

Feature Tracking in VR for Cumulus Cloud Life-Cycle Studies

E. J. Griffith¹, F. H. Post¹, M. Koutek¹, T. Heus² and H. J. J. Jonker²

¹ Data Visualization Group, <http://visualization.tudelft.nl>
Faculty of Electrical Engineering, Mathematics and Computer Science

² Department of Multi-Scale Physics, Faculty of Applied Sciences
Delft University of Technology, Netherlands

Abstract

Feature tracking in large data sets is traditionally an off-line, batch processing operation while virtual reality typically focuses on highly interactive tasks and applications. This paper presents an approach that uses a combination of off-line preprocessing and interactive visualization in VR to simplify and speed up the identification of interesting features for further study. We couch the discussion in terms of our collaborative research on using virtual reality for cumulus cloud life-cycle studies, where selecting suitable clouds for study is simple for the skilled observer but difficult to formalize. The preprocessing involves identifying individual clouds within the data set through a 4D connected components algorithm, and then saving isosurface, bounding box, and volume information. This information is then interactively visualized in our VR Cloud Explorer with various tools and information displays to identify the most interesting clouds. In a small pilot study, reasonable performance, both in the preprocessing phase and the visualization phase, has been measured.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Virtual Reality and I.3.8 [Computer Graphics]: Applications

1. Introduction

Feature tracking and virtual reality often have conflicting requirements. Virtual reality must be interactive to be effective and maintain a feeling of immersion. Feature tracking often requires reading and processing large, time-dependent data sets. Limited reading speeds from storage generally mean that feature tracking cannot be done in real-time. However, certain types of features are relatively simple to track in software, but it is difficult to determine which of those are actually interesting to researchers. In such cases, visualization enables scientists, whose backgrounds are in observational studies, to use their perceptual skills to select features worthy of further study. Traditional scientific visualization, though, can be awkward and unwieldy when dealing with 3D features and, more so, when those features are also time-dependent. VR, on the other hand, brings 3D data sets to life in three dimensions, which evokes a stronger perceptual response. This, coupled with the more natural and direct 3D interaction it provides, makes VR an attractive choice for dealing with 3D data sets. It is our claim that scientific vi-

ualization in virtual reality is the logical choice when aiding scientists in identifying the most interesting, 3D, time-dependent features to investigate further.

This paper arose out of our work on simulating cumulus clouds with Large-Eddy Simulations, or LES, and using VR visualization to explore the results. The long term goals of this collaborative project are to gain better insight into cloud dynamics through interactive exploration in VR. One of the first hurdles to be overcome in the project, though, is the selection of suitable clouds to study.

The size of the data sets and the nature of the clouds make it challenging to identify interesting clouds. The clouds develop unpredictably over the course of a simulation run, and it is common for pieces of clouds, or entire clouds, to merge together or split apart. These factors make it difficult to even track individual clouds through the data set. However, for an individual cloud to be interesting, it must satisfy certain qualitative criteria, such as being of sufficient size, going through the proper life-cycle stages, being a relatively co-

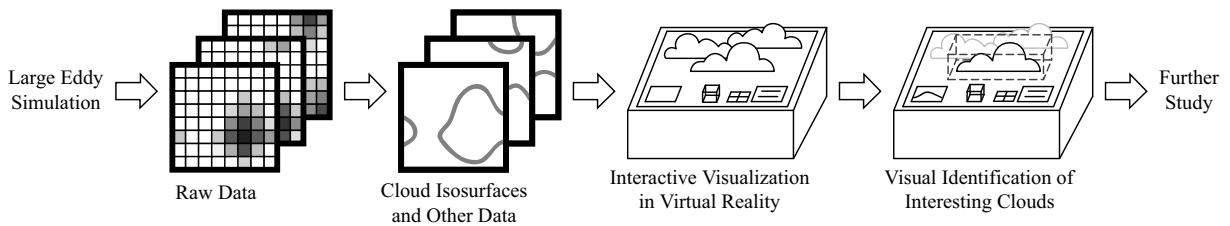


Figure 1: Raw data is generated by the LES, which, in turn, is processed to identify and track clouds in the data and produce isosurfaces and other data for them. The cloud isosurfaces and other relevant data are visualized in our virtual reality Cloud Explorer. A skilled user browses through the cloud field interactively and identifies interesting clouds for further study.

hesive, and not merging with other clouds. These properties are difficult to put into a consistent and automated algorithm. The situation is further complicated by data sets that are several gigabytes, and even up to 1 TB or more, in size. These difficulties are frustrating for atmospheric scientists because they are accustomed to selecting clouds based on observation. When they are presented with the opportunity to observe the clouds evolve over time, it is trivial for them to identify the most interesting clouds.

Traditionally, cloud selection for research has been accomplished through the painstaking efforts of atmospheric scientists. We have streamlined the process with a combination of preprocessing and interactive visualization in virtual reality. Individual clouds evolving over time are identified and tracked in preprocessing through a 4D connected components algorithm, and then isosurface, bounding box, and cloud volume data are generated. This data is then visualized interactively in our Virtual Reality Cloud Explorer, which provides various tools and information displays. Scientists using Cloud Explorer are then able to select certain clouds for further study. Using the connected components data, we are then able to extract only those portions of the data containing the interesting clouds. See Figure 1. This data can then be used to study the cloud dynamics over the course of the life-cycles by, for example, generating mass flux plots or velocity profiles. The authors have successfully employed Cloud Explorer in a small pilot study to identify interesting clouds, and reasonable performance, both in preprocessing and in visualization, has been measured.

The remainder of this paper is structured as follows. The next section provides an overview of feature tracking, LES, and other attempts to visualize clouds. The data preprocessing phase is discussed in Section 3. Cloud Explorer is described in Section 4. Some results are presented in Section 5, and the paper concludes with Section 6.

2. Background

2.1. Feature Tracking

Feature extraction is an approach to visualizing very large data sets. It entails detecting structures or objects within the

data that are of particular interest to scientists for a given research task. The features and their properties can be represented in a format that is both suitable for interactive exploration and much more compact than the original data. In fact, data reduction ratios of up to 10^3 or 10^4 can be achieved.

Feature tracking is the extension of feature extraction to time varying data sets. Applying feature extraction to time varying data sets involves more than just extracting the features from each time step. The correspondence between features in subsequent time steps must also be determined. By tracking the features across time steps, the temporal behavior of the features can be studied. A further step can be the detection of important changes in the lifetime of features. These changes, or so-called events, can be the appearance or disappearance of features, but they can also be the merging together or splitting apart of features.

Several approaches to feature tracking and event detection have been published. Correspondence between features can be determined by extracting the features directly from the 4D spatio-temporal domain [WB98] or by searching subsequent frames for corresponding features. This can be done by looking at the spatial extents of the features [SW97] or by calculating feature attributes and searching for the feature with the most similar attributes in the next frame [RPS01]. For a survey of techniques, see [PVH*03].

2.2. Large-Eddy Simulation

LES is a popular numerical tool in the atmospheric sciences to give insight into characteristics of flows in the Atmospheric Boundary Layer, or ABL, where observational methods, e.g. by satellites, airplanes or radar, are limited because they do not give access to the entire 3D flow field. In LES, the Navier-Stokes equations are solved up to a certain scale. In this way, the largest and most energetic flow circulations, or eddies, are resolved, and the influence of smaller eddies are approximated via a statistical model. This type of simulation was developed during the sixties and seventies, e.g. [Sma63], to solve turbulent flows for large (kilometer sized) domains and time scales while still taking into account the dissipation of energy, which takes place at scales of motion

around $\mathcal{O}(1\text{mm})$ and is of vital importance for the dynamics of the mean flow. Our simulations are performed by a parallelized version of the code described by Cuijpers [CD93]. The data sets used in Section 5 are cases of the Barbados Oceanographic and Meteorological EXperiment (BOMEX), which is documented in [SBB*03].

To simulate a field of cumulus clouds, which are relatively small clouds located in the ABL, a typical domain size is $6.4 \times 6.4 \times 3.2\text{km}$. The simulation keeps track of several variables: three velocity components, temperature, liquid water, and total moisture. These are updated in the simulation by integrating between time steps representing 2 seconds of real time. Using, for example, a grid size of 512^3 and outputting every third simulation time step, the data set produced for simulating 1 hour of real time will be more than 500 GB. Thus, the size of the dataset can be a disadvantage when attempting to investigate specific features of the flow. This is especially true for exploring the evolution of flow characteristics in time since this requires both a suitable cloud in space and knowledge of its history. While a thorough selection procedure can slim the dataset considerably, in many cases it is still insufficient. In those cases, a 3D, interactive environment like VR can be an excellent way to increase the amount of data that can be handled by a user, and thus also enable searches of the dataset for criteria that can be most readily identified by human perception.

2.3. Cloud Visualization

Many examples exist of cloud visualization. One early example ([KH84]) describes techniques for ray tracing volume density fields, resulting in fuzzy, cloud-like images. Later examples are mainly concerned with clouds for application in animations or games, aiming at visual realism using complex lighting models [RES*03, SSEH03], or high performance [HL01, Wan03]. In our case, we are mainly interested in the physical accuracy of the cloud simulation, although visual realism may be of some help to benefit from the observational experience of the atmospheric researchers.

3. Data Preprocessing

The first phase of our approach is data preprocessing, which prepares the raw data for visualization in VR (Figure 1). This phase proceeds in two major stages: feature tracking and iso-surface generation. For the purposes of this work, we refer to clouds as features in the data. We incorporate the usual feature detection step into the tracking phase because it is straightforward in our case. In the tracking phase, individual clouds in the data set are detected, identified and tracked through time. We also record data about the clouds during this stage. Once the tracking is complete, we generate iso-surfaces representing the cloud shape for later visualization.

The cloud data sets that we worked with are three dimensional volumes for each time step. The volumes are

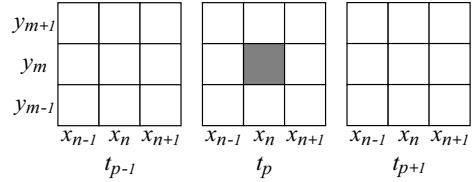


Figure 2: This figure depicts the 26 neighboring cells of cell (x_n, y_m, t_p) in a three dimensional, i.e. two spatial dimensions and one temporal dimension, binary array.

$128 \times 128 \times 80$ or $256 \times 256 \times 160$ grid cells in size. The volumes are periodic in both the x and y directions. This means that clouds may “wrap around” the sides as they move with the wind. For preprocessing, we are only concerned with one of the scalar quantities generated by the LES: the quantity of liquid water in the air, which is centered at each grid cell. A non-zero amount of liquid water indicates there is visible cloud in the grid cell, and we use this for cloud detection.

3.1. Cloud Tracking

Tracking clouds can be quite complex. Fortunately, atmospheric scientists are not interested in clouds that go through large-scale merging events. We take advantage of this to greatly simplify the tracking. We label all cloud masses that ever come in contact with each other as a single cloud. During the later visualization process in VR, scientists can quickly identify and disregard any undesirable “clouds” that result from multiple collisions or subdivisions.

To accomplish the feature tracking, we employ a four dimensional variety of a standard connected components algorithm. This type of algorithm is commonly used in computer vision. In such an algorithm, a binary array, of arbitrary dimension, is examined, and all array cells that have a 1 value are assigned a label based on their neighbors. Specifically, a cell will have the same label as its neighbor cells, and no two distinct sets of cells will have the same label. In this way, neighboring cells with 1 values are “clustered” together. For the four dimensional algorithm, we consider all 80 spatio-temporal neighbors. See Figure 2 for a simplified illustration. We construct the four dimensional binary array by considering each successive time step, and we assign a 1 value for each grid cell with a non-zero amount of liquid water and a 0 value otherwise. All cells in the binary array with a 1 value are considered part of a cloud. Two grid cells that are part of clouds, $v_1 = (x_1, y_1, z_1, t_1)$ and $v_2 = (x_2, y_2, z_2, t_2)$ with t_1 and t_2 representing time steps, are said to be neighbors, and thus part of the same cloud, if:

$$\max(|x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2|, |t_1 - t_2|) \leq 1.$$

The one caveat we must consider is that the x and y axes are periodic, whereas the z and t axes are not.

The 4D connected components is attractive for several

reasons. It will work in almost all cases, except for some small and uninteresting clouds, because of the slow cloud development and significant overlap between frames. It gracefully handles merging and splitting of clouds. This has the added advantage that it lumps large groups of merging clouds together, which allows them to be more readily ignored. It is computationally inexpensive, and it makes identifying corresponding features in adjacent time steps trivial. Furthermore, it offers fine-grained tracking, which other approaches like simple bounding box overlap may lack.

While creating the connected components, we also keep track of other data. We record the cloud volume by totaling the number of grid cells each cloud occupies in each time step. Additionally, we keep track of the bounding boxes for each part of a cloud in each time step. In a brief post processing step, these bounding boxes are enlarged for aesthetic reasons, and overlapping boxes are merged. These bounding boxes are used in isosurfacing and later visualization.

3.2. Isosurface Creation

Isosurface creation is a pipeline process. First, we prepare the data, and then we generate the initial isosurfaces with the marching cubes algorithm [LC87]. We refine the resulting triangle meshes with a series of filters, and finally we convert the refined meshes to triangle strips. We must take some care in this process, however. We want to keep the size of the isosurface data as small as possible to ease the burden on Cloud Explorer's interactivity, and we would like the clouds to have a more cloud-like appearance. Additionally, the triangle stripping algorithm we use requires manifold triangle meshes as input.

Creating the initial isosurfaces is fairly straightforward. First, an empty volume is created that is twice the size of the data volume in both the x and y directions. Next, the data for a particular time step is selectively placed into this empty volume. This is necessary for creating "complete," i.e. manifold, isosurfaces of clouds where clouds wrap around the periodic boundaries. By considering which cloud each grid cell belongs to and what the bounding boxes of that cloud are, it is possible to place the data for just one complete copy of each cloud into the volume. See Figure 3. Marching cubes can then be applied to this volume to generate triangle meshes representing the cloud shape.

After isosurfacing, the triangles are prepared for eventual storage and visualization. The first step is to separate the triangles into sets of meshes representing the individual clouds. By performing a look up into the connected components data for the centroid of each triangle, all the triangles can be separated into appropriate sets. See Figure 4. This produces a collection of sets of triangular meshes where each mesh represents a piece of a cloud and each set of meshes represents a single cloud. Next, we apply filters to the triangle meshes to both reduce the number of triangles and to improve the appearance of the mesh. See Figure 5 for the results of this.

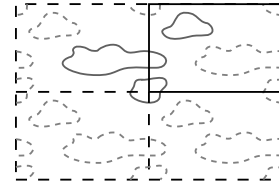


Figure 3: If the data for a time step is selectively placed into a larger volume in the fashion shown, each cloud will only appear in the volume once, and that one instance will be a manifold object.

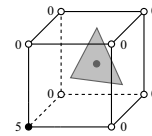


Figure 4: The centroid of the triangle lies within a cube where each corner of the cube is a grid cell. The black corner belongs to cloud 5 while the white corners do not belong to any cloud. Therefore, the triangle is part of cloud 5.

The last step before storing the meshes is to convert them to triangle strips to reduce the size of the stored data. We use a variation of the stripping algorithm presented by Gopi and Eppstein [GE04]. Their algorithm produces a single triangle strip if given a manifold triangle mesh as input, but it introduces new mesh vertices to do so. The cloud meshes are manifold, but we do not insert new vertices so we end up with multiple triangle strips. It is important to note, though, that the resulting triangle strips are just strips of adjacent triangles, and actually consist of alternating sequences of triangle fans and triangle strips (Figure 6). We keep track of when the strips alternate between fans and strips so that we are able to recreate the surface later with consistent winding.

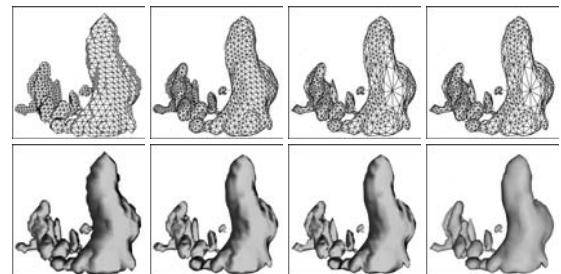


Figure 5: The top row illustrates the triangle mesh while the bottom row shows the corresponding shaded model. From left to right: raw output from marching cubes, after applying a windowed-sinc filter, after decimating the mesh, and after applying a normals filter.

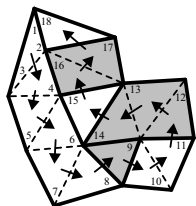


Figure 6: A strip generated by Gopi and Eppstein's algorithm [GE04] begins as a triangle strip, and then alternates between triangle fans and triangle strips. Triangle fans are indicated with gray triangles.

4. Virtual Reality Cloud Explorer

The second phase of our approach is interactive visualization in VR. Our VR Cloud Explorer application allows users to interact with evolving cumulus clouds produced by LES. The interface provides tools to allow users to focus on potentially interesting clouds. By examining these clouds more closely within Cloud Explorer, they can pick out certain ones to study further, e.g. for life-cycle studies. The application was built on top of OpenGL Performer and the RWB library, which is a custom VR library described in [Kou03]. In the remainder of this section, we present a sample scenario for using Cloud Explorer, and then we describe the components and relevant interactions in more detail.

4.1. Interaction Scenario

The user begins her interactions with Cloud Explorer once the visualization data is loaded and the application is running. She first watches the cloud field over the entire simulation run (Figure 7a). This gives her a feel for the clouds that are present, and she can quickly tell if anything went wrong with the simulation run. If the clouds seem reasonable, she proceeds to hide all clouds not going through the entire life-cycle as they are not of interest (Figure 7b). She continues to browse through time to spot potentially interesting clouds among those remaining visible. She selects one of those clouds to examine the plot of its volume over time (Figure 7c, Section 4.2). This gives her an overview of how the cloud evolves. She is looking for clouds with bell-shaped volume plots, which indicates that they go through the proper life-cycle stages. If it is acceptable, she limits the time playback to the cloud's life-cycle so she can focus on the time steps from just before the cloud comes into existence to just after it dies out (Figure 7d). Next, she hides all clouds except the cloud she is examining (Figure 7e). She then lets the application play through the cloud's life-cycle while she orients the cloud field and examines it from various perspectives (Figure 7f). If she is satisfied that the cloud is worthy of further study, she notes the number of the cloud down. Once she is done looking at a cloud, she shows all the clouds going through the complete life-cycle again, and she

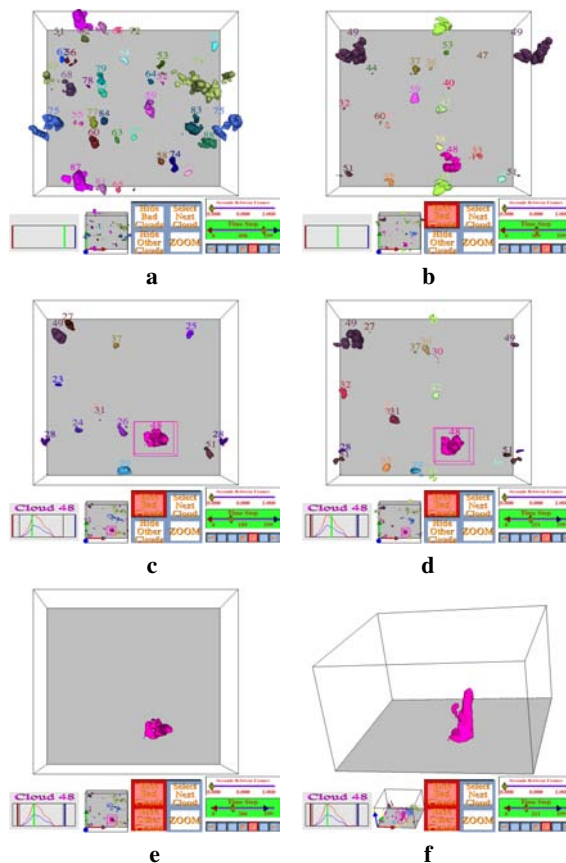


Figure 7: Cloud Explorer in various stages of use. See also Figure 8 and Section 4.2 for more details.

examines any other potentially interesting clouds in a similar manner. When she is finished with the application, she uses the numbers of the clouds she has written down, along with the connected components data, to extract just the portions of the data set that contain the interesting clouds. She can then perform various postprocessing steps on this data to further analyze the clouds, and she can also generate more detailed simulation results focusing on these particular clouds.

4.2. Application Components and Interaction

Cloud Explorer provides users with three interaction techniques: ray casting, direct manipulation, and world-in-miniature, or WIM. Ray casting and direct manipulation are used interchangeably for object selection and manipulation. The default interaction is ray casting, but if the tip of the interaction device, a stylus in our case, is within an object, then we switch to direct manipulation. Currently, we only use WIM as a method for orienting the cloud field.

The user's primary view on the data is the cloud field. This is the large box (Figure 8a), which displays the cloud

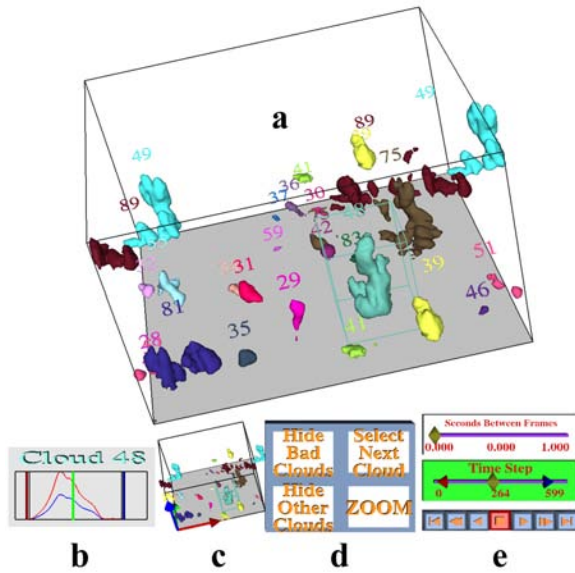


Figure 8: *Cloud Explorer* components: **a)** cloud field, **b)** volume graph, **c)** WIM, **d)** buttons, and **e)** time control panel.

geometries. In the default mode, each cloud appears in the cloud field in a distinct color. Above each cloud, a unique, identifying number is displayed for collaboration between multiple users and use in post processing. If the user has selected a particular cloud in the cloud field, that cloud will be displayed within its bounding box. The user also has the ability to hide certain clouds in the cloud field with the provided buttons (Figure 8d). In particular, he may choose to hide all clouds which do not go through the entire life-cycle or all clouds except the currently selected cloud. If he chooses to only display the selected cloud, then that cloud’s bounding box and identifying number will be hidden.

The user’s secondary view on the data is the volume graph. This panel displays two plots of the currently selected cloud’s volume over time (Figure 8b). One is relative to the cloud’s maximum volume, and the other is relative to the largest cloud’s maximum volume. The plots give users an estimate of the currently selected cloud’s relative size and behavior over time. When the user selects a new cloud, the plots are updated, and the identifying number of the selected cloud is displayed above the plots. To assist the user, the volume graph has several indicators. One indicates where along the time axis the current time step lies. Two others mark the cloud’s birth and death. The final two indicate which time span the playback is currently limited to.

For additional perspective and control, the user is provided with the world-in-miniature, or WIM (Figure 8c). The WIM and the cloud field maintain the same orientation, making the WIM a convenient tool for reorienting the cloud field. However, the WIM also plays a minor informational role

	Grid Size \times Time Steps	q_1 Size	Total Size
A	$(128 \times 128 \times 80) \times 600$	1.46 GB	8.76 GB
B	$(128 \times 128 \times 80) \times 1184$	2.89 GB	17.34 GB
C	$(256 \times 256 \times 160) \times 2169$	42.36 GB	254.16 GB

Table 1: *Three data sets consisting of three dimensional grids for each time step for each of six variables. For pre-processing, we are only interested the variable q_1 , which indicates liquid water. 600 time steps represent one hour of real time, with one time step every 6 seconds.*

since it always displays all the clouds in the current time step and indicates the selected cloud with its bounding box. In addition, the positive directions in each of the x , y , and z directions are indicated in the WIM to serve as a reference point should the user become disoriented. If the user finds the WIM unhelpful, he is able to remove it from view.

The time control panel gives the user control over the time dimension. It consists of two scroll bars and several video player-like buttons (Figure 8e). The top scroll bar controls the frequency at which new time steps are displayed, and the buttons control the time step sequence. The lower scroll bar gives the user more interactive control over which time steps are displayed. He can move the slider around to rapidly browse through time or jump to a time of interest. He can also limit the range of time steps that the playback cycles through. Whenever the slider or the limits are moved on this scroll bar, the corresponding indicators on the volume graph are also updated. In this way, the user can use a combination of the time control panel, the volume graph, and the cloud field to explore the evolution of the clouds over time.

Cloud Explorer provides a set of button widgets for extra functionality (Figure 8d). The buttons provide tools to change the current mode of operation for Cloud Explorer rather than methods to directly interact with the data. With these, he is able to zoom in or out on the cloud field, cycle through the clouds in the data set, and determine which clouds are visible in the cloud field.

5. Results

5.1. Preprocessing

We implemented the preprocessing phase as a multi-threaded, stand-alone application. The application is cross-platform, running on both Linux and Windows. We used three data sets (described in Table 1) to measure the application’s performance in three key areas: overall processing time, memory usage, and data reduction.

For each data set, we produced two sets of output. In all cases, we first identified and tracked all clouds in the data set via the connected components algorithm. We then used the data collected in this stage to filter some clouds out of

	CPU Time	Peak Mem. Usage	Avg. Mem. Usage
A	9m 44s	203 MB	156 MB
B	31m 24s	278 MB	223 MB
C	11h 38m 22s	2557 MB	1836 MB
A*	2m 49s	163 MB	117 MB
B*	6m 59s	238 MB	189 MB
C*	2h 04m 58s	1600 MB	1011 MB

Table 2: Processing time and memory usage for the three data sets. We used a dual 3.60 GHz Pentium Xeon Linux machine with hyperthreading and 3 GB of RAM, and the data was stored on a RAID0 system with read speeds up to 320 MB/s. 4 time steps were processed simultaneously. In those cases marked with a star, only clouds going through the entire life-cycle were fully processed.

	Number of Triangles	Output Size	Data Compression Ratio (vs. q_1)
A	11,731,822	165 MB	54 : 1 (9 : 1)
B	33,468,700	468 MB	36 : 1 (6 : 1)
C	517,110,152	6.9 GB	36 : 1 (6 : 1)
A*	2,493,716	39 MB	228 : 1 (38 : 1)
B*	6,456,066	99 MB	180 : 1 (30 : 1)
C*	46,824,994	679 MB	384 : 1 (64 : 1)

Table 3: The number of triangles, total output data size, and the resulting compression ratios when compared with the total input data set size and just the q_1 size. In those cases marked with a star, only clouds going through the entire life-cycle were fully processed.

the subsequent processing. In both sets, we eliminated all clouds having an average volume of less than ten grid cells per time step. In the second set of output, we also eliminated all clouds that did not go through the full life-cycle, i.e. those clouds present in the first or last time step. Table 2 relates the processing time and memory usage required for the preprocessing. Table 3 gives a sense of the content and size of the resulting data, as well as the compression ratios.

Eliminating clouds that did not go through the entire life-cycle yielded significantly smaller output, which is also reflected in the shorter processing time required. This is because each data set has one or two “super clouds”, which do not go through the entire life-cycle and are present throughout most or all of the time steps. These clouds consist of several clouds that continuously merge and split. Removing them means fewer triangles are generated in isosurfaces, which in turn means less mesh refinement is necessary.

5.2. Cloud Explorer

To test the effectiveness of Cloud Explorer, we performed a simple pilot study on our Responsive Workbench (Figure 9)

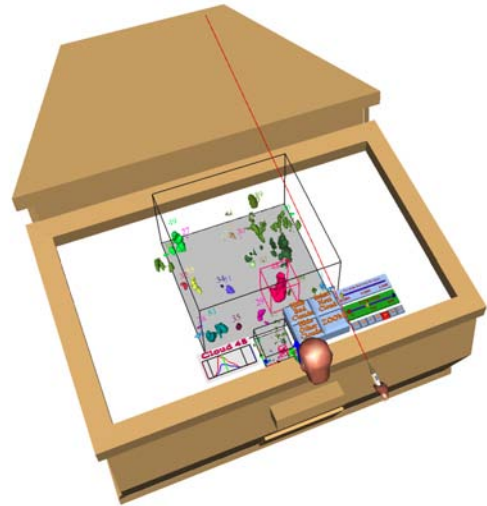


Figure 9: Playback of a recorded session on the Responsive Workbench.

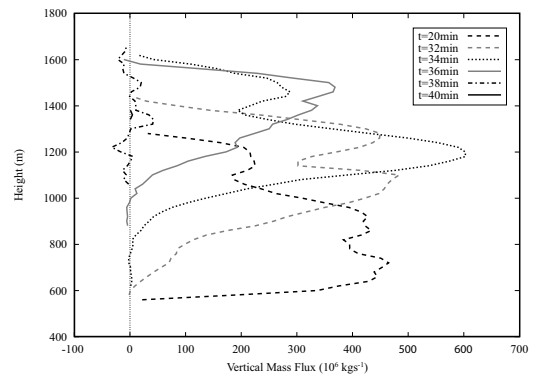


Figure 10: A mass flux plot for a cloud selected with Cloud Explorer. Positive flux (right of the zero line) indicates the cloud mass is moving upwards at that altitude. The cloud begins to decay at approximately $t = 32$ minutes. After this point, the cloud is no longer being fed by a thermal, and it drifts upwards and dies out. Interestingly, the profile retains most of its shape until the final minutes.

using data set C from Table 1. Currently, Cloud Explorer loads all of the visualization data into memory for performance reasons. This did not fit into memory so we, instead, confirmed the suitability of the data set by examining a handful of time steps of the full output, and then we worked with the output containing only clouds going through the entire life-cycle. Aside from these minor differences, our VR session proceeded similarly to Section 4.1. During the session, we experienced frame rates between 20 and 30 FPS.

Using Cloud Explorer, we identified seven clouds of particular interest. We then extracted the relevant parts of the

data set. Using this per cloud data, we generated mass flux plots indicating the motion trends within the clouds over their life-cycles. See Figure 10 for one of the plots. The resulting plots were quite consistent, and they demonstrated interesting profiles when the clouds were decaying. We feel that the consistency in the graphs is a good preliminary indicator that our approach is successful.

6. Conclusions and Future Work

Cumulus cloud simulations generate very large, time-varying data sets of up to hundreds of gigabytes. We showed how feature tracking and VR can help to deal with such large data sets and how it can help scientists to interactively select good clouds for further study, e.g. life-cycle studies.

The interactive selection is performed by first preprocessing the data and then interactively visualizing the results. Through the efficient representation of the clouds as iso-surfaces, sufficient data reduction is achieved to allow the interactive VR visualization as done in our Cloud Explorer application. VR has proved to be a very valuable tool for interactive exploration, allowing theoretical and observational considerations to be combined.

Having made their selections, the scientists extract the relevant parts of the data. They can then perform more detailed analyses of, for example, flow patterns and heat exchange at every stage of a cloud's life-cycle. This combination of simulation and VR visualization techniques will allow these types of studies to be done for the very first time.

This work is part of a larger project with many possible future directions. We hope to further formalize and automate the cloud selection process. We would also like to incorporate documentation facilities and new visualization techniques into Cloud Explorer. The aim of these steps is to further enable the atmospheric scientists to make use of their observational skills, though, rather than to eliminate them from the pipeline. Additionally, we will need to incorporate other large data handling facilities, such as out-of-core and multi-resolution representations, to cope with the full magnitude of the data sets LES can produce.

7. Acknowledgements

We would like to thank the Netherlands Organisation for Scientific Research (NWO) for providing project funding and for providing computer time at the National Computer Facility (NCF) at SARA for simulations.

References

- [CD93] CUIJPERS J., DUYNKERKE P.: Large-eddy simulation of trade wind cumulus clouds. *J. Atmos. Sci* 50 (1993), 3894–3908.

- [GE04] GOPI M., EPPSTEIN D.: Single-strip triangulation of manifolds with arbitrary topology. *Computer Graphics Forums* 23, 3 (2004).
- [HL01] HARRIS M., LASTRA A.: Real-time cloud rendering. *Computer Graphics Forum* 20 (2001), 76–84.
- [KH84] KAJIYA J., HERZEN B. V.: Ray tracing volume densities. In *Proc. ACM SIGGRAPH* (1984), pp. 165–174.
- [Kou03] KOUTEK M.: *Scientific Visualization in Virtual Reality: Interaction Techniques and Application Development*. PhD thesis, Delft University of Technology, 2003.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 163–169.
- [PVH*03] POST F., VROLIJK B., HAUSER H., LARAMEE R., DOLEISH H.: The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum* 22, 4 (2003), 773–790.
- [RES*03] RILEY K., EBERT, S. D., HANSEN C., LEVIT J.: Visually accurate multi-field weather visualization. In *IEEE Visualization 2003* (2003), pp. 279–286.
- [RPS01] REINDERS F., POST F., SPOELDER H.: Visualization of time-dependent data using feature tracking and event detection. *The Visual Computer* 17, 1 (2001), 55–71.
- [SBB*03] SIEBESMA A. P., BRETHERTON C. S., BROWN A., CHLOND A., CUXART J., DUYNKERKE P. G., JIANG H. L., KHAIROUTDINOV M., LEWELLEN D., MOENG C. H., SANCHEZ E., STEVENS B., STEVENS D. E.: A large eddy simulation intercomparison study of shallow cumulus convection. *J. Atmos. Sci.* 60, 10 (2003), 1201–1219.
- [Sma63] SMAGORINSKY J.: General circulation experiments with the primitive equations: I. the basic equations. *Mon. Weather Rev.* 91 (1963), 99–164.
- [SSEH03] SCHPOK J., SIMONS J., EBERT D., HANSEN C.: A real-time cloud modeling, rendering, and animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation* (2003), pp. 160–166.
- [SW97] SILVER D., WANG X.: Tracking and visualizing turbulent 3d features. *IEEE TVCG* 3, 2 (1997), 129–141.
- [Wan03] WANG N.: Realistic and fast cloud rendering in computer games. In *Proceedings of the SIGGRAPH 2003 conference on Sketches & applications* (2003), p. 1.
- [WB98] WEIGLE C., BANKS D. C.: Extracting iso-valued features in 4-dimensional scalar fields. In *Proc. Symposium on Volume Visualization* (1998), pp. 103–110.