Synthetic IR Scene Simulation of Air-borne Targets

Shankar T. More, Avinash A. Pandit, S. N. Merchant, U. B. Desai*
SPANN Lab, Department of Electrical Engg, IIT Bombay.

Abstract

IR scenes of high fidelity are needed to support the development and testing of various target detection and tracking techniques. It is impractical to test detection and tracking algorithms under all conceivable conditions. Therefore, to test the effectiveness of detection and tracking algorithms under variety of scenarios, synthetic IR scenes are generated. For air-borne targets, the presence of clouds plays an important role, since they affect most IR sensors. We propose, a modification of original Gardner's Method [3], in order to generate clouds of richer spectral content. We also explore an algorithm based on self-similarity [5] for cloud texture generation. Synthetic IR cloud images generated by our scene simulation software are radiometrically accurate and have typical cloud texture variations. We use Modtran 4.0 for radiometric calculation and VRML (Virtual Reality Modeling Language) for scene rendering.

1 Introduction

Generation of high-fidelity IR scenes require a combination of infrared radiance calculation model and computer graphics techniques for image rendering. The process of IR scene simulation of air-borne targets consists of,

- Synthetic IR cloud generation,
- Radiance calculation of aircraft, and
- Scene rendering.

In the IR scene simulation of air-borne targets, clouds introduce spatially varying background radiance, which may be a dominant noise source. Foreground clouds degrade or occlude the target signature while background clouds clutter the scene.

Exact models, which take into account detailed microphysics of clouds, require large computation times, and inputs which are difficult to obtain. The proposed models produce realistic cloud scenes in a reasonable amount of time and resources. The approach that we have adopted is to generate typical cloud texture and then modulate it by the average radiance of cloud, calculated using Modtran 4.0. The Model assumes that variation in cloud temperature and emissivity is small. The process of IR cloud generation can be divided into

- Cloud texture generation,
- Radiance calculation of clouds, and
- Texture to radiance mapping.

The proposed models for cloud texture generation incorporate spatial non-uniformity based on $1/f$ spectral shaping of spatial variation. The effect of atmospheric path from the clouds to the observer is incorporated by Modtran 4.0.

2 Cloud Texture Generation

Cloud synthesis poses problems to the image generation technique, as they do not have well-defined surfaces and boundaries. This step is important from the point of view of the appearance of an IR scene. We generate realistic cloud texture with moderate computational cost. The emphasis is on generation of typical cloud spatial texture rather than a detailed micro-meteorological calculation of a particular cloud situation. We propose two models for cloud texture generation:

1. Modified Gardner's Method (MGM), and
2. Self-Similarity Approach (SSA) .

These methods are explained in the following subsections.

2.1 Modified Gardner's Method (MGM)

The texturing function developed by Gardner [3], represents the cloud spectral content using Fourier series.

*Emails: more@ee.iitb.ac.in, avinashp@ee.iitb.ac.in, merchant@acre.iitb.ac.in,ubdesai@ee.iitb.ac.in

The Gardner's texture model is based on the fact that clouds exhibit $1/f$ spectral shaping. The texture function is defined by the following equation:

$$T(X,Y) = k \sum_{i=1}^{n} [C_i \sin(Fx_i X + Px_i)$$

$$+T_o] \sum_{i=1}^{n} [C_i \sin(Fy_i Y + Py_i) + T_o] \quad (1)$$

There are several parameters, that offer flexibility in specifying different types of clouds. Natural-looking clouds can be produced if the frequencies and the coefficients are chosen according to the following relationships:

$$Fx_{i+1} = 2Fx_i \quad (2)$$

$$Fy_{i+1} = 2Fyi \quad (3)$$

$$C_{i+1} = .707C_i \quad (4)$$

The phase shifts Px_i and Py_i add randomness to the texture,

$$Px_i = \pi/2 \sin(0.5Fy_i Y) + Q_i = \pi/2 \sin(Fy_{i-1} Y) + Q_i \quad (5)$$

$$Py_i = \pi/2 \sin(0.5Fx_i X) + Q_i = \pi/2 \sin(Fx_{i-1} X) + Q_i \quad (6)$$

where, Q_i is a random number in the range $(-\pi, \pi)$.

We propose a modification to the texturing function developed by Gardner. The Modified Gardner's texturing function is defined as:

$$T(X,Y) = k \prod_{j=1}^{m} \sum_{i=1}^{n} [C_i \sin(Fx_{ji} X + Px_i)$$

$$+T_o] \sum_{i=1}^{n} [C_i \sin(Fy_{ji} Y + Py_i) + T_o] \quad (7)$$

The Modified Texturing Function has m Fourier series. The result of the generation of cloud texture, based on the original Gardner's Method (GM) and MGM is presented in Fig.(1). The inclusion of m frequency components in the texture function improves the spectral content of the cloud. From the figure, it is clear that cloud textures generated by MGM are spectrally rich, and look natural.

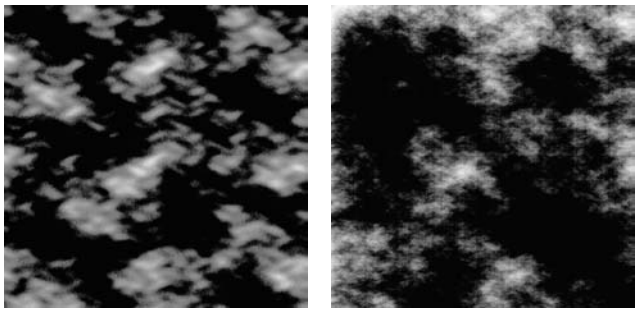


Figure 1: Comparision between images generated by Gardner's method and Modified Gardner's method

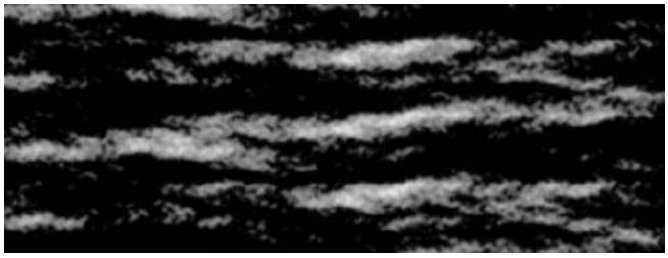


Figure 2: The Cirrus cloud

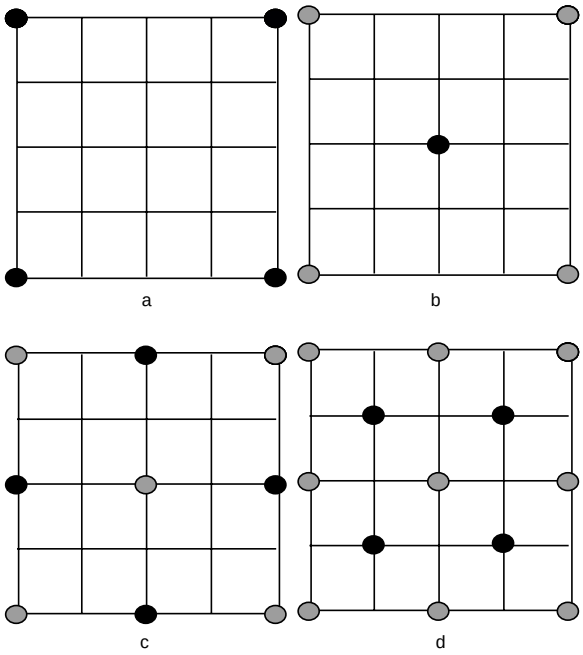
By varying values of the parameters, we can generate clouds of different shapes. For example, we can generate a cirrus cloud characterized by a long wispy tail by specifying Fx_i to be double or greater than Fy_i. This is shown in Fig.(2).

2.2 Self-Similarity Approach (SSA)

Studies have revealed that the clouds have $1/f$ spectral characteristics. A self-similar process can be used for modeling and analysis of $1/f$ phenomena [5]. This property can be used for the generation of cloud images. We have investigated an algorithm based on the self-similar approach. The proposed algorithm is explained below for the 2D case.

We start with an array whose dimension is a power of two plus one, for e.g. 65X65, 129X129. We set four corner points to some value in the range $(-1, 1)$. This is the starting-point for the iterative subdivision routine, which can be divided in two steps:

Step I: Center-point calculation. The value at the midpoint of square is obtained by taking the average of four corner-points, and adding to it a random number, generated using a self-similar process



(a): Four corner points of a square, (b): Center value obtained using Step I, (c): Midpoint values along the edges obtained using Step II, (d): Repeatation of Step I.

Figure 3: Steps explaining SSA

as explained below.

Step II: Edge Midpoint calculation. Values at the points in the center of the four exterior edges, are obtained by averaging corner points nearest to them, and adding a random number generated using a self-similar process.

An array of 5X5 has been used in Fig.(3) to illustrate the above steps.

The characteristics of the random number are important for cloud generation. As mentioned earlier, clouds exhibit $1/f$ spectral characteristics. We exploit this property to generate the random number. The $1/f$ process has two important properties:

a. Strong interdependence between far-apart observations. Such strong correlation is the result of the high energy content of $1/f$ spectral density at low frequencies. So $1/f$ process is also referred to as long-term correlated or long-memory process.

b. A second characterization of $1/f$ process is *statistical self-similarity,* i.e., the statistical properties of the process remains invariant to or within an amplitude factor under an arbitrary scaling of time axis.

In our investigation, we have exploited the above property of self-similar process for cloud texture generation. A rich class of self- similar process can be generated by the scale stationary process [5]. Definitions for the above processes follow next.

Let f be a signal on the positive real axis satisfying the following in-variance property.

$$\int_0^\infty |f(t)|^2 \, d\ln t = \int_0^\infty |f(\lambda t)|^2 \, d\ln t \qquad \forall\, \lambda > 0 \tag{8}$$

We call f a *scale-invariant* finite energy signal and denote signal space by $L^2(R^+, dln\, t)$. The energy of the signal remains invariant under arbitrary scaling of time axis. We can extend the class of scale-invariant signals by allowing a change in energy proportional to the scaling factor, i.e.

$$\int_0^\infty \left|\widetilde{f}(t)\right| \frac{1}{t^{2H}} d\ln t = \lambda^{-2H} \int_0^\infty \left|\widetilde{f}(\lambda t)\right|^2 \frac{1}{t^{2H}} d\ln t \tag{9}$$

where, $\lambda > 0 \, and \, -\infty < H < \infty$.

Here, $\widetilde{f}$ is a finite energy *self-similar* signal with parameter H. It can be shown that given f, $\widetilde{f}$, and $H \neq 0$, the following relation is valid [5].

$$f(t) = t^{-H} \widetilde{f}(t) \qquad \forall\, t > 0 \tag{10}$$

Given any shift stationary process $Y(t)$, $-\infty < t < \infty$, the process $X(\lambda)$, $\lambda > 0$, obtained through the following exponential distortion of time axis Eq. (11).

$$X(\lambda) = Y(\ln \lambda) \qquad \forall\, \lambda > 0 \tag{11}$$

is scale stationary. Scale stationary processes allow us to construct a class of self-similar processes with arbitrary self-similarity parameter H $(-\infty < H < \infty)$, as follows,

$$X(t) = t^H \widetilde{X}(t) \qquad \forall\, t > 0, \qquad \text{for some} -\infty < H < \infty \tag{12}$$

Then, $\{X(t),\ t > 0\}$ is a statistically self-similar process with parameter H, if $\{\widetilde{X}(t)\, t > 0\}$ is a scale stationary process.

Using, the above we can construct a self-similar process. Consider a shift stationary process $y(t)$, with φ being a random variable uniformly distributed in $(-\pi, \pi)$ and constant f_o,

$$y(t) = a\cos(2\pi f_o t + \varphi) \qquad \forall\, t > 0 \tag{13}$$

We can generate a scale stationary process $x(t)$ from the shift stationary process $y(t)$ as follows,

$$x(t) = y(\ln t) \qquad \forall\, t > 0 \tag{14}$$

$$x(t) = a\cos(2\pi f_o \ln t + \varphi) \tag{15}$$

Similarly, using Eq.(12) we can generate a self-similar process $z(t)$,

$$z(t) = at^H \cos(2\pi f_o \ln t + \varphi) \qquad \forall\, t > 0 \tag{16}$$

$$z(t) = at^H \cos(2\pi f_o \ln(t * (1+q)) + \varphi) \qquad \forall\, t > 0 \tag{17}$$

where, q is a uniformly random variable with range less than $(-0.25, 0.25)$.

The random numbers q and φ add randomness to the texture.

3 Radiance Calculation of Cloud

For cloud radiance and transmittance calculations, we use MODTRAN 4.0 software. To assign radiance values to the texture generated, we need to do the radiance calculation for the following :

a. Average radiance of clouds, and
b. Radiance for possible holes in the clouds.

The texture values modulates the average radiance of the cloud. Use of modtran 4.0 makes radiometric calculation accurate.

4 Texture to Radiance Mapping

Once the approximate radiance values have been calculated, a specific radiance must be assigned to each pixel according to its texture value. First, a threshold, Th, is chosen for the clear area. Pixels with texture values below the threshold are assigned the radiance value calculated for a hole in the cloud. The contour of the texture produces a realistic outline for the cloud. Next, a texture value, Te, is chosen for the cloud-edge. Pixels with values above the cloud-edge threshold are considered within the main body of the cloud, whereas pixel values between the cloud-edge and clear sky thresholds are in the translucent region. A contrast parameter allows us to specify the range of radiance variations within the main body of the cloud, which may be due to variations in the cloud thickness or temperature.

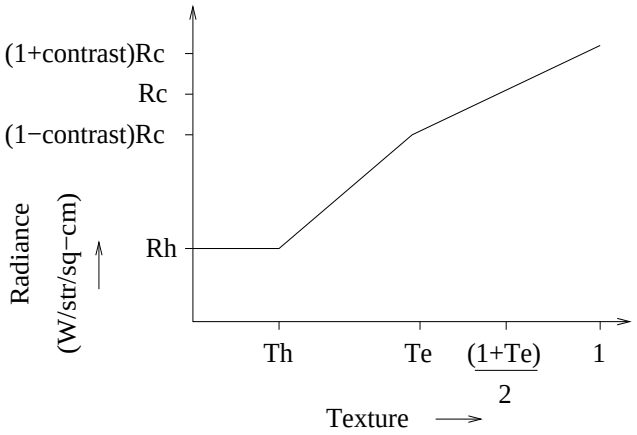


Figure 4: Texture to radiance mapping

The radiance values for pixels within the main body of the cloud are linearly interpolated, with the midpoint, (1+Te)/2, of the texture region, given the average cloud radiance, Rc, and the end points corresponding to (1−contrast) and (1+contrast) times the average cloud radiance, Rc. Pixels in the translucent edge region, are linearly interpolated between the radiance of the hole, Rh, and the radiance, (1−contrast)Rc, at the cloud edge threshold. Fig.(4) illustrates this mapping of texture value to radiance.

5 Radiance Calculation of Aircraft

We have used 3D geometric models of aircraft consisting of planar triangular facets defined by three spatial coordinates corresponding to the vertices of the facets. These "tile-like" triangles are grouped together to form various elements of the aircraft such as nose, wings etc. For radiance calculations of the aircraft, we assume typical temperatures for various elements. These elements are shown in Fig.(5). Modtran 4.0 is used for the radiance calculation of these elements.

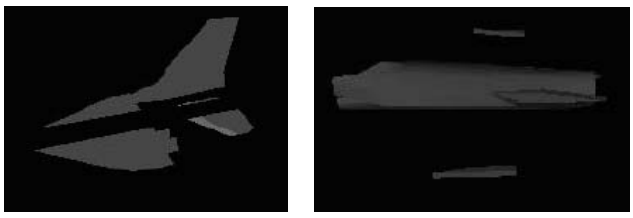


Figure 5: Various elements of aircraft

6 Scene Rendering

Now that various components of the simulated scene, i.e. clouds, aircraft model have been discussed, we address the process whereby these components are assembled into a cohesive scene for use with VRML, the rendering language. VRML uses a geometric based rendering. The background scene consists of 3D cloud objects, generated using images obtained from IR cloud generation. After the radiance calculation of aircraft, it is ready for insertion into the background scene. The GUI developed for this purpose allows us to specify the desired trajectory to the aircraft.

7 Results

The cloud images generated using MGM and SSA are shown in Fig.(6). Clouds generated using SSA look more natural. The number of input parameters required by MGM is large, but this also gives us flexibility and control over the shape of clouds.

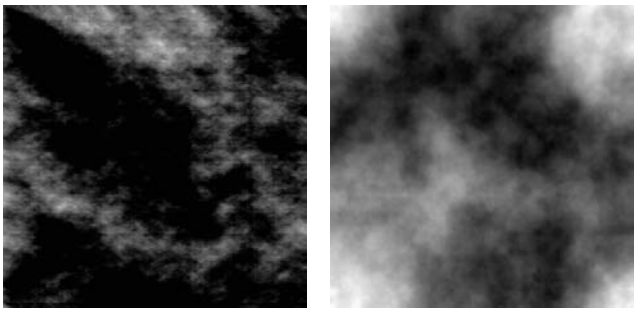


Figure 6: Comparision between images generated by MGM and SSA

In SSA, the smoothness of the cloud texture can be controlled using only one parameter, H. Fig.(7) shows the comparison between cloud images generated using different values for H. The smoothness of clouds increases for increasing values of H.

Fig.(8) shows IR images obtained from our IR Scene Simulator for air-borne targets. The images show aircraft maneuvering through the clouds. We can specify desired aircraft trajectories in our scene simulator. It uses B-Spline interpolation between points specified for the trajectories.

8 Conclusion

The IR scene generated by our simulator provides data under different conditions for evaluation of tracking and detection algorithms. Cloud texture can be chosen as desired. The cloud texture generated by our MGM and SSA gives better results than Gardner's method. Objectization of clouds and aircraft facilitates easy incorporation of different simulation conditions.

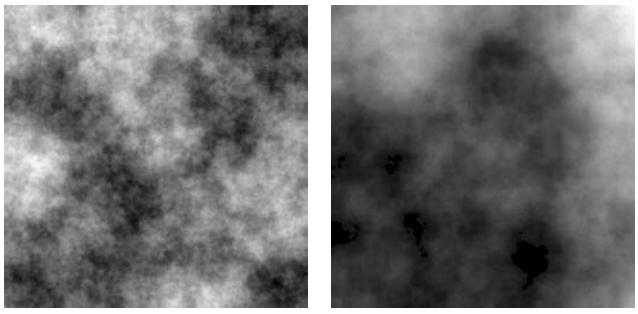


Figure 7: Cloud generated by SSA with H=0.5 and H=0.9

9 Acknowledgment

This research project is supported by Department of Extramural Research and Intellectual Property Rights, DRDO, New Delhi.

8(a)

8(b)

8(c)

Figure 8: IR scene showing aircraft manuevering through clouds

References

[1] *Modtran 4.0* user manual.

[2] A. D. Sheffer Jr., Michael Cathacart, "Generation and application of high resolution infrared computer imagery", *Optical Engineering*, Vol. 30 No. 11, Nov 1991.

[3] Geoffery Gardner, "Visual Simulation of *Clouds"*, *Computer graphics*, vol.19, 3, pp 297-304, 1985.

[4] J. Conant, D. Dvore, and J.Grunizer, "Synthetic IR cloud image model", *SPIE Multispectral Image Processing and Enhancement,* Vol. 933, pp 270-280, 1988.

[5] Birsen Yazici, Rangaswami SSA, "A Class of Second-Order Stationary Self-Similar Processes for $1/f$ phenomena", *IEEE Transaction on Signal Processing,* Vol. 45 No. 2, Feb 1997.

[6] Miller, Gavin S. P., "The Definition and Rendering of Terrain Maps", *SIGGRAPH Conference Proceedings* Computer Graphics, Volume 20, Number 4, August 1986.

[7] S. P. Mahulikar, S. K. Sane, U. N. Gaitonde and A. G. Marathe, "Numerical studies of infrared signature levels of complete aircraft", *The Aeronautical Journal*, pp 185-192, April 2001.