Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 5, Issue, 09, pp.2543-2546, September, 2013

INTERNATIONAL JOURNAL
OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

APPLICATION OF CELLULAR AUTOMATA APPROACH FOR CLOUD SIMULATION AND RENDERING

*,[1]Christopher Immanuel, W., [2]Paul Mary Deborrah, S. and [3]Samuel Selvaraj, R.

[1]Department of Physics, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai 600 062, Tamil Nadu, India
[2]Research Department of Physics, The American College, Madurai 625 002, Tamil Nadu, India
[3]Department of Physics, Presidency College, Chennai, India

ARTICLE INFO

Article History:
Received 24th July, 2013
Received in revised form
18th August, 2013
Accepted 26th August, 2013
Published online 14th September, 2013

Key words:
Cellular Automata;
Simulation;
Rendering;
Meta ball.

ABSTRACT

Current techniques for creating clouds in games and other real time applications produce static, homogenous clouds. These clouds, while viable for real time applications, do not exhibit an organic feel that clouds in nature exhibit. These clouds, when viewed over a time period, were able to deform their initial shape and move in a more organic and dynamic way. With cloud shape technology we should be able in the future to extend to create even more cloud shapes in real time with more forces. Clouds are an essential part of any computer model of a landscape or an animation of an outdoor scene. A realistic animation of clouds is also important for creating scenes for flight simulators, movies, games, and other. Our goal was to create a realistic animation of clouds.

INTRODUCTION

Clouds are an essential component of any outdoor virtual environment. They add an important element of visual detail without which the environment would feel unrealistic. The computer game industry has driven much recent research on the production of realistic looking clouds that one may fly around or through [2]. This work has produced several techniques for quickly rendering clouds that looks visually appealing but are not true cloud simulations. Methods range from billboard imposters to more complex procedurally generated objects. Often these rendering methods are limited to producing simple cumulus clouds. While cumulus clouds are pretty they represent only one of many distinct cloud types. We use cellular automata to model cloud dynamics, providing realistic looking clouds at real time rates. However, unlike previous physically based simulations, we also offer an ability to control the shape and appearance of clouds through custom shaping routines.

Motivations and Theoretical Background

The model studied in this paper was developed in an attempt to increase our understanding of complex behavior in dynamical systems. It is well known that dynamical systems differ dramatically in several important aspects of their behavior, such as periodicity, predictability, stability, dependence on initial conditions, etc. The cellular automata are the new style, high performance simulation tool. We hope researchers can bring into play well the modeling power of the CA approach in future in variety complex systems. Cellular automata have been successfully used to model many variety complex systems, such as biology, chemistry, mathematics, physics field. Formation and evolution of clouds can be simulated by a small amount of computation. The movement of clouds can be controlled by specifying the probability distributions for *hum*, *act* and *ext*. Simulation of the cloud evolution requires only a small amount of computation since it is executed by Boolean operations.

Corresponding author: Christopher Immanuel W., Department of Physics, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai 600 062, Tamil Nadu, India

Previous Work

Cellular Automata have been studied for decades but their connection with nonlinear dynamics is recent and well established only in the particular case of 1D elementary CA. Giovanni *et al*.[4]. The so called Nonlinear dynamics perspective of CA [1] is based on a simple observation: there are countless analytical results about nonlinear dynamics whereas the approach to CA is often empirical; thus, if we define a nonlinear dynamical system implementing a Cellular Automaton, then we obtain without effort many quantitative results about CA as well. The work related to cloud simulation and animation falls in two categories. An obvious approach for the simulation of clouds is to simulate the physical phenomenon itself. Although a successful approach, this can be time and computationally consuming. The second category involves heuristic approaches that are simpler to implement but may involve tweaking parameters by trial and error in order to produce a realistic animation. Dobashi *et al*. [3]. In terms of rendering, a simple well explored way to display clouds is to use mapping, but the resulting images are not photo-realistic. To deal with this problem work such as [5], [6] and [8] aimed at incorporating a physical model that takes into account phenomena such as absorption and scattering [3]. The work that we concentrated [3] describes a method that creates realistic cloud motion with minimal computation and fast rendering that involves using graphics hardware in order to render shafts of light. As mentioned above, we decided to pursue the method described. Dobashi *et al*.[3]. It combines simulating the physical processes of fluid dynamics and using a heuristic approach in order to achieve realistic cloud formation and motion. The method allows somewhat easy control of the shapes and motion of clouds as well as rendering of shafts of light that the sun light creates when passing through gaps in the clouds and being dispersed by particles. The authors base their work on a method. Nagel *et al*.[7] that simplifies cloud dynamics and improve it by incorporating cloud extinction in the model.

Simulation Process

3D cellular automaton for cloud simulation

The cloud evolution process can be summarized as follows. A bubble heated by underlying terrain heat rises to the sky due to its lower density. In the sky, where the air pressure is low, the bubble cools down during its expansion, which causes the relative humidity inside to increase. Then, a phenomenon called phase transition occurs, which changes the water vapor in the bubble into water droplets or clouds. Nagel *et al*. Use cellular automaton to simplify the cloud dynamics. The simulation space is divided into voxels (e. g. 256×256×10) to represent 3D model for clouds. Each voxel is considered to be a cell in the cellular automation. Three logical variables, hum for humidity/vapor, act for activation/phase transition factor, and cld for clouds that are assigned to each cell, are used to show the status of that voxel in the whole cloud evolution process. The value of each variable is either 0 or 1. hum= 1 means there is enough vapor to form droplets/clouds; act= 1 means the phase transition occurs; cld= 1 means clouds have been formed. The 3D voxels models with three variables, as well as the cloud evolution process are shown in the Fig.1.

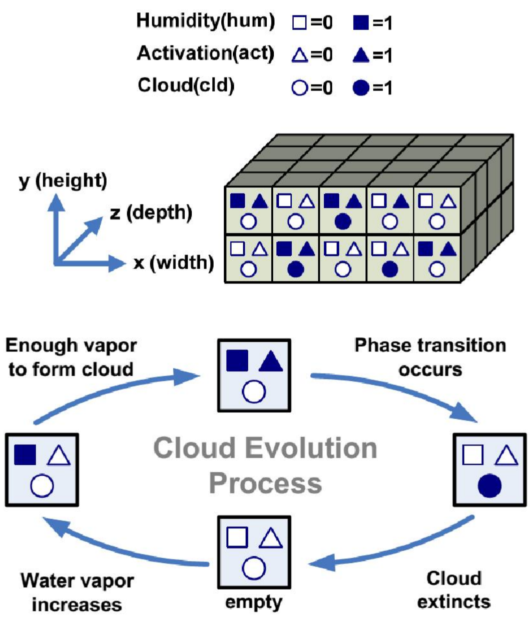

Fig.1. Cloud evolution process

Complex cloud motion simulation

Cloud transition

The rules for the transition of the three variables are as follows:

If currently there is enough humidity, and the phase transition has not yet occurred, it remains humid. If the phase transition takes place, the vapor will change into water, and then clouds form. Whether the phase transition will occur depends on the humidity and the transition factor in the neighboring area. The Boolean functions for the transition of the three variables are shown in the following fig.2.

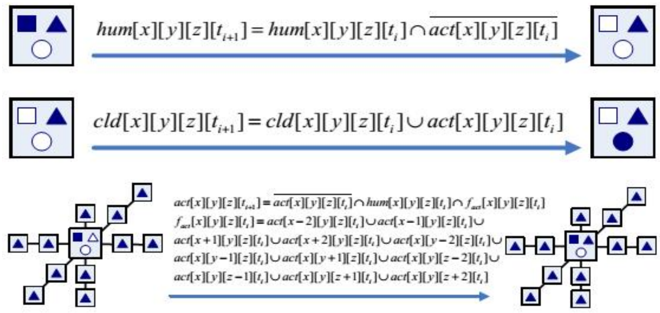

Fig.2. The Boolean functions for the transition of the three variables

Cloud formation and extinction

From the cloud transition formulas above, we can see that once *cld* becomes 1, it will remain 1 forever, which means cloud extinction never occurs from Fig.3. Similarly, once the cloud extinction occurs at a cell, no cloud will be generated there anymore. This will make the cloud simulation unnatural.

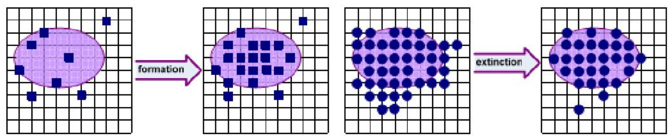

Fig.3. Cloud Formation and Extinction

To solve this, each cell is given an extinction probability P_{ext}, a vapor probability P_{hum}, and a phase transition probability P_{act} specified by the animator, to set the *cld*, *hum* and *act* value randomly. At each cell where *cld* is 1, a random number *rnd* between 0 and 1 is generated. If $rnd > P_{ext}$, *cld* is clear to 0. By varying the P_{ext} probability at each cell at different times, regions where clouds frequently extinct can be specified. In a similar way, if $rnd < P_{hum}$, *hum* is set to 1, and if $rnd < P_{act}$, *act* is changed to 1. Cloud formation and extinction can be controlled by adjusting the probabilities, P_{ext}, P_{hum} and P_{act} at each cell at each time step.

The following figure shows the Boolean functions and a screen shot for cloud formation and extinction Fig.4.

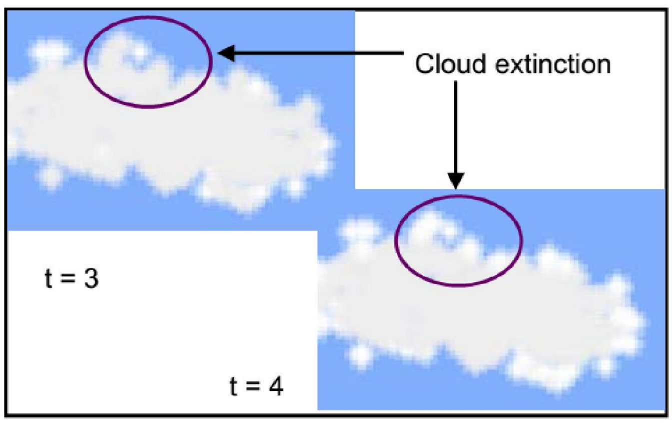

Fig.4. cloud formation and extinction

$$cld[x][y][z][t_{i+1}] = cld[x][y][z][t_i] \cap Is(rnd > P_{ext}[x][y][z][t_i])$$
$$hum[x][y][z][t_{i+1}] = hum[x][y][z][t_i] \cup Is(rnd < P_{hum}[x][y][z][t_i])$$
$$act[x][y][z][t_{i+1}] = act[x][y][z][t_i] \cup Is(rnd < P_{act}[x][y][z][t_i])$$

Our method of Cloud Simulation based on Cellular Automata with Rule 30.

$$a_{i-1}^{(t)} = a_{i-1}^{(t)} \text{ XOR } [a_i^{(t)} \text{ OR } a_{i+1}^{(t)}] \quad (1)$$

$$a_i^{(t+1)} = a_{i-1}^{(t+1)} \text{ XOR } [a_i^{(t+1)} \text{ OR } a_{i+1}^{(t+1)}] \quad (2)$$

$$a_{i+1}^{(t+2)} = a_{i-1}^{(t+2)} \text{ XOR } [a_i^{(t+2)} \text{ OR } a_{i+1}^{(t+2)}] \quad (3)$$

Wind advection

In the natural world, clouds may move in a certain direction as a result of the blowing winds. This wind advection can be simulated simple by shifting the variables towards the direction of the wind Fig.5.

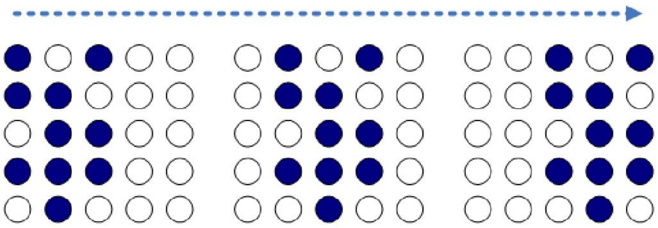

Fig.5. Wind Direction.

Simply assume that the wind blows following the direction of x-axis, and the wind velocity, $v(z_k)$ is considered to be a piecewise linear function of the z coordination of each cell. The wind advection effect can be represented as follows.

$$cld[x][y][z][t_{i+1}] = cld[x - v(z_k)][y][z][t_i] \cap Is(x - v(z_k) > 0)$$
$$hum[x][y][z][t_{i+1}] = hum[x - v(z_k)][y][z][t_i] \cap Is(x - v(z_k) > 0)$$
$$act[x][y][z][t_{i+1}] = act[x - v(z_k)][y][z][t_i] \cap Is(x - v(z_k) > 0)$$

Computation acceleration

Bit field manipulation functions

If we traverse each voxel one by one, it is time consuming. In order to reduce the computational overhead, we make a use of the bit field manipulation functions of the higher level language (C++) we code with. Instead of using a Boolean or integer for each variable, values of 32 voxels are grouped together into an unsigned long number since it is 32-bit long. That is to say, the original WIDTH × HEIGHT × DEPTH × TIME 3D voxel model becomes (WIDTH/32) × HEIGHT × DEPTH × TIME. When applying Boolean functions to the unsigned long variables, 32 voxels are processed at the same time.

Ellipsoids cloud region determination

We specify different cloud region using ellipsoids. The position and size of the ellipsoids are set randomly. The cloud is more dense in the center and sparse in the peripherals, and it is initialized based on this. Also, the cloud formation is more likely to take place in the center while the extinction has a higher probability to occur in the peripherals. According to these observations, for each cell, we predetermine the vapor, activation and extinction probabilities according to the distance from the center of the ellipsoid.

Rendering

We use meatballs and billboards in order to represent cloud density and to account for the attenuation of light through the voxel space. Each voxel cell is assigned a metaball which in its turn has an associated billboard. The billboard stores the density of the metaball, which decrease from the center of the metaball towards the periphery. The density of the billboards along with pre-calculated texture maps are used to determine the color of a cloud.
In order to reduce computation time, the density is discretized into 64 textures that are precomputed. At render time, the texture with density corresponding closest to the density at (x, y, z) is mapped to the billboard with center (x, y, z). The color within a single texture map is consistent, but the transparency increases from the center of the billboard to its periphery as illustrated below in Fig.6.

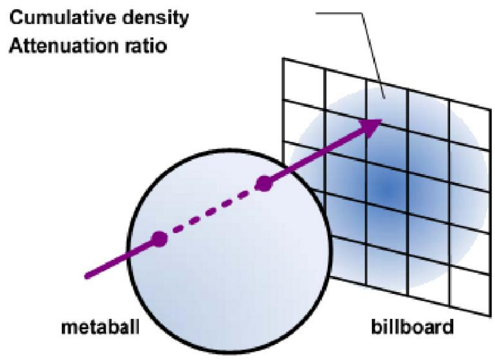

Fig.6. Transparency increases from the center of the billboard

The distribution of the cloud density obtained by the simulation is binary. A cloud either exists or it does not. In order to obtain a continuous distribution of cloud density which is used to calculate the color of the clouds, we used the field function (graph below) suggested. Dobashi *et al*.[3] and proposed. Wyvill *et al*.[9]. The density of a point (x, y) is determined as a weighted function of the densities of the neighboring cells, which allows smoother transition in density/color.

Discussion

Determine formation/extinction probabilities

Determining the probabilities for the various events we were trying to achieve, for example probability for extinction and formation of the clouds, proved to be a challenge. Assigning values to the effect parameters is not intuitive; minute changes to any of the parameters bring about extreme changes in the scene. For example, changing the extinction rate by 0.001 can cause a change from cloudy scene to perfectly clear sky.

Interpolate density between time steps for realistic effects

Currently the time step does not correspond to animation frames, which causes the changes between different frames to appear too fast. For example, the extinction and formation of clouds may be somewhat too obvious between two consecutive frames. To fix this problem we would have had to calculate the density at animation frame at time t by interpolating between the densities at time steps t_k and t_{k+1}.

Shaft of lights

An additional feature that could improve our cloud model would be implementing the shafts of light described. Dobashi *et al.*

Conclusion

This paper has proposed a method for simulating the dynamic movement of clouds using CA and demonstrated its usefulness by realistic animations. The proposed method has the following features. The simulation requires a small amount of computation since the dynamics of clouds is expressed by a simple Boolean operation. The movement of clouds can be controlled by specifying the probability distributions for *hum*, *act*, *ext*. The continuous density distribution is obtained by using metaballs. This makes it possible to create a realistic animation.

REFERENCES

[1] Chua, L. O. 2009. Nonlinear Dynamics Perspective of Wolfram, New Kind of Science. Volumes I, II, and III (World Scientific, Singapore) 57.

[2] Christopher Immanuel W, Paul Mary Deborrah S and Samuel Selvaraj R. 2013. Cellular Automata With Chaotic Rule Formation and Its Characteristics, Asian Pasific. J of Research, 1:97-113.

[3] Dobashi, Y., Kaneda, K., Yamashita, H., Okita, T. & Nishita, T. 2000. "A simple, efficient method for realistic animation of clouds," Proc. of the 27th Annual Conference on Computer Graphics and interactive Techniques, ACM Press/Addison-Wesley Publishing Co., New York, NY.pp.19-28.

[4] Giovanni E. P. 2012."Complexity Index of Outer-Totalistic Binary Cellular Automata With Arbitrary Dimension And Neighborhood," Int. J. Bifurcation and Chaos 22: 1-12.

[5] Kajiya, J., Herzen, B., 1984. "Ray Tracing Volume Densities," Computer Graphics, 18: 165-174.

[6] Max, N. 1994."Efficient Light Propagation for Multiple Anisotropic Volume Scattering," Proc. of the Fifth Euro graphics Workshop on Rendering. pp. 87-104.

[7] Nagel K., Raschke E. 1992. "Self-organizing criticality in cloud formation", Physica A 182: 519-531.

[8] Sakas, G., Gerth, M. 1991. "Sampling and Anti-Aliasing of Discrete 3-D Volume Density Textures ", Proc. of Eurographics'91, pp. 87-102.

[9] Wyvill, G., Trotman A. 1990. "Ray-Tracing Soft Objects," Proc. of CG International. pp.439-475.

****