

TWO METHODS FOR CLOUD VISUALISATION FROM WEATHER SIMULATION DATA

Andrzej Trembilski

Abt. Visualisierung und Virtuelle Realität
Fraunhofer Institut für Graphische Datenverarbeitung
Rundeturmstr. 6
64283 Darmstadt, Germany
Tel: (+49)(0)6151/155-287
Fax: (+49)(0)6151/155-196
trembils@igd.fhg.de www.igd.fhg.de/~trembils/

ABSTRACT

For the local TV presentation of weather forecast data it is important to have high-quality and fast visualisation of clouds. In this paper I present methods for the visualisation of clouds from data produced by a meteorological weather simulation. Isosurfaces, which are originally too coarse because of the data grid resolution are refined and deformed. The resulting geometry is used for cloud visualisation.

Keywords: Meteorological visualisation, cloud visualisation, surface refinement.

1. TV-PRESENTATION OF WEATHER SIMULATION DATA: PROBLEM DESCRIPTION AND MOTIVATION

Numerical weather simulation models, like these of the German Weather Service (DWD), produce once or twice a day large quantities of simulation data, which must be visualised for the presentation on television.

Visualisation systems like TriVis are used to visualise the weather forecast in the overview for whole countries. Special weather phenomena, which occur only locally, would disturb and are not desired in such a general view. Therefore the realistic representation of clouds is not so important, because single clouds would appear too small in the global view anyway. Global cloud fields are visualised for the presentation of the general weather situation in a bigger region.

The requirements change as soon as one tries to make a local weather presentation. New requirements result from the combination of the techniques of Augmented Video and Scientific Visualisation. If weather visualisation should be integrated into a videofilm, the schematic, global representation of the clouds is no longer acceptable. On the one hand the spectator can compare the artificial clouds with the genuine world presented in the video. Any break in the representation diverts

from the actual information and irritates the spectator. On the other hand only a realistic cloud representation yields understandable weather forecasts.

The actuality of the data in the presentation is an important restriction for the selection of the visualisation methods. For a film of only one minute 3000 images must be rendered and stored quickly enough to guarantee that the resulting TV presentation is still up to date. If the production of a frame takes only one second, 50 minutes are needed for the film. For practical applications the approximate rendering time should not exceed 1-3 seconds.

Such rendering times can only be achieved with the help of 3D graphics hardware. The common way is to use OpenGL graphics accelerator hardware which is available at low prices these days. The main disadvantage is the need to use polyhedral models for the visualisation.

This paper presents two methods to generate appropriate polyhedral cloud models from isosurfaces generated by the Marching Cubes algorithm.

2. RELATED WORK

The concept of representing clouds as textured, semitransparent and in fact stiff ellipsoids was proposed by Gardner [Gardner85]. Knöpfle

simplified this approach for the integration of weather forecast data in a virtual environment [Knöpfle97]. He achieved a real-time visualisation of cloud fields by using textured polygons for the top and bottom layer of the clouds. Since these layers are just as low resolved as the data, the resulting clouds appear very schematic.

A new technique was developed for high quality rendering of gas clouds. In the volume rendering approach, the 3D world is subdivided into voxels and the local gas density is computed using spectral turbulence [Stam93, Sakas93, Ebert98]. After these calculations are finished one has to render the scene with special ray tracers. The disadvantages are obvious: ray tracing is too slow for interactive applications and the amount of data for complex scenes is too large to handle. Gamito, Lopes and Gomes developed a particle system based method to simulate turbulence phenomena in gases and liquids [Gamito95]. They used two particle systems: one for the medium in which the particles are moving and one for the particles themselves. The interaction of these two particle systems resulted in the turbulence of a whole cloud. Although the results were promising this approach is generally not suited for visualisation of precalculated simulation data. [Unbescheiden98] presents a particle-based method for the visualisation of smoke in Virtual Reality. It models the movement of the smoke cloud and guarantees that a moving observer gets a correct optical impression especially inside the smoke cloud, but it is also not suitable for the visualisation of any given simulation data

[Koppert98] and [Haase99] are general descriptions of current meteorological visualisation techniques and operational software used by the DWD.

[Schröder97] is a detailed description of the methods for the visualisation of meteorological data and especially of the TriVis visualisation system. However, the introduced techniques are developed for global weather presentation and are not suitable for local presentation with augmented video.

3. CLOUDS FROM ISOFACES

SIMULATION DATA

First it is important is to understand, how much data must be visualised at all. Especially, we must understand how far an observer can look into a completely clear atmosphere, even from a possibly higher position.

Based on Fig. 1:

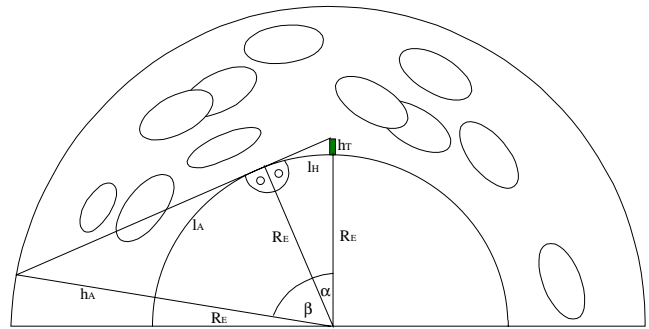


Fig.1: How far could we look into the atmosphere if it was absolutely clear.

with

$R_E \cong 6370$ km, radius of the earth

h_T = Height of the observer position

$h_A \cong 18$ km, relevant height of the atmosphere for clouds, see [WMO87]

l_H = Distance (on the surface) between the observer position and the horizon

$l_A + l_H$ = Distance (on the surface) of the observer position and the last point of the atmosphere, where clouds can become visible.

The distance on the earth surface is:

$$l_A + l_H = R_E \left(\arccos \frac{R_E}{R_E + h_A} + \arccos \frac{R_E}{R_E + h_T} \right)$$

hT [m]	lH [km]	lA+lH [km]
1.8	4.8	483.1
10	11.3	489.6
50	25.2	503.5
100	35.7	514.0
200	50.5	528.8

Table 1: How far could we look into the atmosphere if it was absolutely clear

Without any obstacles we can see a horizon with a radius of 35.7 km from a 100m tower. The visible piece of the atmosphere is however still larger (see fig. 1), with a radius of 514 km. However, this is only the theoretically visible part of the atmosphere. The radius can be further limited, assuming that even on best days the atmosphere is never quite clear and thus the diameter of the visible area will surely remain limited to 200 km. Additionally this is absolutely sufficient for the needs of the local weather forecast.

Using the model of the German weather service with the mesh size of 2.5 km for the local weather

forecast, we get a grid of 160x160x35 data points, i.e. 896000 data points.

GEOMETRY REFINEMENT

An example of an isosurface calculated from such a data grid is presented in fig. 3. It does not like a cloud, parts of it are very sharp-edged, another parts are very flat. The prototype methods, presented in the following, were developed to adapt these forms to what is perceived as a cloud. Figures 4-9 illustrate their visual results.

PRECOMPUTATION

The Marching Cubes algorithm supplies closed, coherent polygonal cloud geometries for the two following algorithms.

METHOD 1: FRACTALS

With this method cumulus clouds can be produced from given isosurfaces. The algorithm is similar to classical fractal algorithms for the production of terrain models.

PART A: DISTORTION OF THE ORIGINAL MESH

The vertices of isosurfaces generated by the Marching Cubes algorithm are very regularly distributed to in horizontal direction due to the regular horizontal data grid. To generate convincing clouds this regular structure must be distorted by random horizontal shifts of the vertices.

PART B: MIDPOINT DISPLACEMENT

All given triangles are then subdivided by subdividing every triangle edge. In this way the algorithm produces four new triangles from every triangle.

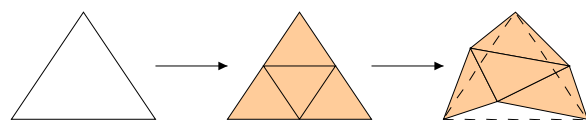


Fig. 2: Midpoint displacement.

Every new vertex is shifted along the normal (N) of the old edge. N interpolates the normal vectors of the adjacent triangles of the current edge. This shift is also influenced by other 3 parameters:

d = user-defined distortion factor
l = length of the subdivided edge
r = random number $\in [0,1]$

The resulting formula is:

$$P = P + d \cdot l \cdot r \cdot N$$

In the next iteration all new edges are partitioned. The procedure can be repeated until the desired degree of detail is achieved. In practice 3-4 iterations proved as perfectly sufficient.

Fig. 4-6 show the cloud geometries produced by this algorithm. I found that they have a strong similarity to cumulus clouds.

METHOD 2: COPY & BLOW UP

The idea of this method is to produce an irregular geometrical form by Brownian movement of the vertices from a smooth isosurface. During the procedure the isosurfaces are dissolving into single triangles, in order to produce a diffuse looking shape.

First, two additional, identical copies of the isosurface are produced. For each of the three resulting isosurfaces, a different scattering factor is specified. Then the vertices are shifted with Brownian particle movement.. The scattering factor decides how far the point can be shifted. Different transparencies are then determined for each of the three isosurfaces. The isosurfaces with small scattering factor receive a low transparency, the isosurfaces with high scattering factors receive a high transparency. In this way the outer parts of the clouds become more transparent and there is a smooth transition between the cloud and the air. In order to mask the single triangles, each triangle receives a semitransparent texture. The texture itself is just white but has an increasing transparency from its inner to the outer side.

Fig. 7 and 8 show the visual results of the algorithm used with isosurfaces, fig. 9 the result of the algorithm used with the output of method 1.

I found the results produced with this algorithm comparable to altocumulus and cirrostratus clouds.

The method can be used on original isosurfaces and cloud geometries produced with the fractal method (1).

COMPUTING THE ILLUMINATION OF A CLOUD

A density function is used for the lighting of the cloud. For each triangle, a ray is sent towards the sun position. Then, the number n of the intersections

with other triangles is determined. Rays computed for triangles on the side where the sun shines have only few or no intersections at all, rays computed for triangles on the other side of the cloud have several intersections. This method also allows to compute shadows of clouds on other clouds.

The exact distance within the cloud cannot be determined with this method, because the triangles do not form a closed surface anymore due to the Brownian particle movement. The light intensity of a triangle is determined from the number of n of the intersections with the following formula:

$$I = \min + (1 - \min) \cdot e^{-kn^2}$$

with

\min = the minimum light intensity: by setting \min to 0, sections of the cloud can become practically black as many intersections are computed. To rather get a more realistic grey as the darkest value, it is better to indicate a minimum intensity greater than 0. In this way, ambient light can be easily simulated.

k = the density of the cloud: The greater it is, the faster the light intensity drops.

The actual mathematical density function is somewhat different. This formula is perception-based and obtains better results. The cloud colour is calculated from the basic colour of the cloud, weighted with the intensity:

$$(R,G,B) \rightarrow (IR, IG, IB)$$

The results of this technique are presented on fig. 8 and 9.

4. RESULTS AND FURTHER WORK

The presented algorithms are prototypes and must be further developed. The main advantage is the change of the shape of the original isosurface geometry towards a realistic looking cloud. The current results show the potential of the two methods. The resulting meshes can be rendered at interactive rates (however depending on the complexity of the input isosurfaces), but the algorithm complexities themselves can still be improved. The fractal algorithm's space complexity is not quite acceptable (see fig. 4 and 9), but the generated cloud mesh can be reduced by a polygon reduction tool. However, a better solution would be to alter the algorithm 1 to produce meshes with an observer-dependent resolution. The current implementation of the lighting algorithm is actually not efficient enough, but can be significantly improved by using standard light simulation methods.

5. SUMMARY

This paper presented two algorithms for generating realistic cloud geometries from isosurfaces produced by a Marching Cube algorithm. Additionally, a simple cloud lighting method was presented to produce better visual results. The achieved rendering performance is sufficient for the generation of weather presentation films.

6. ACKNOWLEDGMENTS

I thank Deutscher Wetterdienst DWD for the meteorological data.

REFERENCES

- [Ebert98] Ebert, D., Musgrave, F.K., Peachey, D., Perlin, K., Worley, S.: Texturing & modelling: a procedural approach. *AP Professional, Academic Press*, Chestnut Hill, MA, USA, 1998.
- [Gardner85] Gardner, G.Y., Visual Simulation of Clouds, *Proceedings of SIGGRAPH 1985*.
- [Gamito95] Gamito, M.N., Lopes, P.F., Gomes, M.R.: Two-Dimensional Simulation of Gaseous Phenomena Using Vortex Particles. In *Eurographics Proceedings 1995*.
- [Haase99] Haase, H., Bock, M., Hergenröther, E., Knöpfle, C., Koppert, H.-J., Schröder, F., Trembilski, A., Weidenhausen, J., Where Weather Meets the Eye - A Case Study on a Wide Range of Meteorological Visualisations for Diverse Audiences. *Proceedings of IEEE/EG VisSym'99* (Springer-Verlag), Vienna, Austria, May 1999
- [Knöpfle97] Knöpfle, Ch., 3D Weather Forecast in a Virtual Studio. *FhG IGD, internal report*, August 1997.
- [Koppert98] Koppert, H.-J., Schröder, F., Hergenröther, E., Lux, M., Trembilski, A., 3D Visualisation in daily operation at the DWD. *Proceedings of the Sixth ECMWF Workshop on Meteorological Operational Systems*, Reading, England, 1998.
- [McCormick87] McCormick, B.H., De Fanti, T.A., Brown, M.D.: Visualisation in Scientific Computing, in *ACM Computer Graphics*, Vol. 21,6, ACM SIGGRAPH, November 1987.
- [Sakas93] Sakas, G., *Fraktale Wolken, virtuelle Flammen. Computer-Emulation und Visualisierung turbulenter Gasbewegung*. Springer Verlag, Berlin, Germany, 1993.
- [Schröder97] Schröder, F., *Visualisierung meteorologischer Daten*, Springer, Berlin, 1997.
- [Stam93] Stam, J., Fiume, E., Turbulent Wind Fields for Gaseous Phenomena. *Proceedings of SIGGRAPH 1993*.

[Unbescheiden98] Unbescheiden, M., Trembilski, A., Cloud Simulation in Virtual Environments, *Proceeding of VRAIS 1998*, Atlanta, USA.

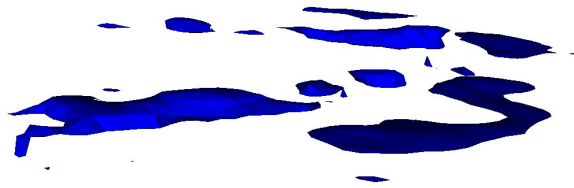


Fig. 3: Original isosurfaces, about 5000 triangles

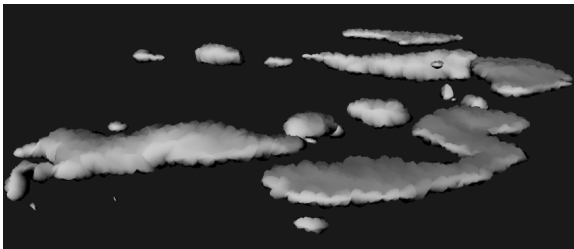


Fig. 4: Refined isosurfaces (method 1), about 80000 triangles

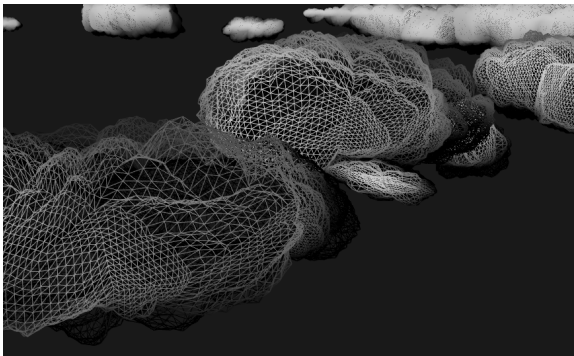


Fig. 5: A closer look at the wireframe of a refined isosurface (method 1)

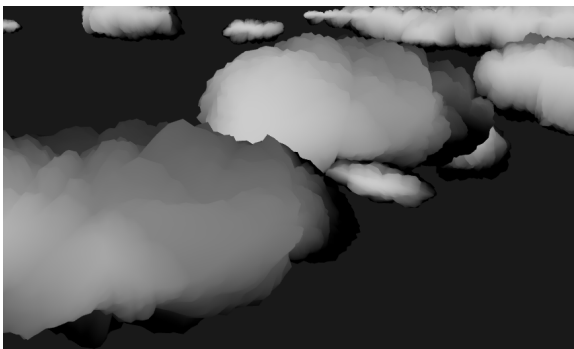


Fig. 6: A closer look at a refined isosurface (method 1)

[WMO87] World Meteorological Organization, *International Cloud Atlas*, Vol I+II, 1987.

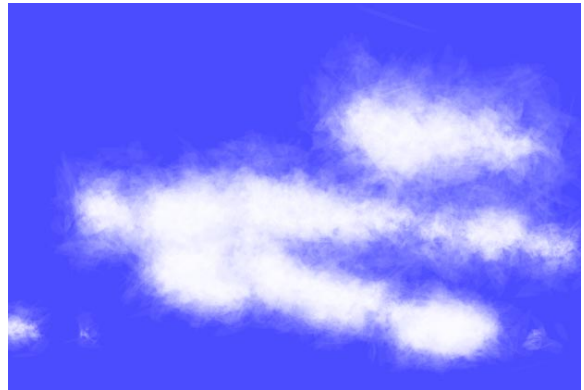


Fig 7: A scattered isosurface with transparency but without lighting, about 15000 triangles

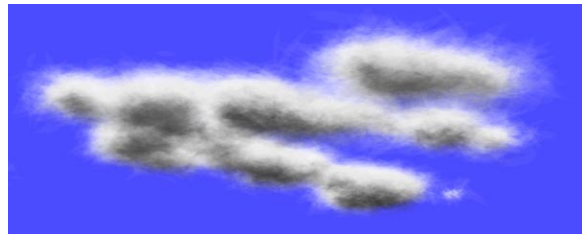


Fig. 8: A scattered isosurface with transparency and lighting

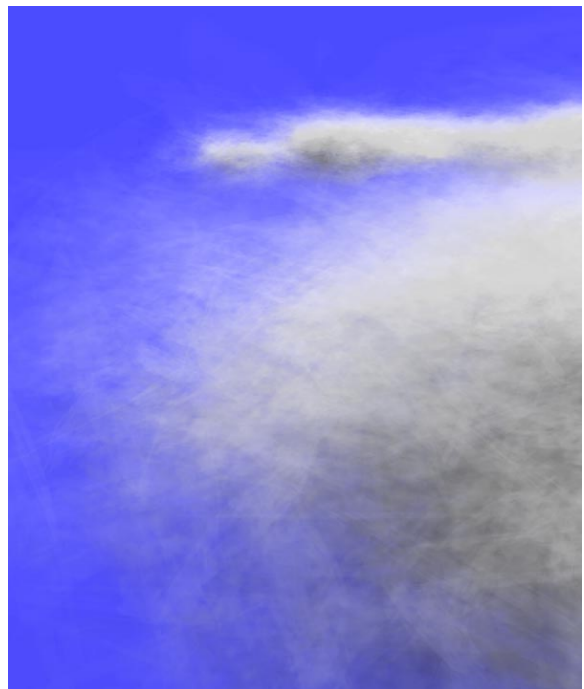


Fig. 9: A scattered cloud geometry (method 1 + method 2) with transparency and lighting, about 120000 triangles.