

Schafer

***Support for
NAWC-China Lake
SSGM Cloud
Generation Utility***

February 1999

Prepared by:

***Schafer Corporation
1901 N. Ft. Myer Drive
Suite 800
Arlington, VA 22209
703/558-7900***

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

19990409 082

**Task Report – Naval Research Laboratory
Contract N00014-97-D-2014/001**

Support for NAWC-China Lake SSGM Cloud Generation Utility

Introduction

The objective of this task is to provide the UAV-BPI IRST development effort with a utility for producing cloud scenes for use in the Synthetic Scene Generation Model (SSGM). As currently configured, SSGM comes with only a handful of cloud scenes, which are not necessarily adequate to fully represent the range of cloud conditions in the theaters of interest. The cloud generation utility produced under this effort utilizes historical cloud coverage data for the regions of interest to produce cloud scenes representative of the location and time of year.

Description

The cloud generation utility produced under this effort is actually a series of utilities which, when applied in sequence, produce the desired SSGM input file. This approach was chosen because SSGM runs on UNIX-based Silicon Graphics workstations. The concept of sequentially running a series of "filters", often via "shell scripts," is a long-standing and very useful paradigm on UNIX-based operating systems. This approach allows each step of the process to be developed and tested independently, thus improving quality control.

The first in the series of utilities is a cloud data extractor. This program permits the user to select cloud information from the cloud databases used (RTNEPH [Reference 1] and HIRS [Reference 2]), based upon certain criteria. These criteria, specified in an input file, fall into three sets. The first is the geographic area over which the cloud field is to be generated. The second is the time period of interest, which permits either calculation (essentially just a database lookup) for a specific date and time, or statistical examination of a season, month, year, etc. Last is the desired statistical condition, namely worst, best, median, nth quintile or decile. Based upon these criteria, a particular entry (year, month, day, hour) in the cloud database is selected as representative.

The RTNEPH and HIRS data corresponding to this entry are then combined in the following manner. Owing to the differing sensitivities of the instruments used to gather the data in these two data sets, the data sets will inevitably differ. The most conservative assumption is that the data will contain few, if any, false positives (cloud detection where no cloud exists). More usual would be the situation where an instrument fails to detect cloud that is actually present. Given this assumption, the proper means of combining the data is to simply accept the value that indicates the greater amount of clouds at a particular location and time. The process is further complicated by the different geographic grid sizes of the two data sets. This is handled by interpolating the sparser data set to the grid of the other. Additionally, the HIRS data set is used solely above the maximum altitude of the RTNEPH data set. Upon completion of the data extraction, the data are formatted into a three dimensional array of percentage cloud content for input into the next utility.

The second utility is the cloud generator that is the key component of this effort. This utility accepts the "percent cloudy" array described above and produces a realistic cloud surface. A description of the algorithm used follows.

The code uses the RSA (Rescale and Add) fractal algorithm. The key equation for this is found in Reference 3, Eqn. (2.2-1), which samples a random input lattice, mapping it to the output lattice,

$$V_n(\bar{x}) = \sum_{k=k_0}^{k_1} \left(\frac{1}{r^{kH}} \right) S_n \left[f \left(r^k \bar{x}' \right) \right] \quad (2.2-1).$$

Here, the summation limits are chosen to represent all the spatial scales of the output field. The parameters r and H control the fractal nature of the output. The index n is the number of topological dimensions (four here). Lastly, S_n is a smooth interpolation from a random lattice to the vector position given in the function's argument.

Basically, a random number between -0.5 and 0.5 (this random number will eventually be converted to a random number between 0 and 1 by adding 0.5 to it) is generated for each point on a 600x600 lattice, the LatticeWater[I][J] array. Eqn. (2.2-1) is then used: for each value of k , the value of Water for each physical point in space (each physical point in space is an element of the Water[i][j] array) is found by simply interpolating between the LatticeWater[I][J] points. The summation index starts with $k = 0$ and goes to k_{max} . The $k_{max} + 1$

terms are added together for each physical point in space to achieve the final Water[i][j] value for each physical point in space.

The primary data sets are gridded in three dimensions with the horizontal dimensions of the grid much larger than the vertical dimension, leaving each data point representing a spatial volume reminiscent of a pizza box. The array, Water[i][j], is calculated for a 2-dimensional set of spatial points corresponding to one of these "pizza boxes." A threshold is then calculated, based on the fraction filled quantity (the fraction of the pizza box filled with cloud) read from the database by the cloud data extractor. All the Water[i][j] values that are below the threshold are set to zero. The threshold value is calculated by the method of steepest descents.

The implementation here differs in several aspects from Reference 3. One difference is that the interpolation method of Reference 3, Eqns.(2.2-3,4) is not used. A standard linear weighting function is used instead,

$$h_i = r^k x_i' - \text{INT}(r^k x_i')$$

where x_i is the discretized version of the \bar{x} vector from Eqn (2.2-1). The more complicated expressions of Reference 3 did not prove to be necessary.

The second difference is that a 600x600 LatticeWater[I][J] lattice is used, rather than the 40x40 lattice suggested by Reference 3. Using a lattice that is much larger than the physical area of interest is necessary in order for the resulting cloud formations to be continuous. The 600x600 LatticeWater[I][J] lattice was found to work well for the 40 km by 40 km pizza box characteristic of the RTNEPH database at mid-latitudes.

The basic algorithm may be explained as follows. The process starts with a spatially large copy of LatticeWater[I][J]. A smaller copy of LatticeWater[I][J] times a fraction (which makes the second term a perturbation on the first term) is added to the previous copy, with the process repeating. It may be seen in Reference 3, Figure 4 that for the k=2 term, LatticeWater[I][J] is half the size of the pizza box (note the repetition). The LatticeWater[I][J] lattice for the k = 0 and k = 1 terms is larger than the pizza box. For the k=3 and k=4 terms, the LatticeWater[I][J] lattice is much smaller than the pizza box. It has been found that it is necessary for LatticeWater[I][J] lattice to at least be larger than the pizza box for the k=1 term, and it probably helps for this to be true for the k=2 and larger terms as

well. Otherwise, there may be discontinuities in the generated cloud. This is why a 600x600 LatticeWater[I][J] lattice is used, rather than the 40x40 lattice suggested in Reference 3.

A third departure from References 3 and 4 is that a uniform random number generator is used to generate the random LatticeWater[I][J] values. References 3 and 4 use a Gaussian random number generator, but then later convert the Gaussian distribution to a uniform distribution. It was decided to use a uniform distribution from the start because it is much easier (most computer languages have a built-in function for generating uniform random numbers) and because the Gaussian distribution seems unnecessary since it is later converted to a uniform distribution (Reference 4, page 14).

According to Reference 4, Eqn. (2-3), Water[i][j] is proportional to the cloud height. One may then use charts such as Reference 4, Figure 7 to calculate moisture as a function of vertical position in the cloud. Reference 4, Figure 7 indicates that moisture is roughly proportional to vertical position until one gets up to one-half of the maximum cloud height, at which point the moisture decreases as vertical position increases in a roughly linear manner. On this basis, one can therefore claim that cloud "thickness" is roughly proportional to Water[i][j].

Finally, the SSGM model, which is primarily oriented toward generation of scenes as viewed from satellites, only utilizes cloud top information. The cloud generation utility therefore only outputs this information. The data are output in the well-known FITS format.

The third step in the process is to convert the cloud top data from FITS to SSGM input format. This is done by means of a utility that comes as part of the full SSGM install. Upon conversion, the file merely needs to be placed into the location where SSGM expects to find cloud data. The shell script therefore concludes by performing all the necessary setup enabling SSGM to properly utilize the new cloud data. It should be noted that, because the intermediate format is FITS, this file should be suitable for manipulation and display by any image analysis program which accept FITS format files.

Summary

The present effort has produced a sequence of utility programs that are able to generate realistic cloud environments for the SSGM model. These utilities, when run together

via the appropriate UNIX shell script, perform the following steps. First, data are extracted from the pertinent cloud environment databases according to user specified criteria. Next, these extracted data are used to generate a cloud scene via a fractal algorithm. Lastly, this cloud scene is converted into a format suitable for input to SSGM. The newly generated file may be treated co-equally with the "canned" cloud scenes provided with SSGM.

References

1. Hamill, T. M., R. P. D'Entremont, and J. T. Bunting, "A Description of the Air Force Real-Time Nephanalysis Model," *Weather and Forecasting*, Vol. 7, Number 2, p.288-306, 1992.
2. Wylie, D. P., W. P. Menzel, H. M. Woolf, and K. I. Strabala, "1994: Four Years of Global Cirrus Cloud Statistics using HIRS," *J. of Clim.*, 7, p.1972-1986.
3. Cianciolo, M. E. and R. G. Rasmussen, *Cloud Scene Simulation Modeling: The Enhanced Model*, PL-TR-92-2106, April 1992.
4. Cianciolo, M. E., M. E. Raffensberger, E. O. Schmidt, and J. R. Stearns, *Atmospheric Scene Simulation Modeling And Visualization*, PL-TR-96-2079, April 15, 1996.