Workshop on Visualisation in Environmental Sciences (EnvirVis) (2014)
O. Kolditz, K. Rink and G. Scheuermann (Editors)

Short Papers

Animation of Clouds Based on the Interpolation of Weather Forecast Data

R. Hufnagel and M. Held

Deptartment of Computer Sciences, University of Salzburg, Austria

Abstract

We present a novel algorithm for interpolating discrete cloud data of numerical weather forecasts over time. The interpolation provides a continuous natural transition of the cloud properties over time intervals of several hours. A diffusion-free advection scheme transports cloud properties through the wind field. The algorithm is designed to be embedded in a rendering loop and provides the basis for employing cloud modeling techniques for creating temporal cloud animations of the future weather.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Clouds provide important visual clues for the perception of weather. Numerical weather predication (NWP) models supply detailed forecasts of future atmospheric conditions, but they do not model clouds explicitly. For presenting the abstract data to a lay audience, like, e.g., in TV weather shows, usually simplified 2D projections are used [SSK93]. A more recent approach creates realistic 3D clouds for presenting the future weather conditions in an intuitively understandable way [HHS07]. Their visualization allows fly-throughs but is restricted to static clouds.

We pick up their results and study the animation of clouds based on NWP data. This problem comprises two subproblems: First, the low-resolution NWP data has to be interpolated in time and space. In a second step, the spatially and temporally continuous cloud representation can be input to cloud modeling and visualization algorithms for creating realistic cloud images. In this work we focus on cloud interpolation on the basis of NWP data and introduce a diffusion-free temporal cloud interpolation scheme which consistently transforms the clouds over time.

1.1. Clouds in Numerical Weather Prediction

NWP data is usually provided as a set of 2D scalar fields where each field represents a certain forecast parameter on a layer of a 3D grid at a certain point in time. The grid either spans the whole Earth (e.g. $1760 \times 880 \times 64$ [GFS13]) or a specific region of interest (e.g. Central Europe at $421 \times 461 \times 50$ [COS13]). The geographic context of the numerical grid is given by a map projection (e.g. Mercator or Gaussian) and a certain level type (usually pressure, terrain-following or hybrid levels). The horizontal resolution ranges from 1 to 50 kilometer, the vertical resolution ranges from around 100 meters near the ground to 1 kilometer at the top of the troposphere.

Due to the coarse spatial resolution clouds, as we see them in the sky, are not directly represented. Rather they are described by their overall properties within a grid cell in terms of *cloud coverage* or *overall cloud water content*. The detailed geometry of the clouds is not provided. Only a coarse classification (convective and non-convective) indicates the type of clouds present in a cell of the NWP grid. The overall structure of the clouds is given by the spatial distribution of cloudy grid cells. Vertically adjacent cloudy cells can be regarded as a connected cloud volume containing clouds which span several levels.

The NWP model simulates atmospheric processes such as the transport of state variables by the flow (the so-called advection). Clouds are usually not advected by the simulation, but are rather derived in place from other forecast parameters. This is reasonable since clouds show a very inhomogeneous distribution, and interpolating cloud data during advection – inevitable on numerical grids – would smear out the data and alienate its information content. In addition, clouds are conceptually a result of atmospheric conditions

DOI: 10.2312/envirvis.20141106

which let water vapor become visible. Some sources that create this condition are advected by the wind (e.g., water vapor, temperature), while others are not advected (e.g., orographic lifting, evaporation sources).

1.2. Clouds in Nature

Clouds appear under certain atmospheric conditions when water vapor condenses and forms droplets or ice crystals. In the majority of cases this stems from cooling if air is lifted, caused by the decrease of pressure with height [Stu00]. Hence, clouds are created in ascending air and dissolve in descending air.

Clouds are embedded in the atmospheric flow and, thus, travel along with the wind. Flow patterns which create upward or downward motion can move faster or slower than the actual wind [Stu00]. That is, while clouds on a small scale (as seen in the sky) travel with the wind, the large patterns of clouds (like frontal cloud systems) do not necessarily move at the same speed. This discrepancy is possible since the clouds are subject to a constant development, showing cloud formation, deformation and dissolution, while being embedded in the atmospheric flow.

1.3. Cloud Interpolation for Visualization

NWP data is provided at intervals of three or six hours. A smooth animation of clouds over time is not possible at this resolution, but requires a temporal interpolation between consecutive forecast data times. The data provides the overall cloud coverage, a basic cloud type, a wind field, and some parameters which allow to derive certain cloud properties. The information "which cloud area refers to which area in the next data time" is not provided by the NWP data. This makes the temporal interpolation of cloud data difficult.

The usual way to create temporal cloud animations for the media is to interpolate the data fields pixel-wise by blending one data field into the next [SSK93]. Superimposed, animated noise and fast blending can be used to mask the obvious shortcoming. Still, such animations are unrealistic since they ignore the wind field completely, see Fig. 1.

Hence there is a strong demand for a practical yet more realistic solution. We stress that the interpolation necessarily involves a heuristic component. Our approach follows the following two key principles: (1) Compliance to the discrete NWP data at all given time stamps, and (2) a natural transition between two consecutive time stamps that takes into account the wind field. We achieve this by interpolating the cloud properties along wind trajectories and smoothly transforming the clouds along their path.

2. The Interpolation Algorithm

The cloud properties given by the NWP data represent the overall cloud pattern which moves according to the pro-

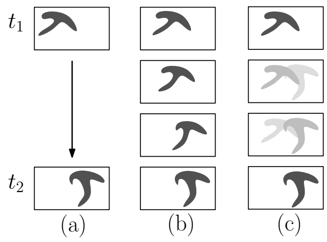


Figure 1: *A cloud coverage field representing, e.g., a frontal cloud system (a). Interpolation based on advection (b) vs. a pixel-wise, linear interpolation (c).*

cesses simulated by the NWP model. Our interpolation algorithm, on the other hand, yields entities for visualization which move along with the wind. These entities represent containers for visible clouds and are called *cloud cells* in the following. The interpolation algorithm creates the link between these two representations. It creates cloud cells resembling the NWP cloud properties at the begin of the interval. The cloud cells are displaced according to the wind and on their way they converge to a state resembling the NWP cloud properties at the end of the time interval; see Fig. 3.

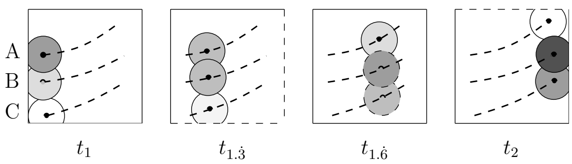
The transition of cloud properties along wind trajectories requires a transport of the properties with the flow from start time t_1 to the end time t_2 of a time interval.

2.1. Advection

Advection has been studied extensively in computational fluid dynamics. The three main approaches are characterized by their underlying spatial structure: grids (semi-Lagrangian approach [Bri08]), meshes (finite-volume method [Wen08]) and particles (smoothed particle hydrodynamics [DC96]). Advection on grids is subjected to numerical diffusion caused by repeated interpolation and is therefore not a good choice for the inhomogeneous cloud data. Meshes would be practical since we could use the initial regular grid and advect its vertices by the wind field. However, to compare values of the advected, deformed mesh at time t_2 to the regular grid of the next data time would require averaging due to the partial overlap of mesh cells, which is again undesirable. Particles transport flow properties without causing numerical diffusion [ZB05]. Hence we choose particles to transport the cloud properties consistently over time (and over time interval boundaries).

Particles are seeded within cloudy regions of the NWP data. The 3D cloud coverage field is sampled discretely (to avoid interpolation) and a particle is placed if a non-zero cloud coverage is encountered. The sampling resolution is chosen independently of the data grid resolution, eventually super-sampling the data grid. The cloud parameters provided by the NWP data at time t_1 determine a particle's initial state. Its final state is evaluated by tracing the particle through the

(a) Forward and backward particle tracing.



(b) Interpolation along trajectories.

Figure 2: *(a): Trajectories of particles seeded at t_1 traced forward to t_2 (left), and at t_2 traced backward to t_1 (right). (b): Interpolation of cloud properties along trajectories.*

wind field until the end of the data time interval and sampling the NWP data fields at time t_2 at its final position; see particles A and B in Fig. 2(a). Seeding particles only in cloudy regions at time t_1 does not account for clouds created after time t_1; see particle C in Fig. 2(a). Therefore particles are also seeded at time t_2. A sampling grid is used to bin the particles traced from t_1 at their final position at t_2. New particles are seeded in empty bins where a non-zero cloud coverage is encountered at the sampling positions. The trajectories of the new particles are calculated by tracing them back to t_1. For the advection we use a time step of a few minutes as common in atmospheric sciences. Based on the trajectories and the initial and final state we can interpolate the cloud properties along the trajectories; see Fig. 2(b).

2.2. Cloud Cells

Cloud cells represent containers for visible clouds of a certain type. The particles transport the clouds' properties and are assumed to be valid within an NWP model level and within a certain radius; see Fig. 3(a). Cloud cells are created by grouping particles of the same cloud type; horizontally in case of cloud layers and vertically in case of convective clouds. Initially each particle represents a cloud cell. Cloud cells on consecutive model levels which show a minimum horizontal overlap are merged to a single cloud cell. We use a slab cut ball [LAM09] as the bounding volume of a cloud cell; see Fig. 3(b). Cloud cells provide a cloud base and top that change consistently over time. They are displaced along the trajectories of the particles they enclose. On their way they may split up, e.g., due to different wind directions on different levels, and form new cloud cells carrying further on the cloud parameters of their initial cloud cell.

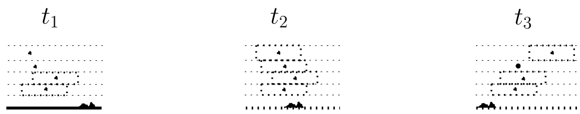
Details of the cloud volume, as needed for visualization, have to be created by cloud modeling techniques for each type of cloud (e.g. [EMP*03] for stratiform clouds, [DKY*00] for broken cloud layers or [Ney97,BN04, DKNY08] for convective clouds). These techniques can be applied for populating the cloud cell up to the cloud coverage given. The cloud cells store the cloud modeling parameters either implicitly (e.g. procedural noise parameters) or explicitly (e.g. bounding shapes) and the transition of the parameters for creating a continuous animation; see Fig. 3(c).

The transition of the clouds depends on the cloud type. Cloud cells changing the cloud type within a time interval either transform one cloud type into the other or dissolve the initial cloud and create a new cloud. For example, a broken cloud layer can be transformed into an overcast cloud layer by steadily extending the cloud patches. A convective cloud, on the other hand, grows vertically, starting at a fixed cloud base, independently of other clouds.

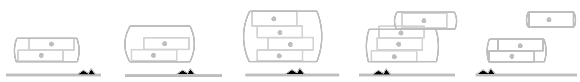
The transformation of clouds can be linked to parameters which are known to influence the cloud formation or dissolution. We propose two parameters: The vertical wind velocity (provided by the NWP data) and an orographic lifting parameter (positive or negative), derived from the horizontal wind field and the orography. The latter can be used to create spatially fixed clouds by choosing a fast cloud formation and dissolution rate and a dense particle distribution.

2.3. The Rendering Loop

The interpolation algorithm is designed to be embedded in a visualization process that allows arbitrary views and temporal animations. For improving the overall rendering performance we restrict the computation to the necessary parts



(a) Particles (black dots) with their radius of influence and slab height (black boxes) within their NWP model level (dashed lines).



(b) Cloud cells (black frames) displaced according to particle trajectories and with consistently developing cloud bases and tops.

(c) Detailed cloud volume modeled within the cloud cells.

Figure 3: *Cloud representations: (a): Cloud properties given at data times t_1, t_2 and t_3 are interpreted as a volume containing convective clouds. (b): Cloud cells describing a vertically growing, later a dissolving cloud volume. (c): Possible output of a cloud modeling technique.*

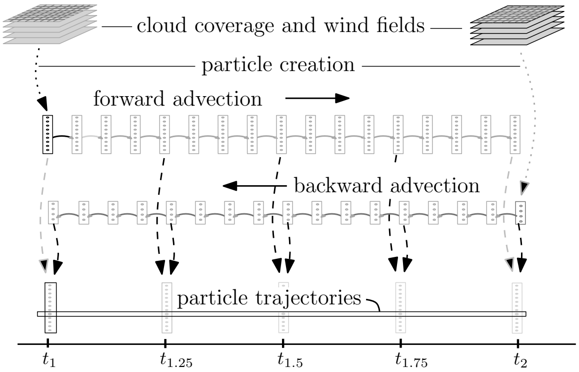


Figure 4: *Particle trajectory calculation.*

required by the current viewing frustum and light position; see the video in the complementary material [Web14].

The particles' trajectories are calculated by seeding, and forth and back tracing of particles as outlined in Fig. 4. The particle positions are stored for all rendering times within a data time interval and are re-used in consecutive frames. Additional computation on the trajectories is only required if a modified camera or light source bring in new tiles or when entering a new data time interval. Cloud cells are updated for each frame based on the current particle positions.

The cloud cells' bounding volumes can be used for a coarse lighting (e.g., for initializing an iterative lighting procedure) or for intersection tests in case of raytracing. The interpolation algorithm is independent of the actual lighting and rendering techniques employed. See, e.g., [HH12] for a recent survey of cloud lighting and rendering techniques.

3. Results and Discussion

The advantage of an advection-based interpolation over a pixel-wise, linear interpolation is exemplified in the videos available in the complementary material [Web14]. Linear interpolation creates a caterpillar-like cloud movement with large areas showing partial cloud coverage instead of a compact, moving cloud field; see Fig. 5. Accounting for the wind field not only yields compact cloud fields consistently moving in space, but it also reveals the complex atmospheric flow with varying winds at different levels; see Fig. 6

Employing particles for the interpolation of flow properties is generally applicable and better suited for inhomogeneous properties than grid- or mesh-based methods. The extraction of entities for visualization (here cloud cells) is, however, very application-specific since it incorporates an interpretation of the coarse, abstract data. The grouping of particles, e.g., is motivated by the observation that clouds grow vertically from a cloud base upwards.

While our algorithm is independent of the projection or the resolution of the NWP data, the extraction of cloud types and parameters for cloud modeling requires a specific interpretation for each NWP model. This is due to the different scales of atmospheric processes captured by the NWP models and the different cloud parameters provided.

Taking the maximum cloud coverage of particles within a cloud cell for creating the cloud base and using the cloud coverage in the upper part as a degree for how many of the cloud columns reach the top incorporates a relaxation of the data carried by the particles; see Fig. 3(c). This provides a better representation of the process captured by the NWP model than achieved by taking the data literally, e.g., by extracting an iso-surface of the cloud coverage distribution.

Our algorithm only distinguishes convective and non-convective clouds for classifying and grouping particles. Finding further visual cloud types [HH12] by taking into account other NWP model parameters (e.g., height, vertical velocity, wind speed and wind shear) or the geometry of connected, cloudy NWP grid cells and formulating meaningful transitions between those types remains future work.

Our interpolation along trajectories need not provide the expected cloud movement in all situations, though. Blocking of frontal clouds on mountain ranges, e.g., could be achieved by a dense particle distribution and a steered interpolation of the cloud properties. A robust approach to detecting such situations in the NWP data is, however, not obvious.

Summarizing, our algorithm is a first but important step towards closing the gap between real-world NWP data and established cloud modeling techniques. This opens the path for creating temporal cloud animations based on NWP data which are meteorologically sound, thus providing a major improvement for cloud animations used in the media.

Acknowledgements

Work supported by the Austrian FFG Grant #830029.

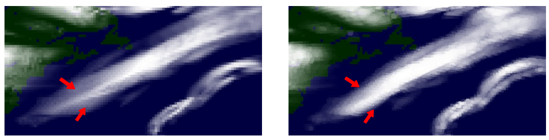


Figure 5: *Linear (l.) vs. advection-based (r.) interpolation.*

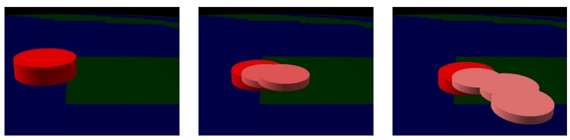


Figure 6: *A convective cloud cell (red) dissolving and spawning diverging cloud cells which are gradually transforming to a non-convective cloud layer (white).*

References

[BN04] BOUTHORS A., NEYRET F.: Modeling clouds shape. In *Proc. Eurographics Short Present.* (2004). 3

[Bri08] BRIDSON R.: *Fluid Simulation for Computer Graphics*. A K Peters, Ltd., Wellesley, Massachusetts, 2008. 2

[COS13] COSMO Model, Consortium for Small-scale Modeling, 2013. URL: http://www.cosmo-model.org/. 1

[DC96] DESBRUN M., CANI M.-P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *Eurographics Worksh. on Comp. Anim. and Sim.* (1996), EGCAS, Springer-Verlag, pp. 61–76. 2

[DKNY08] DOBASHI Y., KUSUMOTO K., NISHITA T., YAMAMOTO T.: Feedback control of cumuliform cloud formation based on computational fluid dynamics. *ACM Trans. Graph. 27*, 3 (2008), 1–8. 3

[DKY*00] DOBASHI Y., KANEDA K., YAMASHITA H., OKITA T., NISHITA T.: A simple, efficient method for realistic animation of clouds. In *Comp. Graphics (SIGGRAPH '00 Proc.)* (2000), pp. 19–28. 3

[EMP*03] EBERT D. S., MUSGRAVE F., PEACHEY P., PERLIN K., WORLEY S.: *Texturing & Modeling: A Procedural Approach*, 3rd ed. Morgan Kaufmann, 2003. 3

[GFS13] Global Forecast System (GFS), NCEP, NOAA, 2013. URL: http://www.emc.ncep.noaa.gov/GFS/. 1

[HH12] HUFNAGEL R., HELD M.: A survey of cloud lighting and rendering techniques. In *Proc. 20th Internat. Conf. Comput. Graphics, Visualizat., Comput. Vision* (2012), WSCG '12, pp. 53–63. 4

[HHS07] HUFNAGEL R., HELD M., SCHRÖDER F.: Large-scale, realistic cloud visualization based on weather forecast data. In *Proc. IASTED Int. Conf. Comput. Graph. and Imaging* (Feb. 2007), CGIM '07, pp. 54–59. 1

[LAM09] LARSSON T., AKENINE-MÖLLER T.: Bounding volume hierarchies of slab cut balls. *Comp. Graph. Forum 28*, 8 (2009), 2379–2395. 3

[Ney97] NEYRET F.: Qualitative simulation of cloud formation and evolution. In *Proc. Eurographics Worksh. on Comp. Anim. and Simul.* (1997), EGCAS, Springer-Verlag, pp. 113–124. 3

[SSK93] SAKAS G., SCHRÖDER F., KOPPERT H.-J.: Pseudo-satellitefilm using fractal clouds to enhance animated weather forecasting. *Comp. Graph. Forum 12, 3* (1993), 329–338. 1, 2

[Stu00] STULL R. B.: *Meteorology for Scientists and Engineers*, 2nd ed. Brooks/Cole, 2000. 2

[Web14] The project homepage, 2014. URL: http://www.cosy.sbg.ac.at/~held/projects/clouds/clouds.html. 4

[Wen08] WENDT J.: *Computational Fluid Dynamics: An Introduction*. A von Karman Institute book. Springer, 2008. 2

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. (Proc. SIGGRAPH '05) 24*, 3 (2005), 965–972. 2