

MODELING AND VISUALIZATION OF CLOUDS FROM REAL WORLD DATA

A. Roditakis

ETHZ, Federal Institute of Technology Zurich, CH 8093 Hoenggerberg Zurich, Switzerland
(roditak@geod.baug.ethz.ch)
Commission V, WG V/6

KEY WORDS: Photogrammetry, Meteorology, Modeling, Visualization, Graphics, Hardware, Rendering, Point Cloud.

ABSTRACT:

Visualization of gaseous phenomena, such as smoke, fire or clouds has been a challenging issue in computer graphics due to the complicated geometry of such natural objects. In the early 80's, approaches that faced this problem tried to simplify the representation of such geometry, using simple particles or ellipsoids. During the last decade work was done on rendering fractal volumes and random textures in order to create a realistic turbulent image of random generated volumes, and in parallel, modeling of soft or 'blobby' objects (also known as 'metaballs') was combined with hardware accelerated volume rendering and gave some first convincing images of 3D gas volumes in near realtime. There has been large technological developments in the computer graphics hardware and a lot of work has been done from the computer graphics community on simulation of clouds, still there is not much done in the direction of modeling and rendering of such objects from real world measurements, and work is lacking on the issues that arise from datasets with incomplete spatial or temporal resolution.

In the presented work we use the modeling technique of metaballs based on cloud bottom and top height measurements taken during the CLOUDMAP2 project, in order to construct a 3D volume from point clouds. We present the problems that arise from fitting the volume to the measured points, combination of cloud top and bottom height estimations with different resolutions and the generation of large volume data. Solutions to these problems are presented which include the calculation of cloud volumes, interpolation with cloud top height estimations from simultaneous ground based and satellite observations and finally we present various methods to render such volumes using hardware and software assisted techniques.

1. INTRODUCTION

1.1 Motivation and aim of work

The work presented in this paper, was conducted in the frame of a EU project called CLOUDMAP2. It targeted on the parameterization of macro- and micro-physical properties of clouds through the combination of space-borne and ground based sensors. This fusion of sensors helps to calibrate and validate satellite products that could provide global coverage for the cloud location and distribution, and thus help us understand better the role of clouds in the interaction between Earth's climate and anthropogenic inputs. The final aim of our work was to create a processing chain that would take the cloud top and bottom heights estimations as input, and deliver a three dimensional visualization of cloud fields, that would provide us with an instant image of the cloud shape and distribution over measured areas. Furthermore we want to underline the difficulties that come from using real world data for cloud modelling and rendering, where we witness blunders, errors and areas with missing values. These problems do not appear in case studies for cloud rendering, since we notice that for the demonstration of new modelling and rendering techniques, authors use artificially generated data with uniform resolution and coverage (*perfect data*).

1.2 Overview

Within the EU project Cloudmap2, several measurement campaigns were conducted in the area of Switzerland, where simultaneous measurements from ground digital cameras and satellite sensors were conducted, in order to estimate cloud bottom and top heights and winds. From the bottom side, a configuration of digital cameras provides stereo cloud bottom heights photogrammetrically, while at the same time a satellite passes over the area acquiring images, which provide us with stereo top heights. As the focus of the present work lies on the

modelling and visualization of these measurements into clouds, details on the methods used in the acquisition step can be found in the published work of other participants of the CLOUDMAP2 project (Seiz, 2002), (Mayer et. al, 2003).

Starting from the cloud bottom heights we cleaned the dataset from points that did not belong to clouds, and as a next step we experimented with modelling the cloud bottom height, using polygonal modelling and texturing to simulate the appearance of clouds. Similar work had been done by (Lee, 1996) and we also concluded that this method presents limited options for 3D cloud fields. We continued on the path of previous work done on visualization of fluids and applied some basic volume rendering techniques. In the modeling aspect we tried two different options for a definition of a 3D cloud field, the first being a straightforward 3D regular grid calculation, and the second influenced by the work from Nishita and others (Nishita, 1996) on soft body modelling using 'meta-balls', also known as 'soft-balls'. These two methods interpolated the distinct points over a 3D grid, with the second having the advantage of smooth blending of neighbouring points.

Rendering of this three-dimensional cloud field was also performed by utilizing two different techniques in parallel. We tested the well known method for visualizing three-dimensional iso-surfaces called 'marching cubes', which provided us with rendering of a surface of the cloud field, with a chosen iso-parametric value, and in parallel we implemented a hardware accelerated volume rendering technique, using OpenGL language and textured two-dimensional slices of the volume. Finally we made use of an existing software library, used at the National Center for Atmospheric Research, at Colorado, USA (Lacroute, 1994), (NCAR URL) that although doesn't deliver real-time rendering of the volume field, its high performance in combination with the realistic lighting calculations, gave until now the best results as far as visual quality is concerned. In order to demonstrate the difference of an extrapolated 3D cloud

field, from a cloud field that is present in all three dimensions, we used the same processing chain both in the cloud bottom field, and the local alpine numerical weather prediction model, used from the Swiss Meteorological Office (MeteoSwiss).

1.3 Related work

The problem of modelling and rendering gaseous phenomena has been an active topic since 20 years now either as a problem not solved in computer graphics at that time (Blinn, 1982), or in more practical approaches in the field of cinema special effects (Reeves, 1983). Point based rendering techniques have given a simple, yet flexible and powerful tool for rendering. Levoy and Whitted extended the use of points further than rendering smoke or fire, into traditional geometry visualisation (Levoy, 1985). The last few years were followed by an increase in applications demanding visualisation solutions, together with huge developments in computer hardware. In (Rusinkiewicz, 2000), (Zwicker, 2001) points in combination with texture splatting were used, while Harris (Harris, 2001) used particles and impostors to deliver high frame rates for scenes with many cloud objects. Nishita and others (Nishita, 1996), (Dobashi, 2000) used a particle system to control the metaballs model that composed the cloud objects and in combination with hardware accelerated OpenGL programming, they achieved impressive results of rendering and animation of clouds in near-real-time (Figure 1).



Figure 1: Clouds modelled using metaballs (from Dobash et. al. 2000)

Alternative methods like the planar and curved surfaces with texture used by Gardner (Gardner, 1985) and textures with noise and turbulence functions (Ebert, 1990), taken from the work of Perlin (Perlin, 1985) provided new perspectives, which were difficult though to apply to real world measurements. Volume rendering has undergone many improvements on the speed of algorithms and the work of Lacroute at the University of Stanford (Lacroute, 1994), (Volpack URL) and its development by Schulze (Schulze, 2001) attracted our attention and part of it was included in our implementation.

2. MODELING AND RENDERING OF CLOUDS

2.1 Initial tests

We started our tests on the ground-based (GB) measurements, creating a triangulation of the cloud bottom height (CBH) surface. Afterwards we used the GB images in combination with the point cloud for determining whether the point belonged to a cloud or not (Figure 3). We first performed an adaptive histogram equalization on the image and then classified the points. The results of this procedure were quite good (Figure 3), but is not full-automatic since it demands some input from the user.

The remaining points formed the triangulated model of the CBH surface, which was textured mapped, using the standard interface OpenGL (OpenGL URL) (Woo, 1999), as an attempt

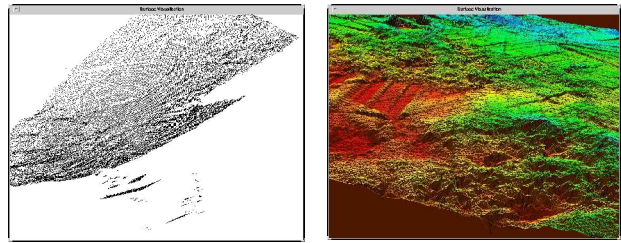


Figure 2: Point cloud from ground measurements (left) and its triangulation (right)



Figure 3: Cloud mask applied in GB measurements

to increase the realism of the visualisation. These first results were fair, but further improvement was not intended since we consider traditional polygonal modelling/rendering as an inefficient method for volumetric phenomena. A part of these first attempts were used, though, in further stages.

2.2 Development of modelling methods

The point cloud which resulted from the first stage, was used as data for developing and testing a point based rendering system, using particles. We constructed a simple system which includes positional and colour information for each particle, based on the point measurements of CBH. We also prepared the source code to import wind direction vectors for the animation of 3D clouds from subsequent measurements, lifetime duration of each particle and normal vector direction. This normal vector direction results from the triangulation of the first stage. Some test animations were performed with GB measurements, adding artificial data for wind direction and speed, with variations in the particle size and anti-aliasing methods with satisfying results as far as performance and memory consumption is concerned.

In the frame of this work, we first looked into different techniques of cloud modelling and concluded that the main issue is the description of clouds as volumes. These volumes consist of cells, where the values of the desired attributes are stored, usually aligned in regular grids. We constructed three-dimensional textures from ground based measurements. The pixel values adjust the material transparency and they are calculated according to the number of point measurements present inside each volume pixel. Volume modelling was tested, using the help of a software used in several weather-related applications, called Vis5d. An open source software, released under the GNU General Public License (Vis5D URL). The positive conclusions are the compression of the volume data, and the ease of including subsequent measurements and creating an animation. The negative conclusions are the absence at that stage of CTH estimations, which lead to an incomplete volume dataset, and a not satisfying final volume rendering. The first impression was that a completed version of the dataset would bring much better results.

The classification method described in section 2.1 was based only upon the radiometric behaviour (colour channels) of the

points measured. A successful segmentation into cloud objects would require to take into consideration the spatial relations between the measured points. An existing method that attracted attention was the modelling of potential 3D fields and the display of iso-potential values, using implicit density functions.

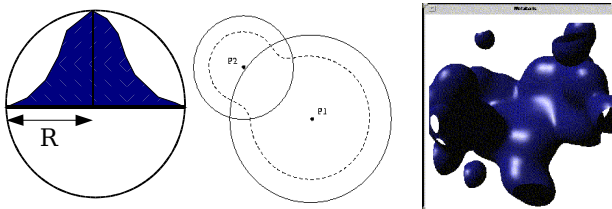


Figure 4: Metaball of radius R (left), blending of metaballs in 2D (middle) and 3D (right)

These set of functions evaluate the value of the field using the distance from specified 'source' points. A sphere around each point defines the area of influence, out of which the density is 0 (Figure 4). This technique was first applied in modelling free form surfaces, which are defined as the iso-potential surface of a value V. The objects modelled in this manner are called 'metaballs', or 'softballs' or 'blobs' (Nishita, 1996). Their

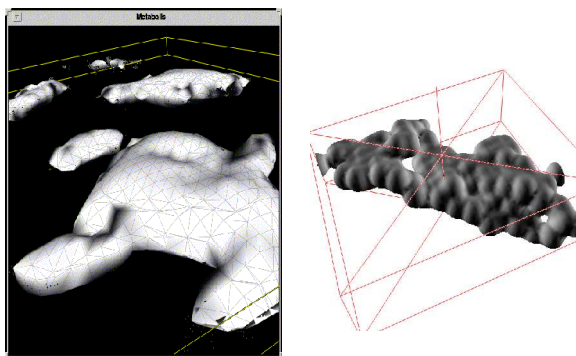


Figure 5: Application of metaballs on GB measurements with different metaball radii

application was extended into modelling clouds by (Dobashi, 2000), in combination with hardware accelerated rendering with realistic results.

Our first effort in this direction was to construct the functions that calculate the field density, based on the distance from the 'source' points. In our case the source points were the CBH point measurements themselves and since the points were organized on a regular grid, we used the resolution of this grid, as the fall-out radius for the metaballs. The visualisation of the isosurface at this stage has been implemented using polygons. The extraction of the iso-surface into a polygonal model was done using the well known 'marching cubes' algorithm. The algorithm can be divided into three sub-routines. First the object space is divided into cubes, at a resolution that we decide and the user supplies the iso-value that desires to extract. Next the algorithm for every cube evaluates the field on every cube vertex and calculates the difference with the desired iso-value. According to the sign of the difference on the cube-vertices, the algorithm finds in a pre-calculated lookup table, in which case falls the intersection of the iso-surface with the cube edges. This speeds up the decision which cube-edges have to be interpolated, in order to create the triangle that represents the iso-surface.

The results depend on the resolution of the cube grid, and in our case this plays a significant role in calculation and rendering times. For example in GB measurements, depending on the average bottom height the coverage varies from 1 to 4 Km in each horizontal direction in a resolution of approximately 5m,

depending again on the cloud average height. The coverage of the scene from the point cloud is irrelevant, since this algorithm covers the whole object space, disregarding holes in the object. The time consumption is largely due to calculation of the intersections and for a scene, with dimensions as described above, 2~3 minutes are required to compute the triangular surface.

We applied the forementioned 'metaballs' technique in order to calculate the pixel values for a cloud scene. (Mayer, 2003), (Nishita, 1996). Around each of the point measurements a sphere of influence is placed, with the amount of influence decreasing as the distance from the centre increases according to a density function (Figure 4). This allows us to estimate a density value on a regular grid, which comes from all 'metaballs' that include the current grid vertex, weighted by their influence. The result of applying this method on a set of ground-based measurements is shown in Figure 5, where the outer shell of the estimated cloud volume is shown in the form of polygonal surface.

Due to the large computational times, for a whole scene taken from EO measurements we started the implementation of hierarchical grid structure, for increasing the storage and speed performance. We noticed that this issue has been addressed in the past by the computer graphics community and we chose a software library used at the National Centre for Atmospheric Research, at Boulder Colorado, for similar purposes as ours (NCAR URL). It is based on the "shear-warp" algorithm for fast volume rendering (Volpack URL), (Lacroute, 1994), and facilitates a complete workflow from reading raw volumetric data, storing colour, transparency and normal vector information in compressed files, to rendering the volume data.

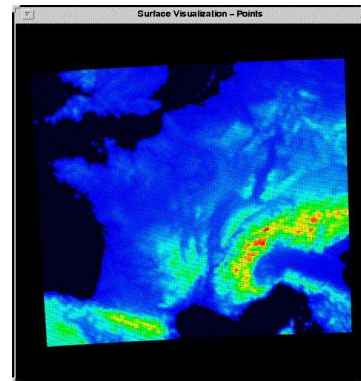


Figure 6: The aLMo coverage

The compression schemes (*rle* and *octree*) allow the storage of the volume data values, together with the transfer functions which determine the colour and opacity of the medium. The *rle* (Run Length Encoded) scheme is optimal for viewing the same volume under different viewing transformations and shading parameters, while the *octree* scheme performs better with volumes with varying transfer function.

The pre-processed volumes have larger file size (up to four times) in comparison with the raw binary data, and this happens due to the material colour/transparency and normal vector information that have to be stored within. Keeping all these information together in one volume file, is though a more important advantage since the processing time is significantly reduced. The software is based on the Tcl scripting language and can be combined with the existing C source easily. The sequence of commands needed to be passed to the Tcl interpreter are saved in script files, and can be called within any C program.

We continued by importing the alpine local model (*aLMo*) (source: Meteo Swiss) into the volume rendering procedure.

Grids from numerical models describe the Cloud Liquid Content in %, in 7×7 Km² cells, covering an area of 1401×1197 Km² (Figure 6). The vertical size is not uniform, and this was an obstacle in importing the grid into a volume suitable for volume rendering. The aLMo NWP data-set is used as test for 3D cloud field modeling and interpolation of irregular grid to regular was performed in IDL programming environment due to the ease of handling 3D arrays of data.

At the current stage we started with importing 3D cloud fields from combined ground-based and satellite observations into the Volsh volume rendering procedure. The data originate from the GB and EO measurements that took place in April 2002, over the Kloten airport near Zurich. The area covered from the EO measurements is 55×55 Km² and approximately 2×2 Km² from the GB observations. The 3D cloud field is described by the cloud top height (CTH), the cloud liquid water (DLR product) estimated on the whole of the grid from the MISR satellite data and the CBH information extrapolated over the area covered from the GB measurements.

2.3 Development of rendering methods

In parallel with the development of 3D cloud fields we used several techniques for visualization of 3D cloud volumes. After the creation of these volumes, based on the metaball technique (see Section 2.2), we implemented a hardware assisted rendering method (Woo, 1999) which is used often in volume rendering. In its simplest form we create planar cross-sections of the volume, oriented towards the viewer, which are in the end blended together. The resulting image is the combination of

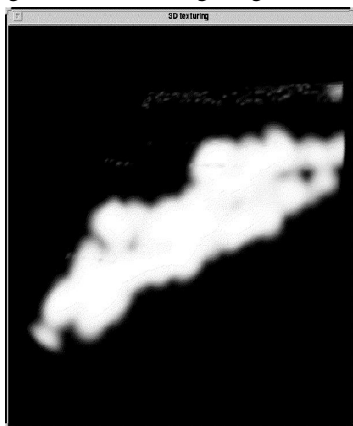


Figure 7: Volume Rendering with planar slices

the colour from all planes, multiplied by their transparency. Using the example of the previous section we supply a sample rendering in Figure 7, where we can see that the absence of lighting calculations (shadows) in the volume results in difficulty understanding the depth of a volume. This problem can be solved as in (Dobashi, 2000), where lighting calculations are performed by sorting each of the planes towards the light source, and calculating an attenuation factor for each plane. This factor represents the amount of light that reaches a plane after passing through the other planes that lie in front of the light source, and determine its shade level.

We further improved the interactivity our rendering method by introducing dynamic levels of detail for the 3-dimensional volume. The levels are created by the graphics processor from the original resolution data, and the OpenGL graphics library handles the display and transition between the levels. In order to avoid aliasing effects from undersampling the original

volume and the LoDs' transition we tried trilinear interpolation, which improved the image quality of the texture, but slowed down the frame rate significantly. In parallel we had implemented the import procedure into the Volsh library (see Section 2.1) and tried its rendering capabilities (Figure 8). The procedure is no more real-time, but the rendering times are quite low. For example a volume of $256 \times 128 \times 64$ pixels is rendered in approximately 3 seconds.

To the advantages of the last rendering approach we count the option to add shading to the volume, which increases the realism of the resulting image, and the option to adjust the transfer function, which defines the material properties of the medium for every value of the scalar variable, in near real-time. This means that we can interactively change the material properties, and focus on some specific range of the scalar value, leaving the outliers transparent. For the animation, we have the option to record some key-positions and create the intermediate

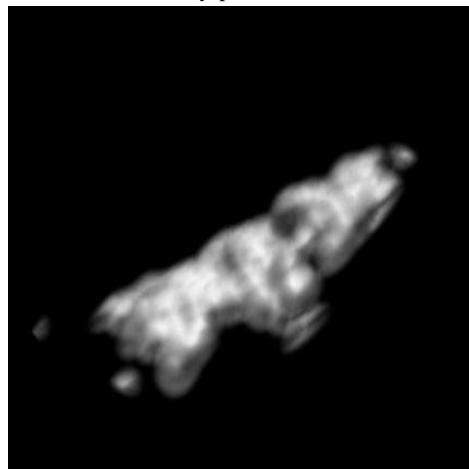


Figure 8: Volume rendering using 'Volsh' library

frames, by interpolating between the neighbour key-positions, thus limiting the number of commands necessary to render the frame sequence. Finally the library is based on the Tcl scripting language, which allows the execution of any external program in combination with the rendering procedure. This allows us, for example, to automate the production of animations, and deliver video from the separate rendered frames.

For the disadvantages we can mention the absence of perspective projection in the library (only parallel projection is available), and the complexity to manipulate the camera flightpath to create key-frames for the animation of volumes.

A rendering of the whole scene (Figure 9) revealed problems with poor rendering quality, which result from the fact that the input data are the surface points measured from the EO observations and do not correspond to a 3D volume, but a surface.

In order to evaluate the performance of the procedure with 3D volumes we imported the CLC - cloud cover- variable from the alpine local model, used at MeteoSwiss (see Section 1), and created an animation of a fly-through around the rendered volume. A snapshot from this animation is given in Figure 10. We continued by visualising the interpolated 3D Cloud Liquid Water, from the combined EO and GB measurements, taken in April 2002. These measurements are a fusion of the MISR Cloud Top Height estimations over Zurich-Kloten and the ground-based CBH estimations, taken at the same time, using the ground camera system developed at ETH Zurich. The measurements cover a different extent of the area and the CBH were extrapolated over the whole extent.

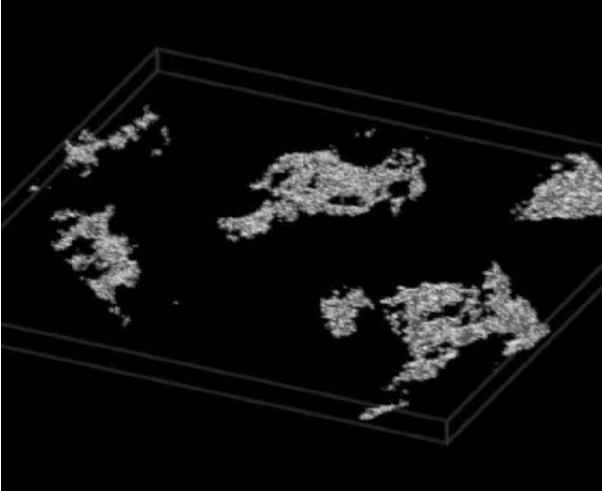


Figure 9: Volsh rendering of a whole GB scene

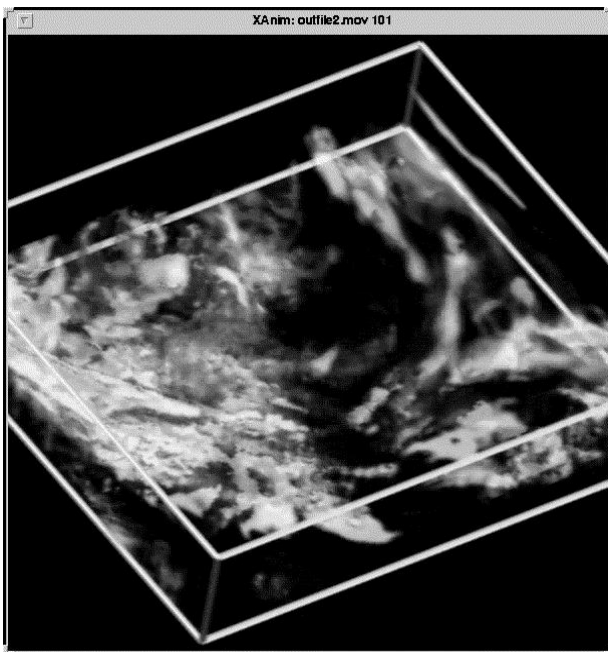


Figure 10: CLC cloud cover from aLMo NWP model

3. CONCLUSIONS AND FURTHER WORK

From the short literature study that we did, we have seen that the problem of modelling and rendering cloud-type data can be approached from different routes. There is a trade-off between the simulation accuracy of the physical phenomenon and the time required to deliver visualisation. One can focus on keeping a realistic image of cloud formation and movement together with interactive rendering speed, but in scientific visualization this would mean drifting away from the results of the measurements. On the other hand, by using strictly the results of measurements, we move to the other end of visualization difficult to understand without detailed knowledge of the instruments used and possible errors that may happen. In the present work, we demonstrated the several problems and difficulties that arise from using real measurements in cloud modelling and rendering. If visualization is about delivery of desired information in a quick and comprehensive way, then the field of 3D volume rendering has a lot of open issues in fields such as environmental sciences and meteorology.

The present combination of particle systems, soft body modelling, hardware accelerated graphics and software-based volume rendering had the purpose to teach us the benefits that scientific visualization can gain from following and adjusting current advances in computer graphics.

In the current status of the presented work we would like to improve the search algorithm that adds up the metaball weights for every grid cell, since at the moment is a straightforward loop that traverses all the metaballs, without taking into account if they will contribute finally or not. We are interested in continuing the work on temporal interpolation from several time steps in order to deliver animations over time. This will be possible to do on the aLMo NWP model since it provides 3D values every hour. Application of this idea on the combined CBH and CTH measurements seems difficult because of the temporal resolution of the satellite images.

4. ACKNOWLEDGEMENTS

The aLMo NWP data were supplied by Gabriela Seiz and MeteoSwiss, Zurich Weather Office. We would like to thank our CLOUDMAP2 colleague Tobias Zinner at DLR, for supplying us with three-dimensional cloud liquid water from the combined CLOUDMAP2 measurements in Zurich Kloten airport, taken on 12 April 2002. This work was funded by the Bundesamt für Bildung und Wissenschaft (BBW), within the EU project CLOUDMAP2 (BBW Nr. 00.0355-1).

5. REFERENCES

- Blinn, J. Light reflection functions for simulation of clouds and dusty surfaces, *Proceedings of the 9th annual conference on Computer Graphics and interactive techniques*, July 1982, Boston, USA, pp. 21-29.
- Dobashi, Y., Tomoyuki, N., Yamashita, H., Okita, T. Modeling of clouds from satellite images using Metaballs, *The Visual Computer*, 1999, 15(9), pp. 471-482.
- Dobashi, Y., Kaneda, K., Yamashita, H., Okita, T., Nishita, T. A simple, efficient method for realistic animation of clouds, *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, July 2000, pp. 19-28.
- Ebert, S., D., Parent, E., R. Rendering and animation of gaseous phenomena by combining fast volume and scanline A-buffer techniques. *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, August 1990, Volume 24, Issue 4, pp. 357-363.
- Ebert, S., D., Musgrave, F., K., Darwyn, P., Perlin, K., Worley, S., 1998. *Texturing and Modeling – a procedural approach*, second edition, Academic Press.
- Gardner, Y., G., Visual simulation of clouds, *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, July 1985, pp 297-304.
- Harris, J., M., Lastra, A. Real time cloud rendering, *Eurographics Computer Graphics Forum*, 2001, 20(3), Blackwell Publishing, pp. 76-84.
- Lacroute, P., Levoy, M. Fast Volume Rendering Using a Shear-Warp Factorization of the Vieweing Transformation, *SIGGRAPH 94, Proceedings of the 21st annual conference on computer graphics and interactive techniques*, July 1994, Orlando, USA, pp. 451-458.

Lee, C., Kesselman, C., Schwab, S. Near-real-time Satellite Image Processing: Metacomputing in CC++ . *IEEE Computer Graphics and Applications*, July 1996, pp. 79-84.

Levoy M., Whitted T. The use of points as a display primitive, *Technical Report 85-022, Computer Science Department, University of North Carolina at Chapel Hill*, January 1985.

Mayer B., Zinner, T., Seiz, G., Roditakis, A. *CLOUDMAP2 Second Interim Report*. February 2003.++

NCAR URL, *Volsh Volume Rendering Software*, <http://www.vets.ucar.edu/software/volsh/> (accessed 4 Feb. 2004)

Nishta T., Dobashi Y., Nakamae E. Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, July 1996, pp.379-386.

OpenGL URL, *The industry standard for high performance graphics*, <http://www.opengl.org> (accessed 4 Feb. 2004)

Perlin K. An image synthesizer, *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, July 1985, 19(3), pp 287-296.

Perlin K., Hoffert M. E. Hypertexture, *Proceedings of the 16th annual on Computer graphics and interactive techniques*, July 1989, 23(3), pp. 253-262.

Reeves, W. Particle Systems – A technique for modeling a class of fuzzy objects, *ACM Transactions on graphics*, 2(2), April 1983, pp. 91-108.

Rusinkiewicz, S., Levoy, M. Qsplat – A multiresolution point rendering system for large meshes, *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, July 2000.

Schulze, P., J., Niemeier, R., Lang, U. The Perspective Shear-Warp algorithm in a virtual environment, *IEEE Visualization 2001*, October 21-26 2001, San Diego, California, USA, pp. 207-213.

Seiz, G., Baltsavias, E., Gruen, A. Cloud mapping from the Ground: Use of Photogrammetric Methods. *Photogrammetric Engineering & Remote Sensing*, 68(9), September 2002, pp. 941-951.

Vis5D URL, *The Vis5d home page*, <http://www.ssec.wisc.edu/~billh/vis5d.html>. (accessed 4 Feb. 2004)

Volpack URL, *Computer Graphics at Stanford University*, <http://www.graphics.stanford.edu>. (accessed 4 Feb. 2004)

Woo, M., Neider, J., Davis, T., Shreiner, D. *OpenGL Programming Guide*, October 1999, third edition, Addison Wesley.

Zwicker M., Pfister H., van Baar J., Gross M. Surface Splatting, *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, Los Angeles, California, USA, August 12-17 2000, pp. 371-378.