used to supplement the

Writing – review & editing:
Yannick Burchart, Christoph Beekmans, Roel Neggers

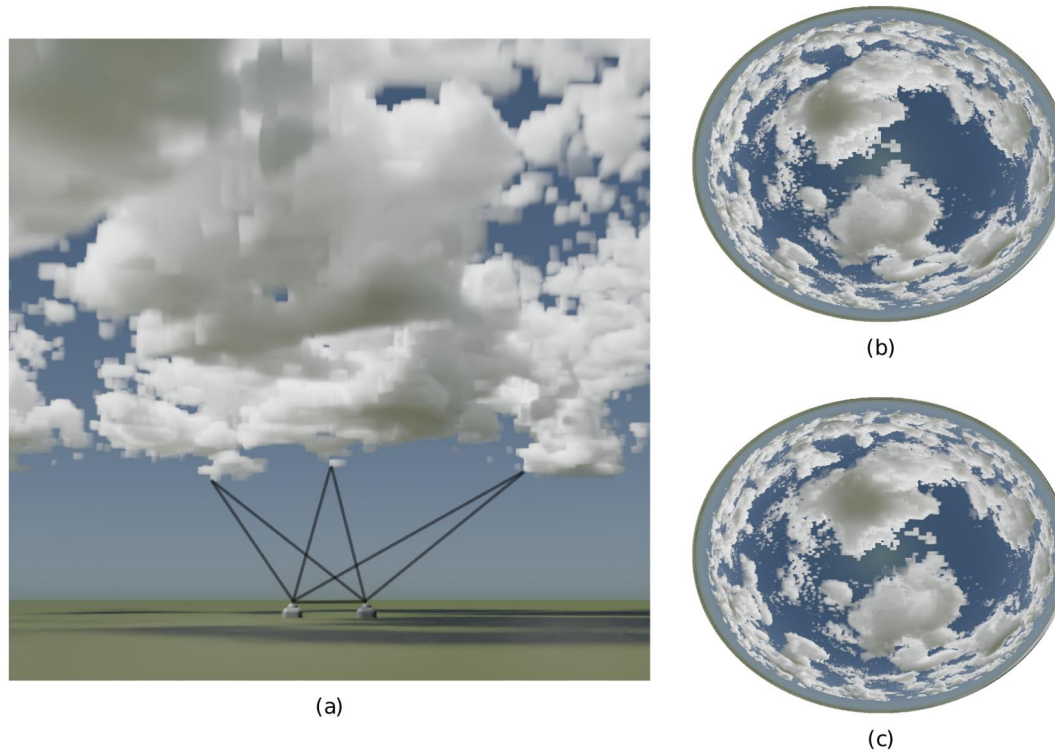

Figure 1. (a) Example of a path-tracing rendering of a cumulus cloud (Cu) field of an large-eddy simulations resolved at 50 m spatial discretization. The latter can still be recognized in the box-like small cloud elements. A stereo camera pair is illustrated with gray boxes (amplified for visualization). The resulting camera images are shown on the right, with (b) and (c) being the images from the left and right cameras, respectively. Opacity is affected by the liquid water path, while the dark cloud bases also reflect the land surface color. Each camera detects visible features of the clouds from a different location and angle. A three-dimensional cloud field can be reconstructed from this pair of images (black lines panel in (a)).

incomplete observational database on Cu geometry, by providing access to full four-dimensional virtual cloud fields (Angevine et al., 2018; Neggers et al., 2011; Shen et al., 2022; Siebesma & Jonker, 2000). Despite showing great promise LES is still only a model, suffering from known shortcomings which do affect resolved clouds.

An alternative way of observing three-dimensional Cu geometry relies on photogrammetry. Observing Cu simultaneously by two or more cameras enables the reconstruction of their three-dimensional information by triangulation (see Section 2.3) or space carving (see Nouri et al., 2019; Veikherman et al., 2015) techniques. The first photogrammetric reconstruction of clouds dates back to Malkus and Ronne (1954). They manually reconstructed four eye-selected cloud points of photographic films from two cameras to estimate oceanic trade-wind cumulonimbus tower heights. Nowadays, most cameras consist of all-sky-imagers since they cover a larger field of view and hence allow better, more complete reconstructions (Beekmans et al., 2016; Crispel & Roberts, 2018; Nguyen & Kleissl, 2014).

A relatively recent development involve using networks of multiple high-resolution cameras to observe clouds in unprecedented spatial detail and high frequency (few seconds) (Blum et al., 2021; Nouri et al., 2019; Romps & Öktem, 2018). While results with such networks are encouraging, careful assessment of the method is still needed. Offline-tests on virtual cloud objects have been performed, but these typically lack the fine-scale geometry of real cloud fields. To fill this gap, this study combines this technique with LES. More precisely, our goal is to apply the cloud reconstruction technique to pairs of hemispheric images taken of LES cloud fields which are resolved at high resolutions. To this purpose a "simulator" is needed; an algorithm for observing the model clouds in exactly the same way as a real hemispheric camera would. In this study a path-tracing algorithm is used to generate virtual camera projections (see Figure 1 for an example). This in principle allows emulating networks of multiple stereo all-sky-imagers, as are now operational at various meteorological sites (Beekmans et al., 2016; Crispel & Roberts, 2018; Nguyen & Kleissl, 2014). Various recent studies have applied Ray Tracing (RT) techniques in

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

combination with LES (Veerman et al., 2022; Villefranque et al., 2019), but not yet for generating hemispheric projections.

Applying the stereo camera reconstruction technique to LES cloud fields creates a few interesting new research opportunities. First, the cloud field reconstructed from the virtual cloud images can be compared to the fully known cloud field in the LES. This comparison allows to statistically analyze the general skill and accuracy of the reconstruction method. Second, it might yield new insights into the optimal configuration of real camera networks for observing Cu. Finally, when proven successful, this method could open up a new pathway of evaluating resolved Cu geometry in high-resolution models against observational data.

A key open question still to be answered is if a path-tracing image simulator can generate images adequately to work well with the reconstruction algorithm. This encompasses a range of questions:

- Is it generally possible to reconstruct cloud shells from rendered images?
- If feasible, can a substantial portion of the cloud field be reconstructed?
- Are there any inherent additional errors?
- Does the stability of the reconstruction method vary under specific hyperparameters?

In practice, we make use of high-resolution LES realizations of a selection of recent diurnal cycles of summertime Shallow Cumulus clouds (ShCu) as observed at the Jülich Observatory for Cloud Evolution (JOYCE), Germany (JOYCE) (Löhnert et al., 2015). Path-tracing images are generated based on the high-frequency output of the LES and subjected to the triangulation method. Various statistical analyses are then undertaken to test this method and gain insight. Note that a network of multiple stereo cameras is also operational at this site, installed as part of the SOCLES project (gepris.dfg.de/gepris/projekt/430226822). In this study, no comparisons are yet made with real camera images; this is not needed to achieve the research goals stated above. Evaluation of LES clouds against such observational data using this method is reserved for future, follow-up studies.

This study is structured as follows. In Section 2 the general method is described, including a short overview of the LES model setup (Section 2.1), a description of the path-tracer algorithm (Section 2.2) and the stereo reconstruction algorithm (Section 2.3). Section 3 provides a short overview of the two simulated ShCu-days. Section 4 presents the main results with this method including a statistical evaluation of its performance, while Section 5 tests its sensitivity to various settings. The obtained results are discussed in Section 6, and concluding remarks are provided in Section 7.

2. Method

To generate virtual camera images of resolved clouds in LES a dedicated workflow is adopted, as schematically illustrated in Figure 2. The method consists of three parts: (a) a LES code (Section 2.1), (b) a path-tracer algorithm (Section 2.2), and (c) a cloud field reconstruction algorithm for camera networks (Section 2.3).

2.1. Large-Eddy Simulation (LES)

The high-resolution LES experiments in this study were conducted in the Dutch Atmospheric Large-Eddy Simulation (DALES) model, an established LES code used for a wide range of boundary layers (Heus et al., 2010). DALES has been vigorously tested in various LES intercomparison studies for continental ShCu convection (Brown et al., 2002; Heinze et al., 2017; Roode et al., 2016; Schemann et al., 2020). Its code solves the discretized conservation equations of momentum, energy, humidity and various scalars, assuming the Boussinesq approximation and applying a turbulent kinetic energy model for their sub-scale transport. The version of DALES used here applies doubly periodic horizontal boundary conditions, spatially homogeneous large-scale forcing, and does not include orographic effects. At the surface an interactive (bulk) flux-boundary condition is applied, making use of prescribed skin temperature and roughness lengths.

The experimental configuration applied in this study follows that described in detail by Van Laar et al. (2019). An overview of the defining characteristics is provided in Table 1. Clouds and precipitation are represented using the two-moment mixed-phase microphysics scheme of Seifert and Beheng (2006). The horizontal domain size is 25.6×25.6 km^2, discretized at a horizontal resolution of 50×50 m^2. The model ceiling is at 12.9 km, with the vertical grid spacing increasing with height from 50 m at the surface to 250 m at the ceiling height, similar to Schemann et al. (2020). As described by Van Laar et al. (2019), hourly reanalysis fields from ERA5 (Hersbach

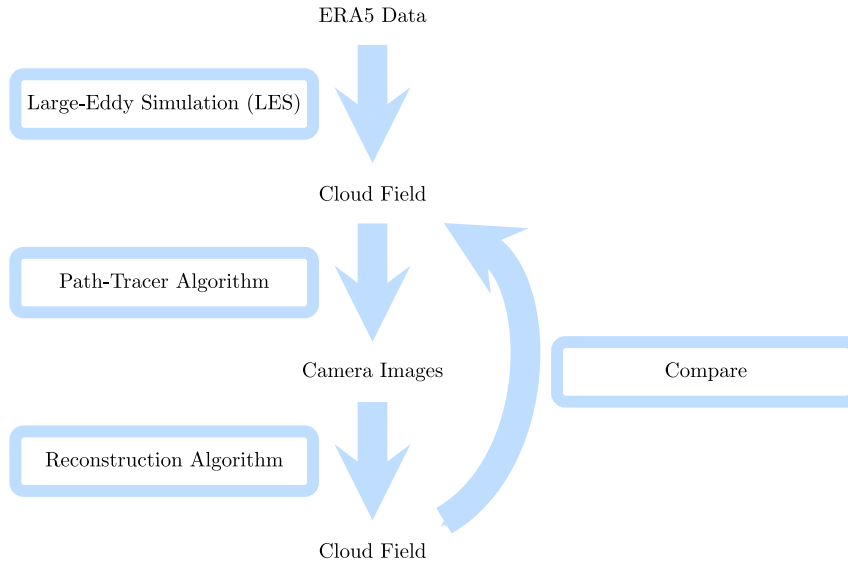

Figure 2. Schematic workflow: The ERA5 data force the large-eddy simulations (LES) (Section 2.1), providing a 3D cloud field at every time step. A path-tracer algorithm (Section 2.2) then generates the virtual camera images. Finally, the initial LES cloud field is reconstructed from these images, similar to using images from a real camera network (Section 2.3). The actual and reconstructed cloud fields are then statistically compared to obtain qualitative information (Section 4).

et al., 2020) are used to calculate the prescribed forcings and boundary conditions. The time- and height varying forcings for wind, temperature and water vapor consist of prescribed profiles of horizontal advective tendencies, subsidence velocity, and horizontal pressure gradients (as expressed through the geostrophic wind). The 3D liquid water content field outputted on a frequency of 30 min to be used as input of our Path-Tracer algorithm described in the next section.

2.2. Path-Tracer Algorithm

To generate the camera images from the LES cloud fields, we follow the method applied by Heus et al. (2021) which makes use of the rendering engine Cycles as included in the open-source 3D computer graphics software Blender (www.blender.org). The idea hereby is to take the liquid water content field from the LES to create cloudy grid boxes in Blender where scattering and absorption depend on the density distribution of the liquid water content. Cycles provides GPU rendering via path-tracing. Path-tracing (first introduced by Kajiya, 1986) is a Monte Carlo method approximating the solution of the rendering equation (Shirley, 1991) which arises from radiative heat transfer theory:

$$L_{out}(x,\omega,\lambda) = L_e(x,\omega,\lambda) + \int_{\omega' \in \Omega} \rho\left(x,\omega',\omega,\lambda\right) L_{in}\left(x,\omega',\lambda\right) \left(\omega' \cdot n_x\right) d\omega', \quad (1)$$

Table 1
Overview of the Large-Eddy Simulations Model Setup

Model	DALES
Number of grid cells x levels	512 × 512 × 144
Horizontal resolution	50 × 50 m^2
Horizontal domain	25.6 × 25.6 km^2
Model top	12.9 km
Horizontal boundary conditions	Doubly periodic
Microphysics parametrization	Seifert and Beheng (2006)
Large-scale forcing	1-hourly ERA5
3D output frequency	30 min

where for a point x, a viewing direction ω and a wavelength λ: L_{out} is the radiance for the outgoing direction, L_e the emitted spectral radiance, ρ the Bidirectional Reflectance Distribution Function (BRDF), L_{in} the radiance for the incoming direction.

The integral in Equation 1 is taken over all incoming directions $\omega' \in \Omega$. The surface normal at x is denoted as n_x. The BRDF is a distribution over the material properties of all objects to be rendered and is defined as

$$\rho(x,\omega_{in},\omega_{out},\lambda) := \frac{dL(x,\omega_{out},\lambda)}{L(x,\omega_{in},\lambda)(\omega_{in} \cdot n_x)\, d\omega_{in}} \quad (2)$$

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Notice that the numerator $dL(x, \omega_{out}, \lambda)$ is the radiance of the surface seen from a point in direction ω_{out}, and the denominator $L(x, \omega_{in}, \lambda)(\omega_{in} \cdot n_x)d\omega_{in}$ is the incident power per unit area.

A Monte Carlo algorithm is applied to solve Equation 1, by shooting probabilistically chosen rays (Kajiya, 1986). In Cycles, the solution of Equation 1 is approximated within the International Commission on Illumination (CIE) XYZ color space, corresponding to the visible light spectrum. This means that instead of solving Equation 1 by sampling over the wavelength spectrum directly, the solution of Equation 1 is approximated by shooting rays of light spectrum saved in the three CIE XYZ color values. To translate between CIE XYZ colors and light spectrum color matching functions are used.

The incoming sunlight is calculated using the constant solar angular diameter of 11.4°, zenith angle of 52.59° and azimuth angle of 77° by setting north aligned with the y-axis. Note that these values are arbitrarily chosen and that their impacts on the results are analyzed in Section 5.1. From this incoming sunlight, the CIE XYZ color values change every time the rays are hitting objects according to their BRDFs. These rays that are hitting the camera lens pixels estimate the color of the corresponding pixel. The final estimation of the pixel colors is done by averaging over all the rays that hit precisely that pixel.

In this procedure, the *render sample* is defined as the number of paths to trace for each pixel. Accordingly, the more samples are taken, the more accurate and less noisy the rendered image becomes. The sensitivity of the results of this study to the render sample was tested, as described in Section 5.

In Blender, the BRDF ρ is defined via sampling the material settings of all the cloud and ground objects in a scene as well as by the parametrization of the atmosphere using the implemented sky model of Hosek and Wilkie (2012). Due to the computational complexity, the scattering and absorption properties are only set for each cloudy grid box in dependence on its liquid water specific humidity instead of considering single cloud droplets. The aerosols are modeled using Linke's turbidity factor and ground albedo is also taken into account (Hosek & Wilkie, 2012). Surface properties over land are modeled using a brown-greenish colored plane. The detailed configurations can be found in the supplementary Blender file, including the hyperparameters used for the sky model of Hosek and Wilkie (2012). Table 2 provides a summary of all the settings employed in the path-tracer algorithm.

We find that setting the cloud volume material to scatter isotropically 97% of the light and absorb 3% works well enough to achieve our main objective, which is primarily to generate hemispheric projections of shallow cumulus clouds that are precise enough to allow stereo reconstruction. Note that these values are arbitrarily chosen, and that their impacts are further analyzed in Section 5.1. Due to memory limitations it is impossible to simulate the full micro-scale physics of the scattering process. To deal with this, the Cycles engine relies on statistical parameterization, making use of a probability density function for the likelihood that a particle inside a grid volume is hit by the light source.

For the all-sky-imager emulation, we used the equidistant fish-eye projection, a focal length of 10.5 mm, a field of view of 180°, and a sensor resolution of 2,456 × 2,054 pixels since that corresponds to our real cameras at JOYCE. After a sensitivity test for the camera distance (not shown), the camera distance was set to 400 m. A full, detailed description of the Blender settings used in this study is provided in Appendix A.

The GPU computation time for rendering the images increases strongly with the number of render samples, as shown in Table 3. To balance the desire for accurate images with the need to limit computation time, we chose to use 50 render samples for the control setup as listed in Table 2. For generating the time analysis shown in Table 3 a single NVIDIA V100 was used, to allow a qualitative time estimation. For the scenario in which we used 50 render samples, the median computation time was about 45 min per image on average, with quite some variation depending on cloud content. By using multiple GPUs the runtime can be reduced drastically, but further analysis of the scaling of the parallelization is needed.

Figures 1b and 1c are examples of renderings generated with these settings. They appear realistic, in that (a) the cloud shapes and coloring seem well reproduced, including bright cloud edges and dark cloud bases, and (b) the hemispheric curvature is clearly recognizable. A careful comparison of the two images shows that they are very similar but not identical, which reflects the different locations of the two cameras. Note that the horizontal resolution of the LES experiment is still detectable in the boxy cloud shapes in the camera snapshots.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 2
Settings of the Path Tracing Algorithm in the Control Configuration

Assumptions	Description
Sky model	Hosek and Wilkie (2012)
Scattering mode	97% isotropically
Absorption mode	3%
Atmospheric turbidity	2.2
Ground albedo	30%
Solar zenith angle	52.59°
Solar angular diameter	11.4°
Solar azimuth angle (setting north along with *y*-axis)	77°
Render sample	50
Minimum transparent bounces	20
Maximum bounces	80
Clamping of direct light	10
Clamping of indirect light	3
Maximum steps through volume	256
Render tile size	128
Camera projection	hemispheric fisheye
Camera distance	400 m
Camera resolution	2,456 × 2,054

Note. More details are provided in the text.

2.3. Stereo Reconstruction Algorithm

To reconstruct the 3D cloud field from the hemispheric camera images generated using the path tracing method described above, the algorithm proposed by Beekmans et al. (2016) is used. This algorithm acts on such hemispheric images, and generates geometrically dense 3D point-data representing the cloud edges. The essentials of the method are briefly described here, for a comprehensive description we refer to Beekmans et al. (2016).

The reconstruction process of a cloudy point P from a camera pair involves multiple steps, as depicted in Figure 3. The first step involves determining the unit vectors x_L ad x_R, which represent the direction toward P in the left and right camera's coordinate system, respectively. These vectors are derived from the internal projection parameters (such as focal length and principal point) and the external orientation (including location and rotation). The epipolar plane is the plane that is spanned by the point P and the two cameras. The angles ψ_L and ψ_R are defined as the rotation of x_L and x_R, respectively, within this epipolar plane. Finally, the location of P in the left camera's coordinate system is computed using the law of sines, which yields:

$$P = s \cdot \vec{x}_L = d \cdot \left(\frac{\cos(\psi_R)}{\sin(\psi_L - \psi_R)} \right) \cdot \vec{x}_L \tag{3}$$

To address the challenge of identifying the same cloud point P in both the left and the right camera image, a series of steps are taken. First the images are rectified, which is necessary for the matching process. Subsequently, a cloud mask is applied to filter out the non-cloudy pixels that are irrelevant to the matching process. Lastly, the Semi-Global Block-Matching algorithm developed by Hirschmuller (2005) is utilized. This algorithm generates a disparity map containing the disparity, or shift distance, of each pixel when transitioning from the left to the right rectified camera image. By utilizing this map, cloud points in one image can be efficiently located in the other image.

Table 3
Computation Time for One Camera Image Rendering. The Computation Was Done on a Single NVIDIA V100 at the Jülich Wizard for European Leadership Science Supercomputer

Render sample	Median time	25%-quartile	75%-quartile
3	192 s	61 s	249 s
10	540 s	120 s	702 s
50	2,684 s	1,382 s	3,366 s
100	5,330 s	2,740 s	6,656 s
256	13,480 s	6,966 s	17,034 s

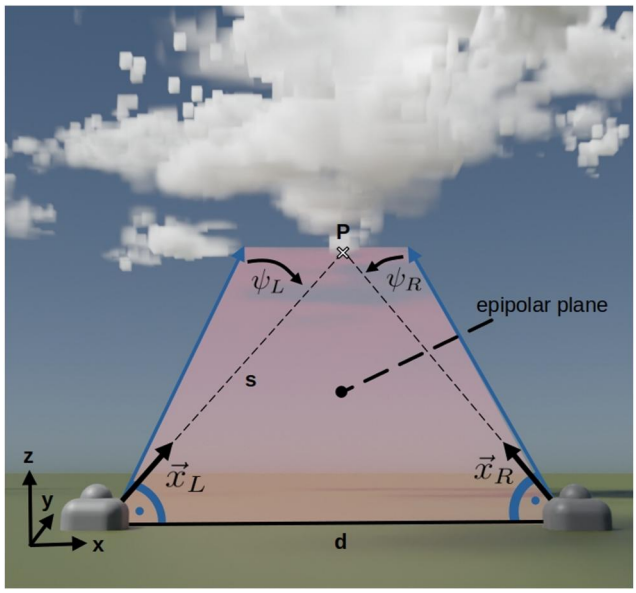

Figure 3. Schematic illustration of the reconstruction calculations for a stereo camera pair setup. In the background, you can see a rendering with our path-tracing algorithm. The visible cloud grid boxes indicate the 50 m horizontal resolution of our simulation. Note that for visualization the cameras are amplified.

3. Simulated Days

To generate the virtual three-dimensional cloud fields required for testing the method, LES experiments are performed for two contiguous non-precipitating ShCu days (21 July 2021 and 22 July 2021) observed at JOYCE (Löhnert et al., 2015) in western Germany. The boundary layer evolution on these 2 days is shown in Figure 4, featuring domain-averaged virtual potential temperature and liquid water specific humidity (q_l). The boundary layer height (z_i), as well as the mixed layer top, are both indicated as red and orange dots, respectively. Here z_i is defined as the height above the minimum in the total liquid water potential temperature flux $\overline{w'\theta_l'}$ (not shown) at which the latter has decreased down to 10% of its peak (negative) amplitude. h_0 is defined as the level of minimal buoyancy flux $\overline{w'\theta_v'}$, following Neggers, Stevens, and Neelin (2007). A separation between heights z_i and h_0 indicate the presence of an active ShCu cloud layer. Compared to 22 July, 21 July featured (a) a larger sky cover (full coverage indicated by the gray shading), (b) more cloud liquid water, (c) a stronger variation in cloud base height, (d) a shallower cloud layer. On 22 July the clouds persisted after sunset (at about 19:00 UTC).

Figure 5 compares vertical profiles of total water specific humidity (q_t) from the LES against radiosonde data, launched every hour on the 2 days as part of an Intensive Observation Period at JOYCE. Four periods are distinguished:

Early Morning:	06:00–08:00 UTC
Late Morning:	09:00–11:00 UTC
Noon:	11:00–13:00 UTC
Afternoon:	12:00–14:00 UTC

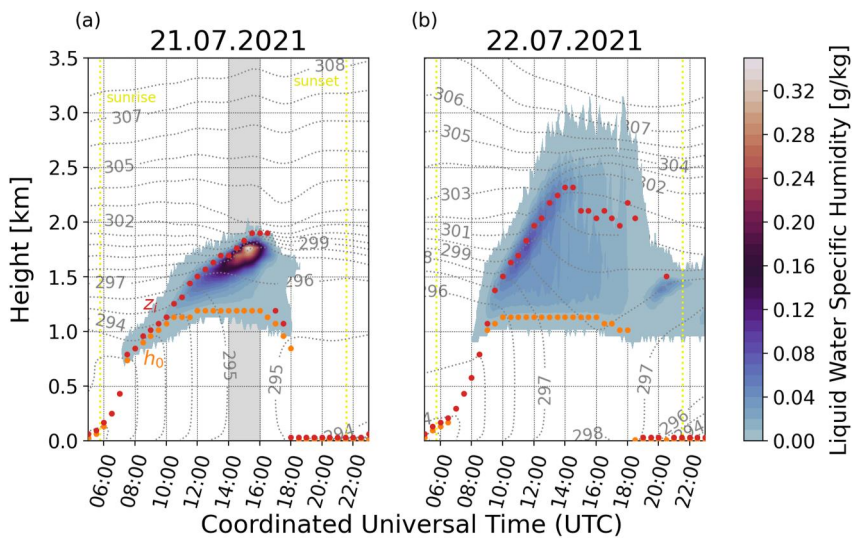

Figure 4. Time height plot of the domain-averaged liquid water specific humidity in the large-eddy simulations realizations of the two selected ShCu days 21 July 2021 (a) and 22 July 2021 (b). The red dots indicate the atmospheric boundary layer top (z_i), and the orange dots are the mixed layer top (h_0). The gray lines indicate the labeled virtual potential temperature in Kelvin. The gray area indicates the period when the sky was fully covered.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

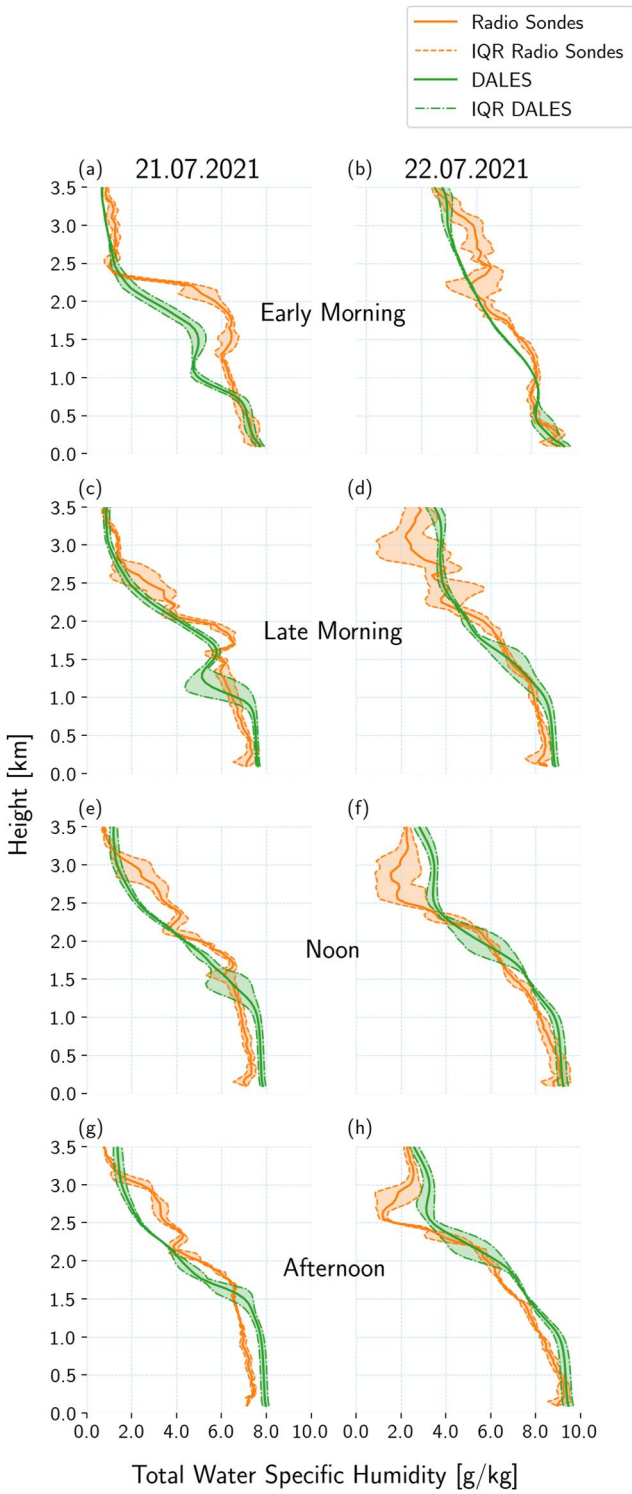

Figure 5. Diurnal Cycle of the averaged total water specific humidity profiles of the selected contiguous ShCu days. The orange line indicates the radiosondes measurements, and the green line the large-eddy simulations simulation. The shaded areas represent the min-max range.

In general the vertical humidity structure is well represented in the simulations. A well-mixed layer is situated below a cloud layer across which q_t lapses significantly, capped by an inversion featuring an even sharper drop-off in humidity. Minor deviations do exist, but are always of second order compared to the mean structure; on average, the difference is only on the order of about 1 g/kg. In addition, the strongest deviations are mainly located above the boundary layer inversion. Based on the recent analysis by Schemann et al. (2020) we speculate that these minor deviations could be due to uncertainties in the large-scale forcing, initialization, or idealized setup of the simulations.

The general agreement of the model with the observed humidity structure on both days justifies emulating camera network images of the Cu in the LES. For the analysis presented in Sections 4 and 5, images are generated at a half-hourly frequency between 06:00–17:00 UTC, thus covering most of the diurnal cycle. On 21 July 2021 the period 14:00–16:00 UTC (gray area in Figure 4) was excluded from the analysis to avoid total sky coverage.

4. Results

4.1. Single Cloud Example

To first demonstrate the workflow including the hemispheric camera emulation, Figure 6 shows RT renderings of a single ShCu cloud from the LES on 22 July 2021 at 14:00 UTC. Two cross-sections are indicated, one along the *x*-Axis (purple plane) and one along the *y*-Axis (red plane). The side panels (Figures 6b and 6c show the emulated hemispheric images from the two single cameras at the surface, which are 400 m apart and are also indicated in Figure 6 (a). These two hemispheric images, generated using the control setup as listed in Table 2, form the input for the stereo reconstruction method.

Cloud data on the red and purple cross-sections is shown in Figure 7. When within range the locations of the two cameras are marked as black crosses at the surface, for reference. Apart from the cloudy state of the grid-boxes (orange), two aspects of the reconstruction method are also indicated. Following the reconstruction algorithm described in Section 2.3, a reconstructable cloud point must be part of the cloud hull (i.e., its outer boundary). But only part of the full cloud hull is visible to both cameras, due to obstruction by the cloud itself and/or by extruding cloud parts. The visible section of the full cloud hull is referred to as the "visible hull." In a discretized atmosphere, the full cloud hull is referred to as the "Hull Grid" (HG), while its subset of boxes that is visible to both cameras is named the "Visible Grid" (VG). A detailed description of the computation of VG is provided in Appendix B. In Figure 7 the VG is indicated in green.

Letting the stereo reconstruction algorithm work on the two hemispheric images shown in Figures 6b and 6c yields the reconstructed hull, indicated in red in Figure 7. In Figure 7a most grid boxes in the HG on the left and bottom side of the cloud are visible from both cameras (VG, green). Here the reconstruction is reasonably successful, with many VG gridboxes also reconstructed (red). The cloud geometrical structure on the perpendicular cross section (Figure 7b) is more complex. Only parts of the VG are reconstructed, and some reconstructed grid boxes are also placed outside the VG. One speculates that such mismatches in the reconstruction might happens for the following reasons:

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

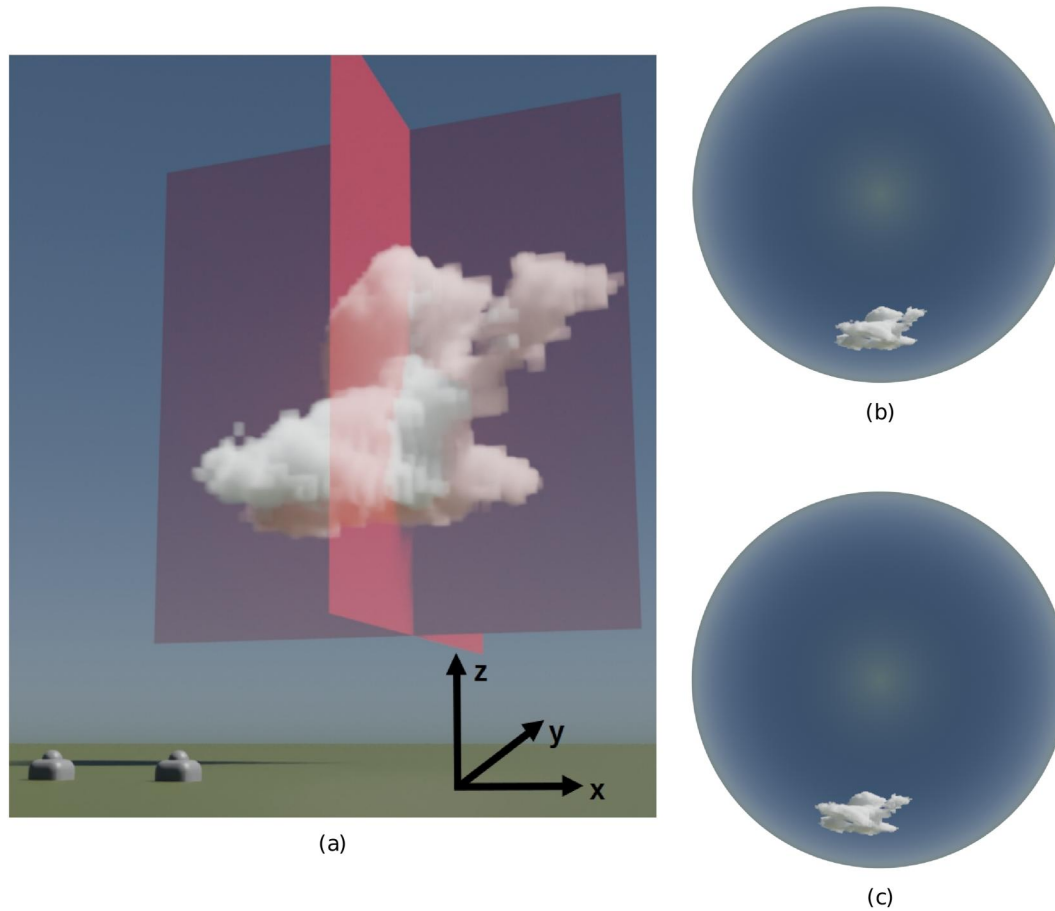

Figure 6. (a) Example rendering of a single ShCu cloud in the large-eddy simulations. The red and purple planes are two vertical cross sections along the x- and *y*-axis shown in Figures 7a and 7b, respectively. Panels (b) and (c) show the emulated hemispheric images of the left and right camera, respectively.

- Cloud edges with very low liquid water content are missed by the matching algorithm;
- Impacts of other aspects of the reconstruction algorithm;
- Artifacts of the space discretization via pixels by the hemispheric images;
- Impacts of the RT rendering of the LES clouds.

The first three possible causes are inherent to the stereo reconstruction algorithm, and are also effective when using real camera images. Only the fourth possible cause is exclusively related to the RT reconstruction of LES images. Gaining insight into these impacts is one of the research objectives of this study.

4.2. Visible Hull

The following sections present a more statistical evaluation of the method, proceeding from a single cloud to a full LES cloud field. The first step in the three-step approach is the calculation of the VG for the full cloud field, for which the procedure as described in Appendix B is used. The ratio of the full VG to the full HG is then defined as:

$$R_{\text{visible}} := \frac{|VG|}{|HG|}. \tag{4}$$

This ratio can be interpreted and used as a benchmark for what a perfect 3D reconstruction method would reproduce for a given camera network and a given cloud field.

Figure 8 shows a scatter plot of the number of VG versus HG, sampled at each half-hourly 3D output time step during the full 2-day period as described in Section 3. The black dashed diagonal represents the ideal scenario in

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

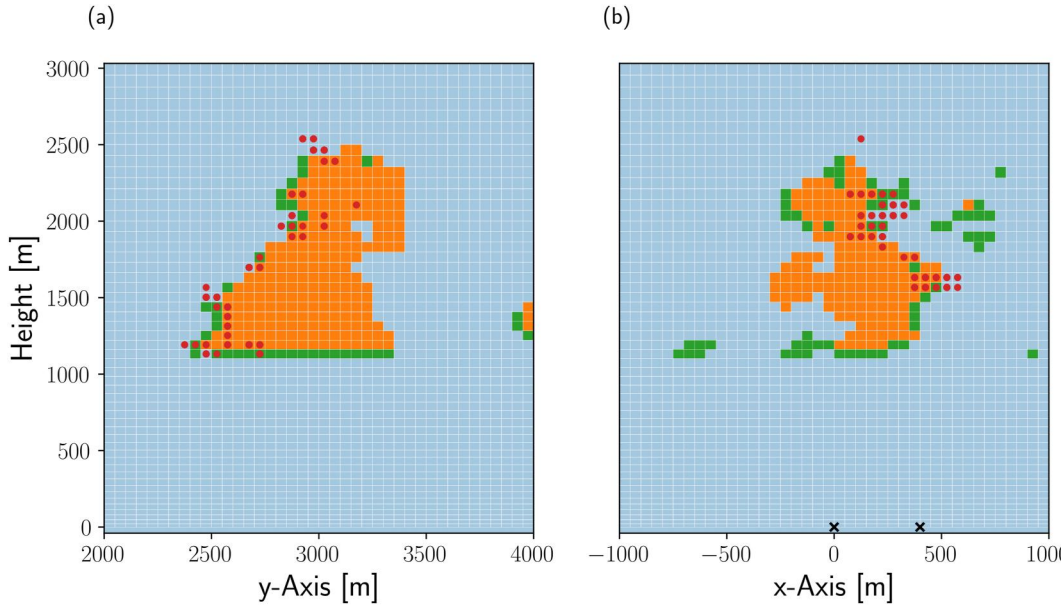

Figure 7. Grid box status along the *y*-axis (a) and along the *x*-axis (b) vertical slices as shown in Figure 6a. The orange grid boxes indicate the cloud. The green boxes represent the VG. The red dots represent the reconstructed grid boxes, using the hemispheric images shown in Figures 6b and 6c as input. The cameras are marked as black crosses near the surface in panel (b). For (a) the cameras are at $y = 0$. The box size is equal to large-eddy simulations discretization.

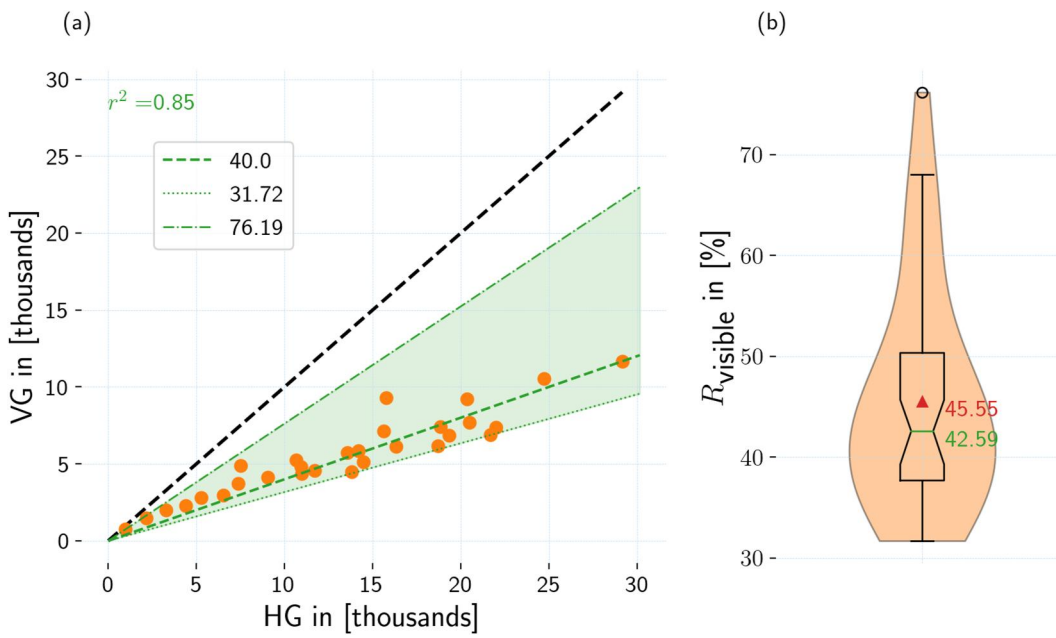

Figure 8. (a) Scatter plot of $R_{visible}$ (Equation 4) for analyzing the potential of the method used. Each orange data point represents the ratio of the number of HG to the number of VG in the domain for both days at a single sampling time. The least-square linear fit is also displayed (dashed green line, with spread marking the minimal fitting triangle). (b) Violin-Box plot summarizing the data in panel (a), showing the median (green line), the mean (red triangle), its 95%-confidence interval (notches), the interquartile range (boxed), 1.5 × that (whiskers), and outliers (circles). The brown area indicates the mirrored data distribution.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

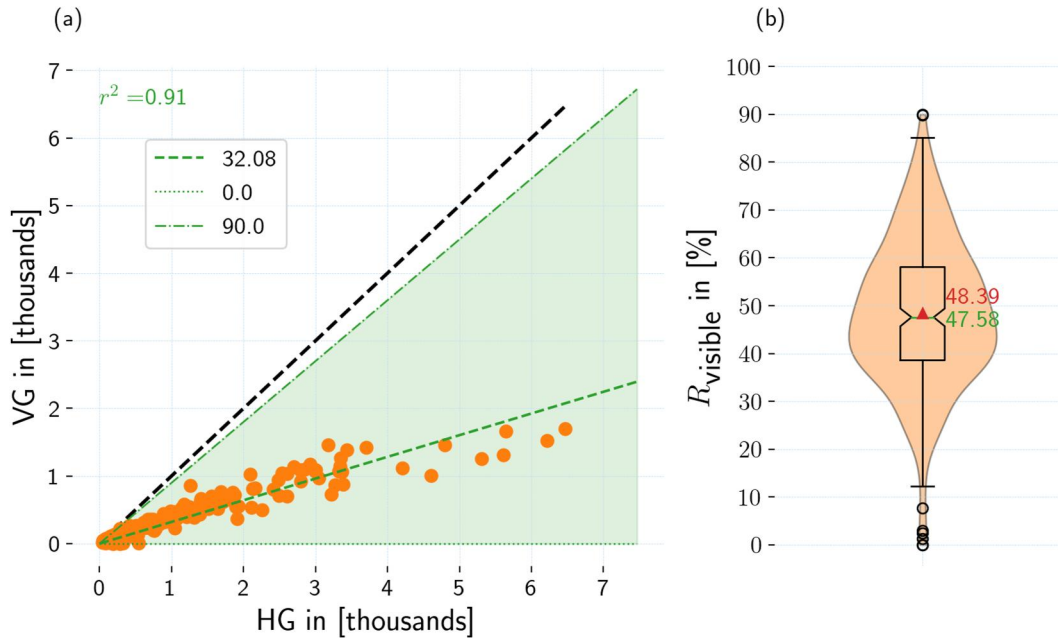

Figure 9. Same as Figure 8, but this time the averaging is performed over all single clouds rather than time steps.

which the cameras can see every hull grid box, a skill not achievable for any camera network that does not fully cover all cloud sides. The sampled data points are highly correlated at a linear coefficient of $r^2 = 0.85$. The green area is the minimal fitting triangle defined by the minimum and maximum values of $R_{visible}$ at each time step (32% and 76%, respectively). The Violin-Box Plot in Figure 8b statistically summarizes these results and their spread across the sampled period. The median $R_{visible}$ value for these two cumulus days expresses that, under some variation, approximately 43% of all gridded cloud hulls HG is visible to the stereo camera pair, and is theoretically reconstructable.

An informative alternative approach is to first calculate $R_{visible}$ for individual clouds and then average, instead of calculating VG and HG for the full cloud field in one go. To archive this, at each time step single clouds (defined as spatially adjacent cloudy gridboxes) are identified using the DBSCAN algorithm of the scikit-learn python library (Ester et al., 1996; Pedregosa et al., 2011; Schubert et al., 2017). To ensure that each cloud represents a characteristic ShCu, in this procedure only clouds with effective radii $r_{eff} = \sqrt{A/\pi}$ in the range 100–800 m are considered, where A is the projected area of each cloud. Figure 9 shows the VG and HG of all individual clouds encountered in the simulations for these 2 days. Because individual clouds are considered, the hull gridbox numbers are smaller compared to those of a full cloud field (as shown earlier in Figure 8).

Roughly half of all single ShCu cloud shells are visible, under considerable spread. $R_{visible}$ tends to be smaller for larger clouds, expressing that larger clouds are more challenging to observe from a single viewpoint. The large spread expresses that many clouds (in particular small ones) are easily obstructed by other clouds when far removed from the camera pair. While the values of $R_{visible}$ for the single cloud method are slightly higher compared to those produced by the full field method (shown in Figure 8) the average ratios are still roughly comparable. From this we conclude that the VG from both methods can be used as benchmark for assessing the skill of the 3D cloud reconstruction using path-tracing methods, as described next.

4.3. Hull Reconstruction

Given VG, the effectiveness of the reconstruction algorithm as applied to the hemispheric RT renderings is investigated using two ratios. The first ratio, R_{right}, measures the proportion of correctly reconstructed points among the VG. It is calculated as the number of correctly reconstructed points divided by the number of VG. The second ratio, R_{wrong}, measures the number of additional, incorrectly reconstructed points and is introduced in Section 4.4. These ratios help us to evaluate the performance of the reconstruction algorithm and identify areas for improvement.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

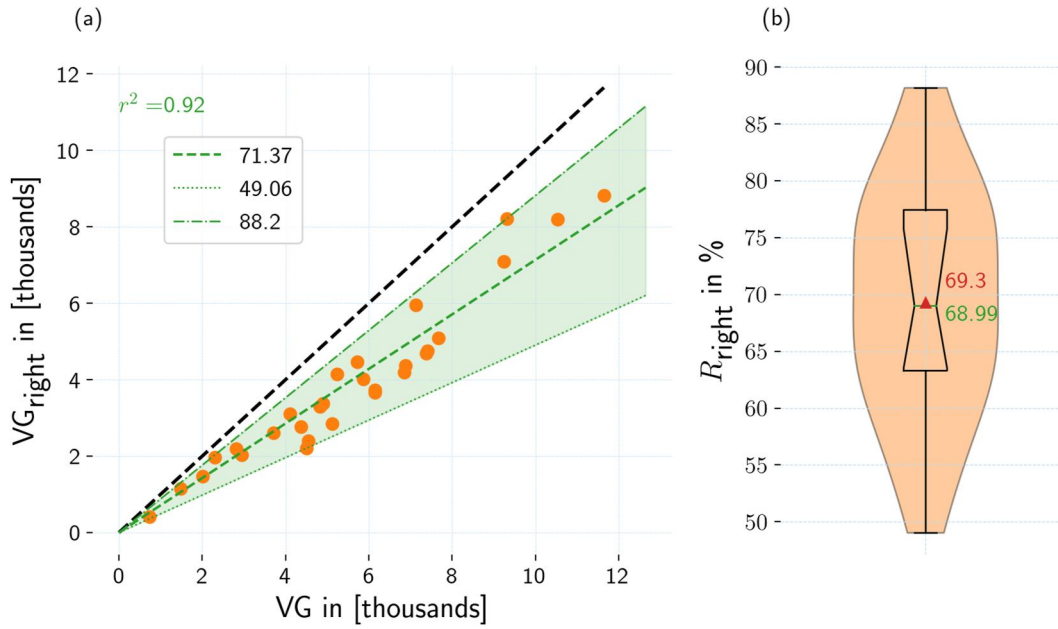

Figure 10. Same as Figure 8, but now showing R_{right} (Equation 6), representing the ratio of the number of correctly reconstructed visible grid boxes VG_{right} (Equation 5) to the total number of VG. The tolerance distance is 150 m.

To identify in practice which VG boxes are correctly reconstructed, the tolerance distance d is now introduced. If a 3D location produced by the reconstruction method is within d m from the closest visible grid box, then the latter is labeled as correctly reconstructed. This set of grid boxes, denoted as $VG_{right}(d)$, can be formally defined as follows:

$$VG_{right}(d) := \{g \in VG | \text{There is a } p \in Rec \text{ with } dist(g,p) \leq d\}, \quad (5)$$

where Rec is the set of reconstruction points, and dist(g, p) is the Euclidean distance from the midpoint of the grid box g to the reconstruction point p. Ratio $R_{right}(d)$ can now be defined as

$$R_{right}(d) := \frac{|VG_{right}(d)|}{|VG|}, \quad (6)$$

A value of 1 means that all hull grid boxes that are theoretically visible to the camera pair are also reconstructed, and thus represent optimal efficiency. A value less than one means that the reconstruction method for some reason fails to detect some of the VG subset of hull grid boxes. Tolerance distance d thus translates between reconstructed points in continuous space and a discretized spatial grid such as used in LES.

Figure 10a shows a scatter plot of VG_{right} versus VG, for a tolerance distance d = 150 m. Each data point again represents a single time step, with the data covering both days. The data is again well-correlated, with the linear fit (green dashed line) associated with a high correlation coefficient of r^2 = 0.92. Figure 10b shows that both the mean and the median R_{right} are about 69%, expressing that the algorithm as applied to the hemispheric renderings for the two cumulus days is able to reconstruct. This level of efficiency is statistically significant, and provides proof of principle that cloud hull reconstruction algorithms developed for stereo camera networks can in principle be applied to hemispheric RT imagery of LES clouds.

4.4. Reconstruction Errors

Figure 10 suggests that a significant fraction of VG boxes (~31%) remains unreconstructed. An alternative expression for the error in the reconstruction is R_{wrong}, defined here as the ratio of the number of incorrectly reconstructed points to the total number of reconstruction points. Mathematically, we can define the set of these wrongly reconstructed points as the points in the set of reconstruction points, Rec, that are farther than d m away from any visible grid box. This can be expressed as:

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

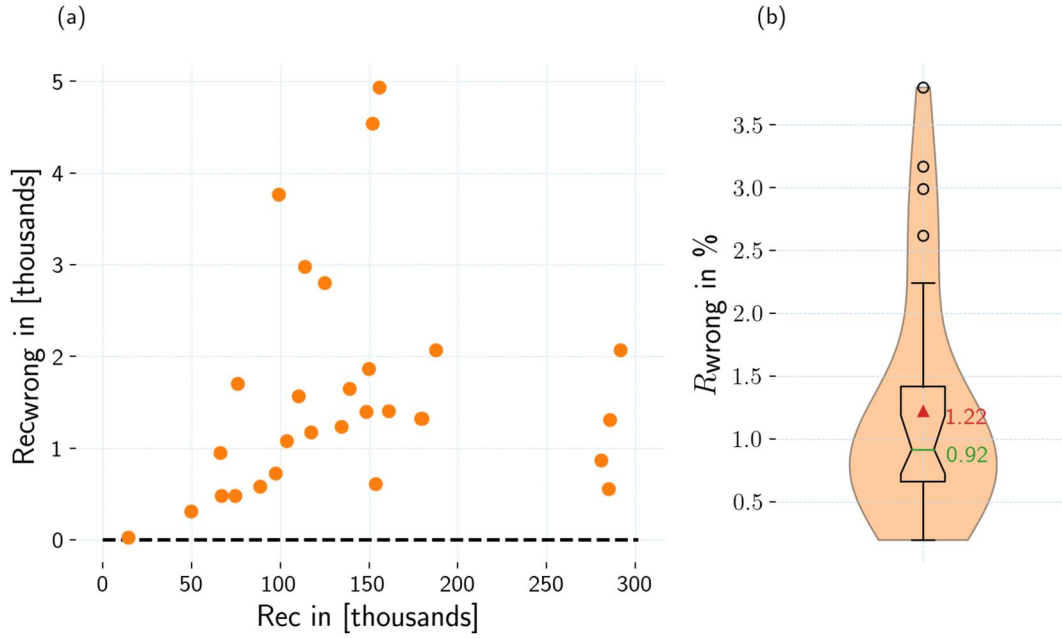

Figure 11. Same as Figure 10, but now showing R_{wrong} (Equation 8), representing the ratio of wrongly reconstructed points $\text{Rec}_{\text{wrong}}$ (Equation 7) to the total reconstruction points (Rec). The tolerance distance is 150 m.

$$\text{Rec}_{\text{wrong}}(d) := \{p \in \text{Rec} | \text{dist}(p, \text{VG}) > d\}, \tag{7}$$

where dist(p, VG) is the minimum Euclidean distance from the reconstruction point p to a visible grid box. With that we can define a new ratio $R_{\text{wrong}}(d)$ as

$$R_{\text{wrong}}(d) := \frac{|\text{Rec}_{\text{wrong}}(d)|}{|\text{Rec}|} \tag{8}$$

A ratio of 0 means all reconstructed points were within acceptable distance from a VG box, while a ratio of 1 indicates that all reconstructed points were too far from a VG box. A larger d widens the spatial margin of acceptance in the reconstruction of the cloud hull. Early tests with the reconstruction method revealed that the reconstruction frequently struggles to detect grid boxes with very low liquid water content. The spatial distribution of liquid water inside cumulus clouds is non-uniform and complex, with low liquid water contents frequently found near the cloud edges (Eytan et al., 2022). Distance d in effect creates a buffer zone near the cloud edge in successful reproduction of the cloud edge of all types. Note that d does not affect the reconstruction itself, only the diagnostics of its success rate in the form of R_{right} and R_{wrong}. Also, optically thin cloud edges are not unique for LES, but also occur in reality.

Figure 11 shows $\text{Rec}_{\text{wrong}}$ as a function of Rec, using a tolerance distance of 150 m. The line $y = 0$ (black dashed line) indicates a completely successful reconstruction; all reconstructed points are sufficiently close to a visible hull. In contrast to Figures 8 and 10, there is no clear correlation between Rec and $\text{Rec}_{\text{wrong}}$, suggesting that the number of failed reconstruction points does not depend on the total number of reconstructed points. This suggests that these errors appear randomly, regardless of the number of clouds in the camera images. Despite some outliers, R_{wrong} is generally in the range of 1%, expressing low errors for this value of d.

The sensitivities to d and cloud liquid water content are investigated in more detail in the next section.

5. Sensitivity Tests

While the cloud hull reconstruction efficiencies reported above are encouraging, the method is subject to various sensitivities that can affect the results. These are now examined in more detail. A distinction is made between

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 4
Medians of R_{right} *and* R_{wrong} *for Five Sensitivity Tests Discussed in 5.1*

Sensitivity test cases	R_{right}	R_{wrong}
Control setup	68.99%	0.92%
Anisotropic scattering inside clouds with an asymmetry factor of 0.85	65.23%	1.55%
Solar angular diameter of 0.53°	68.95%	0.94%
Solar zenith angle of 10°	63.27%	1.22%
Ocean surface setting	68.08%	0.99%
Isotropic scattering of 90% and in-cloud absorption of 10%	68.99%	0.91%

Note. The control setup (first row) represents the setup listed in Table 2.

world settings, reflecting physical aspects of the environment in which the clouds live including the behavior of daylight, and settings uniquely related to the RT algorithm. The impacts of world settings apply to imagery from both true and virtual cameras.

5.1. Miscellaneous World Settings

Table 4 compares the R_{right} and R_{wrong} for five different reconstructions, in each of which a single physical aspect of the virtual world as created in Blender is changed from its control value as listed in Table 2. These include (a) adjusting the cloud volume material to scatter anisotropically with an asymmetry factor of 0.85, (b) changing the solar angular diameter to 0.53°, (c) adjusting the solar zenith angle to 10°, (d) changing the surface color to

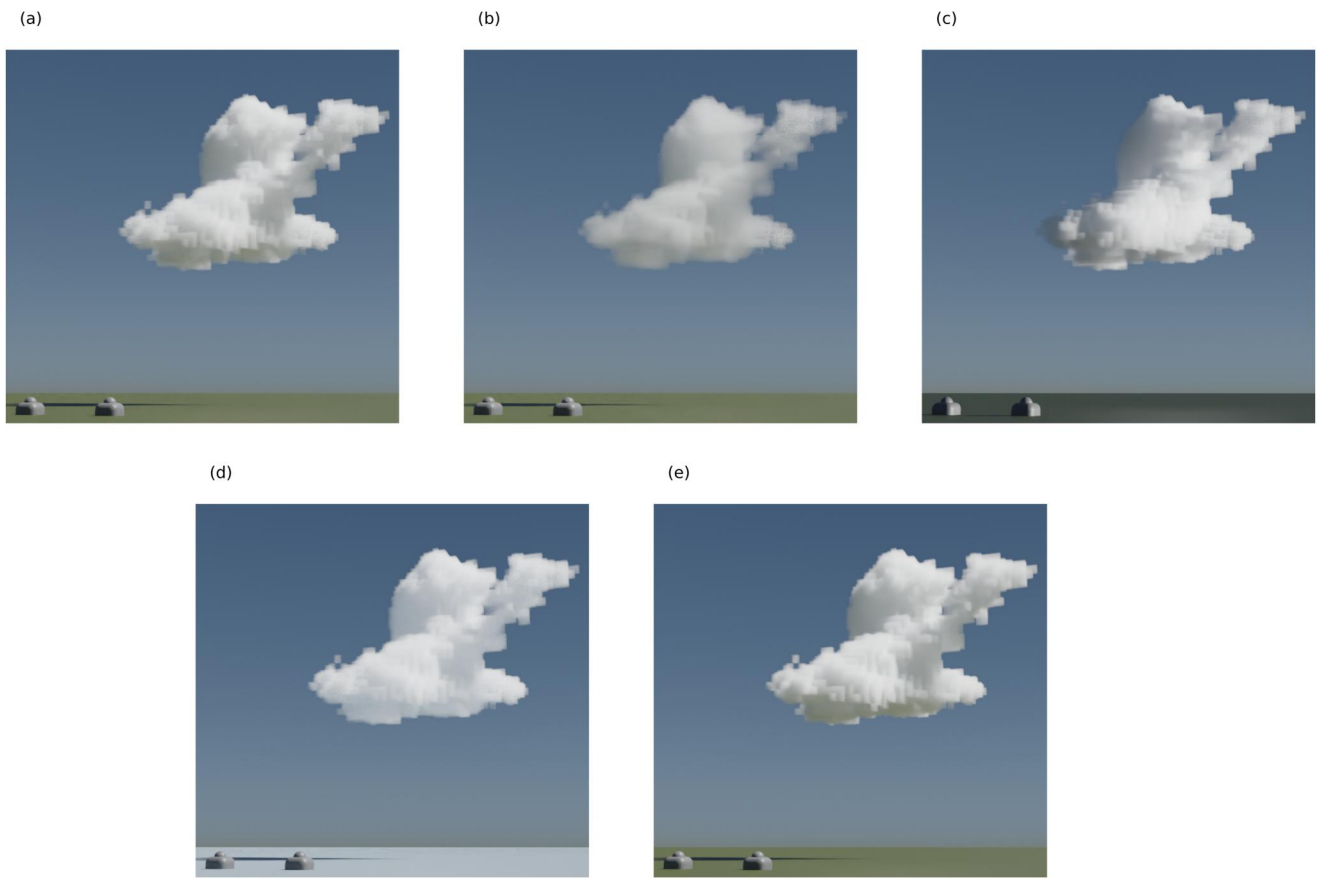

Figure 12. Same as Figure 6a, but for different world settings: (a) Control setup (b) anisotropic scattering with an asymmetry factor of 0.85, (c) solar zenith angle of 10° (d) sea surface, and (e) constant q_l of 1 g/kg.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

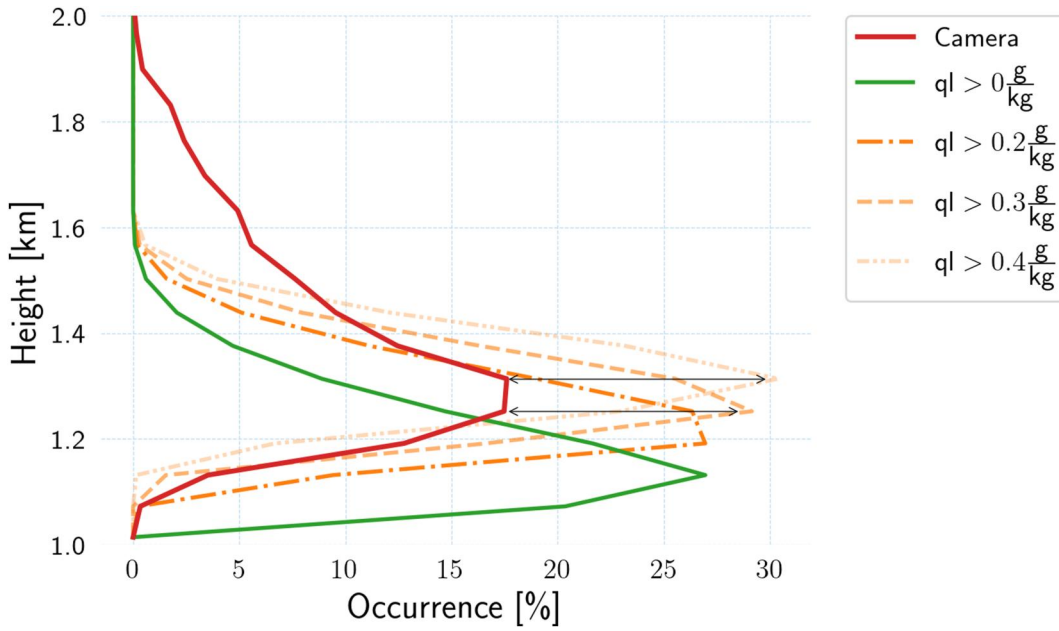

Figure 13. Average vertical profile of the camera reconstruction (red line) compared against the original simulation over the two simulation days. The green line (which has the lowest peak) represents the profile of all grid boxes that have a liquid water specific humidity (ql) bigger than 0. The other partly dotted orange lines illustrate the other corresponding thresholds written in the legend: >0.2 $\frac{g}{kg}$ are dash-dotted, >0.3 $\frac{g}{kg}$ are dashed, and >0.4 $\frac{g}{kg}$ are dash-dot-dotted. The black arrows indicate that the reconstruction algorithm detects grids with a ql-threshold bigger than somewhere in between 0.3 $\frac{g}{kg}$ and 0.4 $\frac{g}{kg}$.

resemble a blue-ish ocean, and (e) adjusting the cloud volume material was adjusted to scatter isotropically 90% and absorb 10%. The exact ocean surface setting can be found in the supplementing material. Rendered images for settings where significant differences against the control setup (Figure 12 (a)) are observed are displayed in Figures 12b–12d. The results suggest that none of these changes significantly affects the overall performance of the cloud hull reconstruction, with the solar zenith angle and the anisotropic scatter having the largest (but still modest) impacts.

5.2. Cloud Optical Thickness

Another world setting concerns cloud optical thickness. As previously mentioned in Sections 4.1 and 4.4, the reconstruction algorithm based on optical stereo imagery, either from true cameras or rendered from LES fields, sometimes fails to register cloudy points with very low liquid water content, which might be semi-transparent (Eytan et al., 2022). We find that in these situations the reconstructed cloud hull tends to be situated a short distance inside the cloud.

To gain more insight, Figure 13 first compares the average vertical profile of cloud occurrence as produced by the stereo reconstruction (red line) to similar profiles as sampled from the LES, in which the full three-dimensional cloud liquid water field is known. Four sampled profiles are included, each using a different threshold of liquid water content (q_l), and each displaying the bottom-heavy vertical structure typical of ShCu cloud fields (Brown et al., 2002; Ghate et al., 2011). The peaks in the stereo reconstruction profile fall between the peaks of the simulation for thresholds of 0.3 $\frac{g}{kg}$ and 0.4 $\frac{g}{kg}$, as indicated by the black arrows. This suggests that the cloud reconstruction algorithm can detect grid boxes with liquid water content larger than this threshold range. Furthermore, the difference between the peaks for ql > 0 and ql > 0.2 $\frac{g}{kg}$ is 182 m, suggesting that the error in the location of the reconstructed cloud boundary due to low optical thickness is of the same order of magnitude.

Another way of testing the impact of cloud optical thickness is to artificially modify the cloud liquid water field such that all cumulus clouds become completely opaque. In practice this is achieved by setting $q_l = 1\ g\ kg^{-1}$ in all gridboxes where $q_l > 0$ in the original three-dimensional LES cloud fields. A rendered image of this setting can be seen in Figure 12 (e). Upon meticulous comparison with the control setup depicted in Figure 12 (a), it becomes

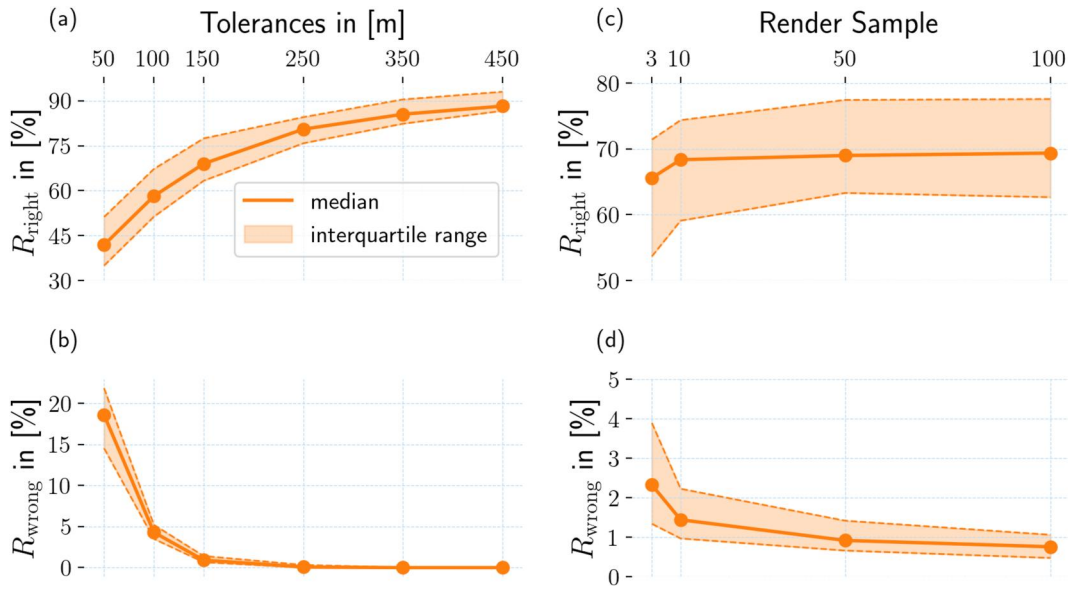

Figure 14. Medians of the two ratios (a) R_{right} (Equation 6) and (b) R_{wrong} (Equation 8) as a function of tolerance distance on the left panel. On the right panel the medians of the two ratios (c) R_{right} (Equation 6) and (d) R_{wrong} (Equation 8) as a function of the number of render samples.

evident that the cloud edges appear more pronounced and less foggy, particularly at the bottom left and the top right of the cloud. The impact of this cloud hack increases the efficiency of the cloud hull reconstruction, as expressed by $R_{\text{right}} = 72.63\%$ and $R_{\text{wrong}} = 0.89\%$. Artificially removing the effect of thin cloud edges thus significantly improves the hull reconstruction, but does not make it perfect either, due to other remaining error sources. The main value of this test is that it quantifies the thin cloud edge effect, which is also relevant for reconstructions based on real camera images.

5.3. Tolerance Distance

The tolerance distance is not a world setting, but is introduced to provide a spatial buffer around the cloud edge in which the cloud hull reconstruction is deemed successful (as described in Sections 4.3 and 4.4). In this section, we investigate how the R ratios depend on tolerance distance d.

Figure 14a shows the medians of R_{right} for various tolerance distances. As can be expected, R_{right} approaches 100% toward large d, reflecting that the method is highly tolerant and includes reconstructed cloud hull points that are far away from where they should be in 3D space. R_{right} reduces to 42% at $d = 50$ m, a relevant value being the spatial discretization Δx of these LES experiments, used here as lower limit. At this minimum tolerance level at least 42% of the VG are reconstructed. Toward larger d the ratio then gradually increases, via 81% at $d = 250$ m toward 95% at $d = 750$ m. At this high tolerance level most visible grid boxes are within range of a reconstructed point, as expressed by the much weaker increase of R_{right} with d in this range. The effect of d on R_{wrong} is examined in Figure 14b, showing that R_{wrong} quickly decreases with larger d, dropping to sub-percentage values for $d > 3\Delta x$ (the LES discretization).

From these results we conclude that the reconstruction algorithm performs well (i.e., above 70%) at a spatial tolerance of about 3× the LES discretization. This tolerance distance more or less matches what is commonly considered the effective resolution of an LES, which motivated adopting this value in the control setup. It can be interpreted as a trade-off between tolerance for the effect of thin cloud edges and the wish to adequately reconstruct small-scale cloud geometry (in this case as resolved in the LES).

5.4. Path-Tracing Render Sample

The render sample is another non-world setting, and expresses the area density of rays that is used to render the hemispheric projections of LES clouds. As explained in Section 2.2, a too small number of render samples causes

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

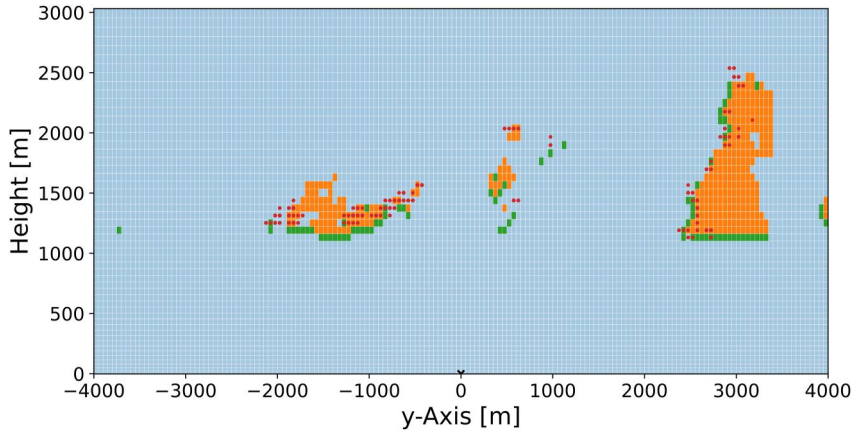

Figure 15. Same as Figure 7a, but extended to include the cloud field. The cameras' positions are denoted by a black cross near the surface.

RT images to look "grainy," which can affect the cloud hull reconstruction. Figures 14c and 14d show the median values of R_{right} and R_{wrong} for different number of render samples, respectively, at a tolerance distance $d = 150$ m. The results indicate that the number of render samples has a minimal impact on the algorithm's performance. Only below a render sample of 10 an impact exists, a range in which the images look very grainy (usually RT renderings use a much higher sample). Above this range the results remain constant, even for a render sample of 256 (not shown). The interquartile range also remains stable in this range, implying consistent and robust performance. The lack of sensitivity above a render sample of 10 motivated adopting a value of 50 for the control setup as listed in Table 2.

6. Discussion

In recent years, various methods have been proposed for investigating cumulus (Cu) cloud geometry and dynamics using both stereo camera systems (e.g., Beekmans et al., 2016; Crispel & Roberts, 2018; Nguyen & Kleissl, 2014) and multi-camera networks (Blum et al., 2021; Nouri et al., 2019; Romps & Öktem, 2018). To our knowledge there has not been an attempt yet to emulate such hemispheric camera images from LES output using path-tracing rendering, and to subsequently use these renderings to three-dimensionally reconstruct the simulated cloud field. One of the main outcomes of this study is that this is in principle possible, supporting the use of this technique as an instrument simulator for evaluating LES cloud fields against true hemispheric camera data.

Figure 15 displays a vertical cross-section of the cloud field, visually demonstrating what data a stereo camera simulator would produce in practice. The cross-section highlights a few important considerations and potential limitations. First, sometimes the reconstructed grid (red dots) is situated a short distance inside the LES clouds. In this cross-section, this behavior is visible in the flat cloud situated between $y \in (-2, 0)$ km. The analysis in 5.2 shows that this effect can at least partially be attributed to the sensitivity of the stereo reconstruction algorithm to the integrated liquid water content along the line of sight (optical thickness). In the case of optically thick and sharp cloud boundaries, such as the tall cloud at $y \in (2.5, 3.5)$ km, the reconstructed grid matches the true cloud edge to a high degree. It is crucial to note that the sensitivity to optical thickness is solely a result of the reconstruction algorithm, and not related to LES or the path-tracer algorithm. Having full knowledge of the LES cloud field, including gridboxes with very low liquid water content, allowed us to gain insight into this effect. The sensitivity depth of 182 m as reported in 5.2 can thus be interpreted as a typical error in cloud boundaries as reconstructed from stereo camera imagery in general. Accordingly, this result can guide future scientific studies of cloud geometry based on true camera imagery.

The camera network emulated in this study consists of an all-sky imager stereo pair, spatially separated by 400 m. But the method can similarly be applied to different camera networks. For example, the path-tracer algorithm (as described in Section 2.2)also works for other camera projections, including rectilinear images (see Figures 1a, 3, and 6a), those used by Romps and Öktem (2018); Blum et al. (2021), and networks

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

using the space carving technique Nouri et al. (2019); Veikherman et al. (2015). Another opportunity created is to use the LES fields to identify the ideal spacing of such networks for an optimal reconstruction of a cumulus cloud field. The method also still needs to be tested for goals going beyond pure location reconstruction, such as estimating cloud dynamics and evolution. These ideas are for now considered a future research topics.

In Section 5 the sensitivity of the method for a selection of aspects was investigated. These included both (virtual) world settings and settings of the RT algorithm. But the reconstruction algorithm itself also introduces a few potential error sources. These include (a) the matching algorithm used in the reconstruction process, becoming confused by the lower contrast at the cloud base; (b) the conversion from the hemispheric perspective to the 3D space, which uses a step-like pattern that may overlook some in-between grid boxes; and (c) the 3D space discretization by the number of pixels in the camera images. Further investigation is needed to gain insight.

In this study the open-source Cycles algorithm is applied for the RT, as included in the 3D compute graphic software Blender (www.blender.org). Cycles uses its own parameterizations of how light behaves in nature. The impact of some parametric settings was tested in Section 5. Based on the robustness that we find, we speculate that other RT engines should produce similar results, an assumption that still needs investigation. Also note that in the rendering of the virtual clouds, a few atmospheric and non-atmospheric processes were ignored (for simplicity) that could affect the comparability to real camera images. These include

- Blurring and brightening effects of the camera images;
- Haze effects due to atmospheric aerosol and water vapor;
- Surface obstacles such as towers and buildings;
- Pollution on the acrylic glass dome protecting the camera sensors, such as insects, water droplets or dirt;

A careful assessment of these further impacts is for now considered future work. Finally, in this study no image denoising was used for any results. It was only used to enhance the visualization in the Figures (namely for 1, 3, 6, and 12); a brief investigation of how this could work is described in Appendix C. We find that denoising the images has no significant impact on the results (not shown).

A final aspect of the rendering that could matter is the way LES grid boxes are treated, visualizing them as homogeneously filled cubes. This makes the spatial discretization of the model visible in the camera renderings (see Figures 1, 3, 6, and 12). As our objective is the scientific analysis of model simulations including resolution impacts, this approach is arguably the most appropriate and honest. One could adopt further smoothing algorithms to achieve a more realistic-looking, less blocky rendering, perhaps for the purpose of illustration or presentation. However, one then artificially alters the actual cloud shapes in the simulation. Accordingly, for scientific analyses such prettified renderings should perhaps best be avoided.

7. Concluding Remarks

This study employs path-tracing rendering to generate hemispheric images of simulated continental mid-latitude ShCu cloud fields in high-resolution LES. The first objective is to thus emulate how networks of virtual all-sky imagers observe clouds in nature. The second objective is to then reconstruct the three-dimensional model cloud field again from these renderings, using the same method as commonly applied with real-world camera images. The main advantage is that the model cloud field, which is reconstructed from the virtual hemispheric images, is also fully known, in three dimensions. This enables the rigorous statistical analysis of the stereo camera cloud reconstruction method in general. The results also provide insight into the applicability of the path-tracing based method as an instrument simulator to evaluate clouds in LES models against real stereo camera data. These include enabling statistics of spatial and temporal changes of clouds by analyzing cloud areas, cloud shapes, and their fractal dimensions over time.

We find that, in general, the method is successful in reconstructing the visible cloud hulls from a stereo pair of virtual LES cloud field renderings. This result is also relatively robust to certain hyperparameters. More precisely, we find that, on average, a single stereo all-sky imager pair can theoretically detect approximately 42% of the cloud hulls of an entire ShCu cloud field. Of this subset, about 69% is successfully reconstructed by the reconstruction method as applied to path-tracing renderings, with only about 1% wrongly reconstructed grid

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

boxes. This is the key result coming out of this study, and provides proof of principle for cloud reconstruction based on RT renderings of LES cloud fields.

The sensitivity to various settings in the method was investigated. Adopting a tolerance of 3× the LES discretization yields the reconstruction skill as mentioned above. Interestingly, this tolerance distance more or less coincides with the effective resolution of the LES in resolving ShCu clouds. It is well known that a stereo camera reconstruction algorithm is sensitive to cloud optical thickness; using our analysis of model cloud fields, we find that the associated error in the cloud boundary placement is approximately 182 m. Surprisingly, the render sample or "graininess" of the RT rendering has little impact on the results.

The results obtained in this study are relevant for both cloud photogrammetry and atmospheric modeling. From an observational perspective, it enables the optimization of camera network configurations, including the mutual distances between cameras, before installation in the field. This can help to improve the accuracy and efficiency of data collection during field campaigns. From a modeling standpoint, the method provides a novel means of evaluating cloud-resolving simulations against real-world data, by directly emulating optical camera images. This allows a direct and fair comparison of observed and simulated three-dimensional cloud geometry. As convective clouds are becoming ever more resolved in numerical simulations of atmospheric flow, for example, in operational numerical weather forecasting, this creates new opportunities for critically evaluating and improving such next-generation models. Investigating the impact of model resolution on the cloud reconstruction can provide insight into the science question at what discretization the model starts to reproduce the observed cumulus cloud geometry at small scales.

Appendix A: Blender Settings

The Blender settings were chosen in the following way: The ground is modeled by a simple green surface texture. Regarding the cloud material, for each time step, we created a grid box for every simulation grid box that contains liquid water. Thereby, it is important that there are no facets between two grid boxes and that the vertex order of the facets is chosen rightly since it also determines the normal of the path-tracing. The light scattering and absorption are then modeled by using the Point Density Node of Cycles in Blender. Since this Point Density Node only works for integer numbers we took the rounded integer number of

$$\left(\frac{\mathrm{ql}}{\max(\mathrm{ql})}\right)^{\frac{1}{4}}$$

as an approximation, where ql is the vector of the liquid water specific humidity. The radius of the Point Density Node is set to be the biggest ball inside each grid box, and hence the square root of the horizontal resolution. The Cycles Nodes of the cloud material can be seen in Figure A1.

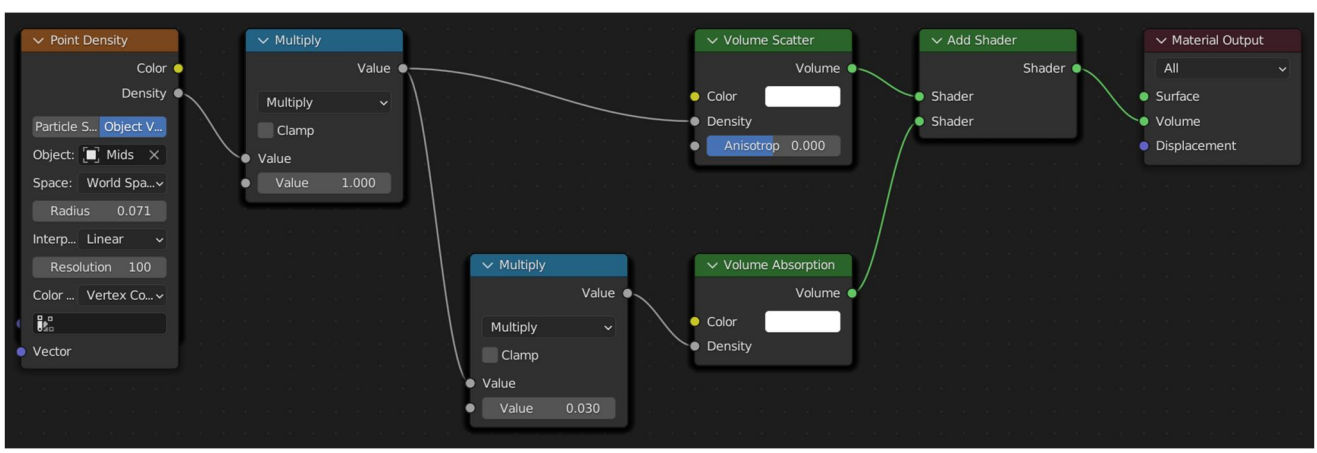

Figure A1. The Cycle Nodes of used cloud material in Blender.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Appendix B: Visible Grids Algorithm

To compute the visible grids (green boxes in Figure 7), we developed an algorithm (Algorithm 1). To achieve this, first, all grid boxes that are visible by just one camera are computed. These grid boxes can only be the ones constructing the cloud hulls.

To calculate these hull grid boxes, the grid boxes are tracked along each dimension to identify the grid boxes that are not surrounded by other grid boxes. For every hull grid box, we utilize Bresenham's Line Algorithm (Algorithm 2), originating from Bresenham (1965) to compute the grid boxes composing the path from the hull grid box to the camera (Line 5 of Algorithm 1).

The resulting path grid boxes are intersected with the hull grid boxes (Line 6). The grid box closest to the camera is saved (Lines 7–9). This process is done for both cameras (Lines 10–14). Afterward, the intersection of the visible sets for both single cameras is taken (Line 15).

Algorithm 1. Visible Grids Algorithm

Input: *boundary_grids*, *camera_location_1*, *camera_location_2*
Output: *visible_grids*

```
1: visible_grids = {}
2: visible_grids_1 = {}
3: visible_grids_2 = {}
4: for grid in boundary_grids do
5:   line = Bresenham's_Line_Algorithm(camera_location_1, grid)
6:   intersection = line ∩ boundary_grids
7:   distances = ||intersection − camera_location_1||
8:   if arg min(distances) not in visible_grids_1 then
9:       visible_grids_1.append(argmin(distances))
10:  line = Bresenham's_Line_Algorithm(camera_location_2, grid)
11:  intersection = line ∩ boundary_grids
12:  distances = ||intersection − camera_location_2||
13:  if argmin(distances) not in visible_grids_2 then
14:      visible_grids_2.append(argmin(distances))
15: visible_grids = visible_grids_1 ∩ visible_grids_2
16: return visible_grids
```

Algorithm 2. Bresenham's Line Algorithm

Input: *location_1*, *location_2*
Output: *line*

```
1: line = {}
2: x, y, z = location_1
3: d = location_2 − location_1
4: if |d[0]| ≥ |]d[1]| and |d[0]| ≥ |d[2]| then          ▷ Walking along the X-axis
5:   a_1 = 2 · |d[1]| − |d[0]|
6:   a_2 = 2 · |d[2]| − |d[0]|
7:  while x ≠ location_2[0] do
8: x = x + sgn(d[0])
9: if a_1 ≥ 0 then
```

```
y = y + sgn(d[1])
a_1 = a_1 − 2·|d[0]|
if a_2 ≥ 0 then
z = z + sgn(d[2])
a_2 = a_2 − 2·|d[0]|
a_1 = a_1 + 2·|d[1]|
a_2 = a_2 + 2·|d[2]|
line.append([x, y, z])
else if |d[1]| ≥ |d[0]| and |d[1]| ≥ |d[2]| then        ▷ Walking along the Y-axis
  a_1 = 2·|d[0]| − |d[1]|
  a_2 = 2·|d[2]| − |d[1]|
  while y ≠ location_2[1] do
y = y + sgn(d[1])
if a_1 ≥ 0 then
x = x + sgn(d[0])
a_1 = a_1 − 2·|d[1]|
if a_2 ≥ 0 then
z = z + sgn(d[2])
a_2 = a_2 − 2·|d[1]|
a_1 = a_1 + 2·|d[0]|
a_2 = a_2 + 2·|d[2]|
line.append([x, y, z])
else                                                  ▷ Walking along the Z-axis
  a_1 = 2·|d[1]| − |d[2]|
  a_2 = 2·|d[0]| − |d[2]|
  while z ≠ location_2[2] do
z = z + sgn(d[2])
if a_1 ≥ 0 then
y = y + sgn(d[1])
a_1 = a_1 − 2·|d[2]|
if a_2 ≥ 0 then
x = x + sgn(d[0])
a_2 = a_2 − 2·|d[2]|
a_1 = a_1 + 2·|d[1]|
a_2 = a_2 + 2·|d[0]|
line.append([x, y, z])
return line
```

Appendix C: Denoising

The used Open Image Denoising Ray Tracing (RT) Filter is a Convolutional Neural Network based filter that improves the quality of images generated by path-tracing by removing the inherent noise. This filter is part of the open-source Open Image Denoised library (www.openimagedenoise.org) and it is integrated in Blender. The denoising process relies on three inputs: (a) the rendered image, (b) an albedo image that represents surface colors without considering lighting or viewing angles, and (c) a normal image that contains shading normals. Dielectric surfaces, including cloud grid boxes, are set to white in the albedo image. Figure C1 compares the camera image Figure 1b in its denoised (Figure C1b) and non-denoised (Figure C1a) form, demonstrating the impact of the denoising.

Conflict of Interest

The authors declare that they have no conflict of interest.

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

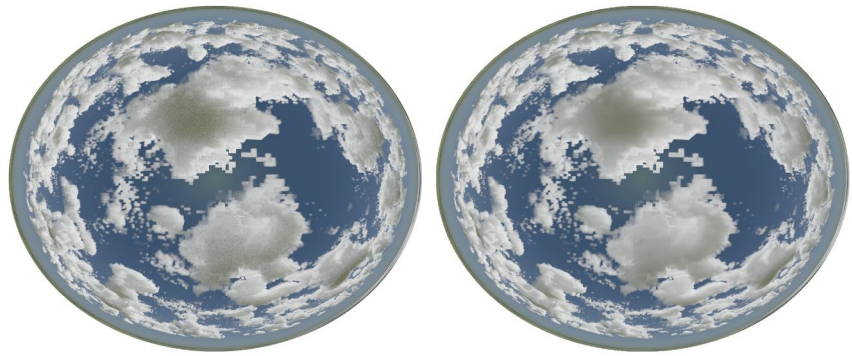

Figure C1. Example image to compare the effect of denoising. (b) Is the same as Figures 1a and 1b the same image, but not denoised.

Data Availability Statement

The data set to reproduce the results is available at https://doi.org/10.5281/zenodo.10185345 (Burchart et al., 2023). It contains the Large-Eddy Simulation (LES) model configuration files, selected output data of the LES experiments, data and analysis scripts for the figures, the rendered camera images, the cloud field, cloud hulls, and reconstructed hulls. Furthermore, it contains a frozen version of the open-source Blender code (version 2.90) as used in this study. The recent Blender version is available at www.blender.org (Blender Online Community, 2023). It is important to note that the method was specifically tested only on version 2.90. The DALES version used (dales 4.3) is available at https://doi.org/10.5281/zenodo.5642477 (van Heerwaarden et al., 2021).

Acknowledgments
This research is supported by the German Research Foundation (DFG) under project number 430226822 (https://gepris.dfg.de/gepris/projekt/430226822). This research was supported by the U.S. Department of Energy's Atmospheric System Research, an Office of Science Biological and Environmental Research program, under grant DE-SC0022126. This work used resources of the Deutsches Klimarechenzentrum (DKRZ) granted by its Scientific Steering Committee (WLA) under project ID bb1086. The Gauss Centre for Supercomputing e.V. (https://www.gauss-centre.eu/) is acknowledged for providing computing time on the Gauss Centre for Supercomputing (GCS) supercomputer Jülich Wizard for European Leadership Science at the Jülich Supercomputing Centre under projects VIRTUALLAB and RCONGM. We thank two anonymous reviewers for their insightful comments and suggestions that significantly improved the manuscript. YB and RN collaborated to generate the research idea. RN designed the model framework and strategy, and prepared the model forcing files. RN developed the Path-Tracer Algorithm. CB contributed the Stereo Reconstruction Algorithm and provided guidance on handling stereo camera reconstructions. YB carried out the radiosonde measurements and devised the algorithm for computing the visible hull. YB combined the Path-Tracer Algorithm with the Stereo Reconstruction Algorithm and conducted tests to determine potential sensitivities, ultimately leading to the obtained results. YB and RN jointly analyzed and evaluated the results, with some insights from CB. YB conducted the data visualization and interpretation, with significant advice from RN. YB prepared the manuscript, which underwent revisions by RN and CB in multiple intermediate versions. YB curated the research data. Open Access funding enabled and organized by Projekt DEAL.

References

Angevine, W. M., Olson, J., Kenyon, J., Gustafson, W. I., Endo, S., Suselj, K., & Turner, D. D. (2018). Shallow cumulus in WRF parameterizations evaluated against LASSO large-eddy simulations. *Monthly Weather Review*, *146*(12), 4303–4322. https://doi.org/10.1175/MWR-D-18-0115.1

Beekmans, C., Schneider, J., Läbe, T., Lennefer, M., Stachniss, C., & Simmer, C. (2016). Cloud photogrammetry with dense stereo for fisheye cameras. *Atmospheric Chemistry and Physics*, *16*(22), 14231–14248. https://doi.org/10.5194/acp-16-14231-2016

Blender Online Community. (2023). Blender – A 3d modelling and rendering package [software]. Blender Institute, Amsterdam. http://www.blender.org

Blum, N. B., Nouri, B., Wilbert, S., Schmidt, T., Lünsdorf, O., Stührenberg, J., et al. (2021). Cloud height measurement by a network of all-sky imagers. *Atmospheric Measurement Techniques*, *14*(7), 5199–5224. https://doi.org/10.5194/amt-14-5199-2021

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, *4*(1), 25–30. https://doi.org/10.1147/sj.41.0025

Brown, A. R., Cederwall, R. T., Chlond, A., Duynkerke, P. G., Golaz, J.-C., Khairoutdinov, M., et al. (2002). Large-eddy simulation of the diurnal cycle of shallow cumulus convection over land. *Quarterly Journal of the Royal Meteorological Society*, *128*(582), 1075–1093. https://doi.org/10.1256/003590002320373210

Burchart, Y., Beekmans, C., & Neggers, R. (2023). A stereo camera simulator for large-eddy simulations of continental shallow cumulus clouds based on three-dimensional path-tracing (version 3) [dataset]. https://doi.org/10.5281/zenodo.10185345

Cesana, G. V., & Del Genio, A. D. (2021). Observational constraint on cloud feedbacks suggests moderate climate sensitivity. *Nature Climate Change*, *11*(3), 213–218. https://doi.org/10.1038/s41558-020-00970-y

Crispel, P., & Roberts, G. (2018). All-sky photogrammetry techniques to georeference a cloud field. *Atmospheric Measurement Techniques*, *11*(1), 593–609. https://doi.org/10.5194/amt-11-593-2018

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). AAAI Press.

Eytan, E., Khain, A., Pinsky, M., Altaratz, O., Shpund, J., & Koren, I. (2022). Shallow cumulus properties as captured by adiabatic fraction in high-resolution LES simulations. *Journal of the Atmospheric Sciences*, *79*(2), 409–428. https://doi.org/10.1175/JAS-D-21-0201.1

Fielding, M. D., Schäfer, S. A. K., Hogan, R. J., & Forbes, R. M. (2020). Parametrizing cloud geometry and its application in a subgrid cloud-edge erosion scheme. *Quarterly Journal of the Royal Meteorological Society*, *146*(729), 1651–1667. https://doi.org/10.1002/qj.3758

Ghate, V. P., Miller, M. A., & DiPretore, L. (2011). Vertical velocity structure of marine boundary layer trade wind cumulus clouds. *Journal of Geophysical Research*, *116*(D16), D16206. https://doi.org/10.1029/2010JD015344

Heinze, R., Dipankar, A., Henken, C. C., Moseley, C., Sourdeval, O., Trömel, S., et al. (2017). Large-eddy simulations over Germany using ICON: A comprehensive evaluation. *Quarterly Journal of the Royal Meteorological Society*, *143*(702), 69–100. https://doi.org/10.1002/qj.2947

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., et al. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Heus, T., Barron, N., Neggers, R., Pschera, A., & Burchart, Y. R. (2021). *Efficient rendering of simulated cloud fields using blender*. AGU. Retrieved from https://agu.confex.com/agu/fm21/meetingapp.cgi/Paper/865504

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Heus, T., Van Heerwaarden, C. C., Jonker, H. J. J., Siebesma, P., Axelsen, S., Van den Dries, K., et al. (2010). Formulation of the Dutch atmospheric large-eddy simulation (DALES) and overview of its applications. *Geoscientific Model Development*, *3*(2), 415–444. https://doi.org/10.5194/gmd-3-415-2010

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 2, pp. 807–814). https://doi.org/10.1109/CVPR.2005.56

Hosek, L., & Wilkie, A. (2012). An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics*, *31*(4), 95:1–95:9. https://doi.org/10.1145/2185520.2185591

Kajiya, J. T. (1986). The rendering equation. In *Proceedings of the 13th annual conference on computer graphics and interactive techniques* (pp. 143–150). Association for Computing Machinery. https://doi.org/10.1145/15922.15902

Löhnert, U., Schween, J. H., Acquistapace, C., Ebell, K., Maahn, M., Barrera-Verdejo, M., et al. (2015). JOYCE: Jülich observatory for cloud evolution. *Bulletin of the American Meteorological Society*, *96*(7), 1157–1174. https://doi.org/10.1175/BAMS-D-14-00105.1

Malkus, J. S., & Ronne, C. (1954). On the structure of some cumulonimbus clouds which penetrated the high tropical troposphere. *Tellus*, *6*(4), 351–366. https://doi.org/10.3402/tellusa.v6i4.8758

Myers, T. A., Scott, R. C., Zelinka, M. D., Klein, S. A., Norris, J. R., & Caldwell, P. M. (2021). Observational constraints on low cloud feedback reduce uncertainty of climate sensitivity. *Nature Climate Change*, *11*(6), 501–507. https://doi.org/10.1038/s41558-021-01039-0

Neggers, R., Heus, T., & Siebesma, A. P. (2011). Overlap statistics of cumuliform boundary-layer cloud fields in large-eddy simulations. *Journal of Geophysical Research*, *116*(D21). https://doi.org/10.1029/2011JD015650

Neggers, R., Neelin, J. D., & Stevens, B. (2007). Impact mechanisms of shallow cumulus convection on tropical climate dynamics. *Journal of Climate*, *20*(11), 2623–2642. https://doi.org/10.1175/JCLI4079.1

Neggers, R., Stevens, B., & Neelin, J. D. (2007). Variance scaling in shallow-cumulus-topped mixed layers. *Quarterly Journal of the Royal Meteorological Society*, *133*(628), 1629–1641. https://doi.org/10.1002/qj.105

Nguyen, D. A., & Kleissl, J. (2014). Stereographic methods for cloud base height determination using two sky imagers. *Solar Energy*, *107*, 495–509. https://doi.org/10.1016/j.solener.2014.05.005

Nouri, B., Kuhn, P., Wilbert, S., Hanrieder, N., Prahl, C., Zarzalejo, L., et al. (2019). Cloud height and tracking accuracy of three all sky imager systems for individual clouds. *Solar Energy*, *177*, 213–228. https://doi.org/10.1016/j.solener.2018.10.079

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Romps, D. M., & Öktem, R. (2018). Observing clouds in 4d with multiview stereophotogrammetry. *Bulletin of the American Meteorological Society*, *99*(12), 2575–2586. https://doi.org/10.1175/BAMS-D-18-0029.1

Roode, S. R. d., Sandu, I., Dussen, J. J. v. d., Ackerman, A. S., Blossey, P., Jarecka, D., et al. (2016). Large-eddy simulations of EUCLIPSE–GASS Lagrangian stratocumulus-to-cumulus transitions: Mean state, turbulence, and decoupling. *Journal of the Atmospheric Sciences*, *73*(6), 2485–2508. https://doi.org/10.1175/JAS-D-15-0215.1

Schemann, V., Ebell, K., Pospichal, B., Neggers, R., Moseley, C., & Stevens, B. (2020). Linking large-eddy simulations to local cloud observations. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002209. https://doi.org/10.1029/2020MS002209

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, *42*(3), 19:1–19:21. https://doi.org/10.1145/3068335

Schulz, H., Eastman, R., & Stevens, B. (2021). Characterization and evolution of organized shallow convection in the downstream north Atlantic trades. *Journal of Geophysical Research: Atmospheres*, *126*(17), e2021JD034575. https://doi.org/10.1029/2021JD034575

Seifert, A., & Beheng, K. D. (2006). A two-moment cloud microphysics parameterization for mixed-phase clouds. Part 1: Model description. *Meteorology and Atmospheric Physics*, *92*(1), 45–66. https://doi.org/10.1007/s00703-005-0112-4

Shen, Z., Sridhar, A., Tan, Z., Jaruga, A., & Schneider, T. (2022). A library of large-eddy simulations forced by global climate models. *Journal of Advances in Modeling Earth Systems*, *14*(3), e2021MS002631. https://doi.org/10.1029/2021MS002631

Shirley, P. S. (1991). Physically based lighting calculations for computer graphics (PhD thesis). Retrieved from https://www.proquest.com/docview/303946949/abstract/97436A8422A24E76PQ/1

Siebesma, A. P., & Jonker, H. J. J. (2000). Anomalous scaling of cumulus cloud boundaries. *Physical Review Letters*, *85*(1), 214–217. https://doi.org/10.1103/PhysRevLett.85.214

Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Monthly Weather Review*, *117*(8), 1779–1800. https://doi.org/10.1175/1520-0493(1989)117⟨1779:ACMFSF⟩2.0.CO;2

van Heerwaarden, C., Boeing, S., Ouwersloot, H., Heus, T., Attema, J., Jansson, F., et al. (2021). jchylik/dales: Dales4.3_sb3 (version 4.3) [software]. https://doi.org/10.5281/zenodo.5642477

Van Laar, T. W., Schemann, V., & Neggers, R. (2019). Investigating the diurnal evolution of the cloud size distribution of continental cumulus convection using multiday LES. *Journal of the Atmospheric Sciences*, *76*(3), 729–747. https://doi.org/10.1175/JAS-D-18-0084.1

Veerman, M. A., van Stratum, B. J. H., & van Heerwaarden, C. C. (2022). A case study of cumulus convection over land in cloud-resolving simulations with a coupled ray tracer. *Geophysical Research Letters*, *49*(23), e2022GL100808. https://doi.org/10.1029/2022GL100808

Veikherman, D., Aides, A., Schechner, Y. Y., & Levis, A. (2015). Clouds in the cloud. In D. Cremers, I. Reid, H. Saito, & M.-H. Yang (Eds.), *Computer vision – ACCV 2014* (Vol. 9006, pp. 659–674). Springer International Publishing. https://doi.org/10.1007/978-3-319-16817-3_43

Villefranque, N., Fournier, R., Couvreux, F., Blanco, S., Cornet, C., Eymet, V., et al. (2019). A path-tracing Monte Carlo library for 3-d radiative transfer in highly resolved cloudy atmospheres. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2449–2473. https://doi.org/10.1029/2018MS001602

Zhan, Y., Yi, F., Liu, F., Zhang, Y., Yu, C., & Zhou, J. (2021). Convective boundary layer clouds as observed with ground-based lidar at a mid-latitude plain site. *Remote Sensing*, *13*(7), 1281. https://doi.org/10.3390/rs13071281

19422466, 2024, 3, Downloaded from https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003797 by Bibl. der Universitat zu Koln, Wiley Online Library on [12/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License