ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (2022), Number 8

Cumulus Cloud Animation by Continuous Cellular Automata on 3D Scalar Field with Formation and Dissipation

William Trottier

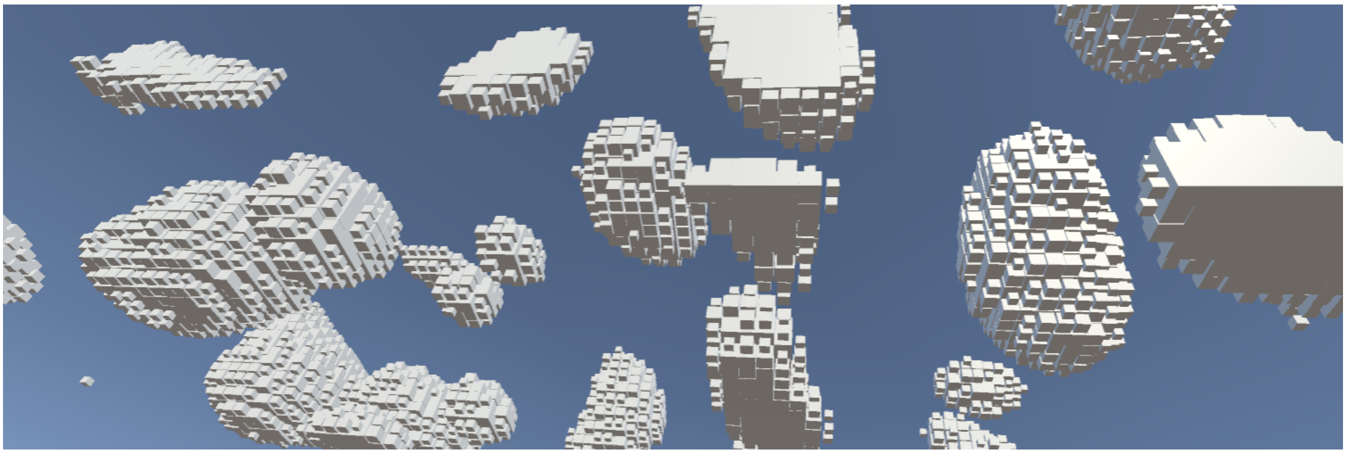

Figure 1: *Formed clouds by proposed method*

Abstract

Clouds play an integral role in maintaining realism for numerous outdoor scenes in computer graphics. Balancing realism and variety adds further complexity to an already complex task. Several noise-based approaches have been proposed, which provide cost-effective and moderately realistic clouds at the sacrifice of having unique cloud formations. Additionally, cloud formation and dissipation is lackluster for these solutions and tends to opt for a simple fade in and out approach. This project explores a method that utilizes noise and continuous cellular automata to create unique cumulus-type clouds with realistic formation and dissipation.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Simulation by animation;***

1. Introduction

Clouds play an integral role in maintaining realism for numerous outdoor scenes such as in movies and games. The absence of clouds can make an environment feel one dimensional and static and clouds allow a visual representation of the weather. Certain software such as flight simulators demand for realistic clouds and make use of a scalar field representation of air pressure which is directly correlated to the cloud density. In contrast, video games often choose noise-based methods due to their computational efficiency, resulting in simplistic and repetitive clouds with limited variation. The cloud type this report will be focused on is cumulus and cumulonimbus clouds since almost all of the other types are well represented by noise and have little room for variation.

This project explores a method that uses continuous cellular automata and random noise to produce cumulus clouds that are both unique and realistic. It will additionally have a strong focus on the formation and collapsing to further enhance the realism aspect.

2. Related Work

In *A survey of modeling, rendering and animation of clouds in computer graphics* [Gos21], Goswami explores both a procedural method and a physics based method to the animation of clouds. The procedural method looks at 3 sub-methods, one of which is rule-based / cellular automata. This method uses finite cellular automata and no noise. Additionally, bounding ellipsoids are used to control the cloud sizes and shapes.

In *Application of cellular automata approach for cloud simulation and rendering* [CIPMDSS14], Christopher Immanuel employs finite cellular automata in which each cell has 3 binary variables.

Both of these methods accurately and efficiently simulate clouds, however they tend to produce very structured clouds with low variation, and no flexibility in their output. To further these and allow for unique generation and scalar field representation, I propose a solution that incorporates continuous cellular automata and inject noise into the rule set to allow for unique cloud formations. This proposed solution will additionally allow for various adjustments in the output and ability to modify the variables to control cloud growth and collapse.

3. Overview

The base representation of the clouds will be a 3 dimensional scalar-field in which each cell has a humidity that can exist between 0 and 1. One cell is considered "adjacent" to the other if it is orthogonal, including diagonals, therefore for a 3 dimensional grid there are 26 cells considered adjacent. To avoid clouds collapsing at the edges and corners, the grid will wrap and all cells will have 26 adjacent tiles.

Since rendering is not the topic of this report, a simplistic rendering method will be used that displays each point of the scalar field as a cube in which the size is directly proportional to the humidity value. For example a humidity value of 0 being no cube rendered due to a size of 0, and a humidity value of 1 being a full-sized cube in that it takes up the entire volume of its cell. To avoid visual clutter, humidity values under 0.1 are not rendered.

3.1. Algorithm

The algorithm utilizes a continuous cellular-automata rule set and random noise to promote unique generation of clouds. The rule set consists of 1 accumulating rule, 2 decaying rules, and one clamping rule. Each of these rules contributions is multiplied by a scalar before being added to the previous states humidity. Equation 1 shows the calculation performed to get the next state of the animation from the previous. The specific scalars change depending on the environment that is trying to be represented. This equation then undergoes the clamping rule before being applied to the next iteration.

$$S_n = c_1 A(S_{n-1}) + c_2 D_1(S_{n-1}) + c_3 D_2(S_{n-1}) \tag{1}$$

3.1.1. Accumulation Rule

The accumulation rule utilizes a global wind vector that promotes clouds building in a certain direction. Each cell accumulates the adjacent cells' humidity to it and the contribution to how much it accumulates is determined by the similarity in the wind vector, and the displacement vector from the adjacent cell to the main cell. Similarity is calculated by the dot product, and each adjacent cell has a minimum contribution of 10 percent of its humidity.

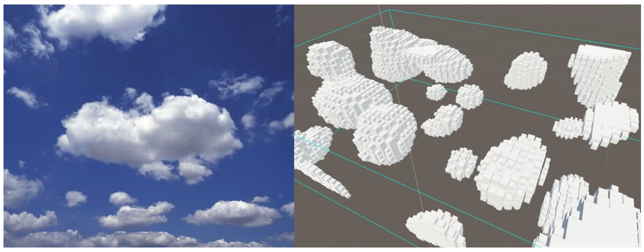

Figure 2: *Outcome of Simulated Clouds after Settling*

Figure 3: *Comparison of Cumulus Clouds (left) and Generated Clouds (right)*

3.1.2. Random Decay Rule

The random decay rule decays a cell by a random amount between the full value of the cell and 0. The purpose of this rule is to add randomness to the animation to promote interesting and unique generation and avoid converging on "ideal" shapes.

3.2. Fixed Proportional Decay Rule

The fixed proportional decay first calculates the mean humidity value across all cloud cells. This average is then negated from all cells equally. This rule acts to stabilize the average humidity and prevent overgrowth or over-decaying.

3.3. Clamping Rule

The clamping rule ensures that no values exceed the maximum humidity of 1 and ensures that a minimum amount of humidity is present in each cell. For all shown environments this minimum humidity value is kept at 0.07. This rule ensures that the adjacent sum will always accumulate something to prevent the simulation from decaying to nothing and being unable to rebuild clouds.

4. Evaluation

In this section, we will evaluate the outcome of this method of cloud generation in X ways: direct comparison of a frame with a reference

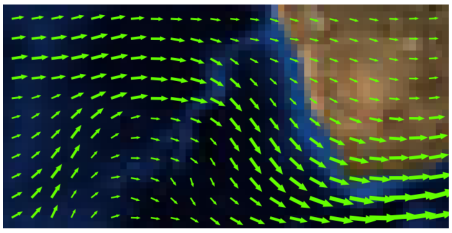

Figure 4: *Wind Vector Field Example*

picture, and comparison of the animation (including formation and dissipation) to a reference video. Evaluation will be quantified in these categories by giving it a rating from 1 to 5.

4.1. Visual Similarity by Frame Comparison

When viewing the comparison of the generated clouds and a reference photo of clouds as seen in figure 3 we can see that the generated clouds form correct sizing proportions in comparison to normal clouds. The clouds tend to look more spherical in the generated figure then the reference photo. Overall visually the clouds form similar volumes but with the drawback of having mostly spherical or elliptical clouds missing some of the wispy shapes you see in the other picture. This is partially in due to the nature of the adjacency sum and additionally the rendering method. Overall the visual similarity scores a 3/5.

4.2. Animation Accuracy by Video Comparison

When viewing the comparison of this method's animation of forming and collapsing clouds with the reference video [Str] we can see a strong similarity with how the clouds form and dissipate. There is a slight drawback observed that when the cloud animation is left at a stable state for a long period of time, clouds converge to an "ideal" shape where they continue to merge together as seen in figure 2. Overall, it receives a score of 3/5 for its strong formation and dissipation, and lack of accurate sustaining of clouds.

5. Conclusion

Numerous cloud generation methods exist, each with its own set of advantages and drawbacks in terms of factors like runtime, realism, and cloud diversity. Different industries often have to choose between the methods by which drawback can be worked around. The simulation method proposed in this work shows another method of cloud animation with additional advantages that tend to be overlooked by other generation methods such as diversity in clouds and realistic formation / dissipation. However, the algorithm still has some significant drawbacks that currently prevent its full implementation in systems.

5.1. Future Work

First improvement that could be made to this algorithm would involve changing the global wind vector to a vector field that is calculated by a mix of noise and a general wind direction. An example of this can be seen in Figure 4. This would give more control over how the wind behaves and also ideally act as a way of minimizing the settling of the clouds by "tearing apart" clouds that have formed together. A method to implement this could be to have a static vector field that points in the direction of the global wind vector, and a perlin noise field, and linearly interpolate between the 2 by the turbulence value.

Secondly, implementing a way to promote the existence of wisps of clouds off of main bodies would prove to be beneficial. This could be done by modifying the existing rule-set, or by adding additional rules on top of the existing. A method to implement this could be by adding a rule that functions similar to the main life/death rule of *Conway's Game of Life*[Gam70]. This rule works on a binary system and prevents cells from forming when there is too many cells around it by only forming a cell when there is exactly 3 neighboring cells. A similar rule could be implemented to this where a cell can accumulate a certain amount from its neighbors, and the remaining amount that it accumulates will be counted against the humidity value (i.e. count as negative).

References

[CIPMDSS14] Christopher Immanuel W., Paul Mary Deborrah S., Samuel Selvaraj R.: Application of cellular automata approach for cloud simulation and rendering. *Chaos (Woodbury, N.Y.) 24*, 1 (2014), 013125–013125.

[Gam70] Games M.: The fantastic combinations of john conway's new solitaire game "life" by martin gardner. *Scientific American 223* (1970), 120–123.

[Gos21] Goswami P.: A survey of modeling, rendering and animation of clouds in computer graphics. *The Visual computer 37*, 7 (2021), 1931–1948.

[Str] Strong M.: Time lapse of cumulus clouds near gallup, nm (4k). URL: https://www.youtube.com/watch?v=79vXAzW2yGY.