Scene Generation for Tracking Small Targets Against Cloud Backgrounds

John N. Sanders-Reed and W. Michael Dimmler

Boeing-SVS, Inc.
4411 The 25 Way, NE
Albuquerque, NM 87109
JNS@SVSINC.COM

ABSTRACT

Techniques for creating accurate scenes of small targets against high clutter cloud backgrounds have been developed. This scene generation capability is a crucial first step in developing acquisition, tracking, and mode logic algorithms for real time implementation. This scene generation capability features multiple targets combined with cloud backgrounds and numerous options for real world sensor effects. The generated frames are radiometrically and spatially accurate to sub-pixel precision.

Keywords: Scene generation, clouds, clutter, backgrounds, tracking

1. INTRODUCTION

A realistic tracking algorithm development simulation test-bed is an invaluable tool to fully benefit from ever increasing real-time image processing capabilities. Creating representative, realistic, and accurate 2 dimensional images of scene containing targets, backgrounds and sensor effects is a critical component of this algorithm development test-bed. The speed and flexibility offered by such an algorithm test bed capability dramatically reduces tracker design, development, and testing costs. Furthermore, algorithm performance and robustness is vastly improved by allowing algorithm designers to experiment against countless scenarios and combinations of targets, backgrounds, and sensors imperfections.

The simulation includes scene background effects, target characteristics, atmospheric propagation, optical system effects, detector sampling effects and noise. These scenes were used to verify algorithm performance in the areas of clutter suppression and target detection, track accuracy, and latency.

This paper describes the scene generation and methodology used.

2. SCENE GENERATION

2.1 Overview

In order to support tracker algorithm development and performance validation, a physics based, radiometrically and spatially representative scene generation tool was developed (Figure 1). The scene generation begins with a cloud background model which is both radiometrically correct and which includes proper edge and intensity statistics. Line Of Sight (LOS) motion smearing can be added to this background to simulate the effects of LOS motion during a single frame integration time. A radiometrically accurate target model is inserted into the scene at high resolution (much higher than the eventual sensor sampling) in order to avoid aliasing and inaccurate edge effects. Next a range and elevation angle dependent atmospheric attenuation factor is applied. The image is then convolved with an Optical Point Spread Function (PSF) determined by the sensor's aperture diffraction effects and geometric aberrations. The image is also scaled by the optical system attenuation. Photon Poisson statistics are added to the input scene. Next the scene is sampled by the focal plane. This includes spatial quantization, detection quantum efficiency, spatial and temporal electronic noise effects, and finally intensity quantization. The scene generation tool was used to generate a number of different image sequences involving different levels of cloud cover, target ranges, and elevation angles.

Figure 1. Scene Generation Components

The scene generation process is described below in more detail.

2.2 Cloud Backgrounds

The cloud background scenes were generated using a technique developed by Lashansky [1-4]. His cloud model was developed by observing and analyzing a number of cloud scenes using MWIR and LWIR imagery with various amounts of cloud cover. Lashansky's technique uses a 2D Weiner spectrum (also known as the Power Spectral Density) to generate images which have realistic looking spatial distributions for various amounts of cloud cover. He then uses a histogram remapping technique to produce the correct radiometric distribution for clouds, sky, and transition regions.

The size of the cloud scene array should be significantly bigger than the focal plane array in order to approximate a continuous background. This is necessary to avoid introducing artifacts when the targets are added to the scene (next section). An over sampling of 4 times in each dimension was used. Thus if a 128x128 pixel final image is desired, we create a 512x512 pixel cloud scene. Note that the sensor array image and the cloud scene image are aligned so that the sensor array image pixel boundaries coincide with cloud scene pixel boundaries. No fidelity is gained by shifting the two arrays so they don't align and in fact unrealistic artifacts would be introduced.

2.2.1 Structure: Spatial Distribution

The 2D Weiner spectrum is formed by collecting a number of cloud scenes having a similar amount of cloud cover, removing sensor effects, taking the 2D Fourier transform of the images, taking the modulus of these transforms, and then averaging the results. By doing this with several different sets of images, each set having a different amount of cloud cover, Lashansky was able to develop a 2D Weiner spectrum equation which captured the spatial frequency effects of various amounts of cloud cover (equation 1):

$$S(u, v) = \frac{\sigma_R^2 K}{\left[\frac{1}{\beta^2} + (2\pi u)^2 + (2\pi v)^2 \right]^{\alpha/2}} \quad (1)$$

The parameters in equation (1) are the standard deviation of the cloud radiance (σ_R), the correlation length (β), and the two spatial frequency dimensions (u, v). The form of the parameter K is given by $K = 2\pi/\beta^{(\alpha-2)}$ for $\alpha = 3$ to 6. The correlation length is related to the percentage cloud cover in the scene by the equation:

$$\beta = 0.00333\sqrt{(\% \text{ cover})} + 0.00193 \quad \text{radians.} \quad (2)$$

The percent cloud cover in equation (2) is a number between 0 and 100. However, the only allowed numbers are 10, 50, and 90. For cloud covers of 10 and 90%, $\alpha = 3$ while for a cloud cover of 50%, $\alpha = 5$. The exact value of σ_R ($\mu\text{W}/\text{cm}^2/\text{sr}$) is not important since radiometric calibration is performed as part of the histogram modification.

In order to generate a cloud scene from the 2D Wiener spectrum (equation 1), it is necessary to introduce spatial frequency noise. This is accomplished using two random distributions. The first distribution R_1 is a normal distribution having zero mean and unit variance. The second distribution, R_2 is a uniform distribution between 0 and 2π . Both sets of random numbers are generated to have N^2 values (1 for each pixel location).

The Fourier domain “image” is now computed as:

$$F(u, v) = \sqrt{S(u, v)} R_1 (\cos R_2 + i \sin R_2) \quad , \quad (3)$$

where $S(u, v)$, R_1 , and R_2 are all real numbers. The values R_1 and R_2 in equation (3) are just individual values selected from the respective distributions. $F(u, v)$ of course is complex.

If the cloud scene image has N pixels in each dimension, then the spatial frequency increment, du and dv is given by $df = 1/(\text{IFOV} * N)$, where IFOV is the Incremental Field Of View, the field of view of a single pixel. If the final spatial domain image will have N pixels in each dimension, then in the Fourier domain the N pixel positions correspond to spatial frequency values of $-(N/2)*df$ to $+(N/2-1)*df$. Note that at this point the origin (zero frequency) is in the center of the image.

Since we wish to generate a strictly real image, we now impose several complex conjugate constraints. In essence we demand that $F(N-u, M-v) = F^*(u, v)$. A bit of care is required because the image does not go from $-(N/2)*df$ to $+(N/2)*df$ but rather it goes from $-(N/2)*df$ to $+(N/2-1)*df$. This requires special treatment of the $-(N/2)*df$ lines in the image as well as the 0 frequency lines in the image. Finally we impose the requirement that $F(0,0)$, $F(N/2,0)$, $F(0, N/2)$, $F(N/2, N/2)$ are all strictly real by simply setting their imaginary components to zero.

The Fourier domain image we have created has the origin in the center. Most FFT computer routines shuffle the quadrants around, so in order to use a standard FFT routine it is necessary to exchange the first and third quadrants and the second and fourth quadrants. Performing the FFT operation now yields a strictly real image. This image is a floating point image. Since in the next step we will form the histogram which requires binning, we want to convert this image to an integer representation. Since the final image, after target insertion, sensor sampling, and other operations will be a 12 bit image, we want this intermediate stage to have many bins per final output bit. Therefore we scale this image to be an integer with values from 0 to $2^{20}-1$.

The image at this point is not radiometrically calibrated and a histogram of the image is approximately gaussian. The edge transitions from cloud to sky are relatively soft. Actual cloud scenes were observed by Lashansky to have bi-modal or tri-modal distributions, one each for sky, cloud, and transition regions.

2.2.2 Histogram Remapping: Radiometric calibration

Histogram remapping uses two cumulative distribution functions, that of the original image, and the desired target cumulative distribution function. For each function, the x-axis is the pixel grey scale value in the image, ranging from 0 to the maximum grey scale value (in our case $2^{20}-1$). The y-axis is the number of pixels in the image whose grey scale value is less than or equal to the value on the x-axis. The cumulative distribution function is thus a monotonically increasing function which can run either from 0 to the total number of pixels (N^2) or can be normalized to run from 0 to 1 (we used 0 to N^2). Histogram remapping is performed as follows. For each pixel in the original image, the pixel grey scale value is selected on the x-axis of the original image cumulative distribution function and the y-axis value is read off (number of pixels with that grey scale value or less). Now one finds the same y-value on the target cumulative distribution function graph and reads off the corresponding grey scale level from the x-axis. This is the new grey scale value for the pixel.

The histogram remapping function accomplishes two tasks for us. First, it remaps the image histogram allowing us to “tune” the edge sharpness. Second, it allows us to impose different radiometric mean values on different portions of the image, such as sky, cloud, and transition regions. Lashansky divides his cloud scenes into three regions: cloud, sky, and transition. Lashansky models each region as a normal distribution having a unique mean and standard deviation. Thus a histogram of a “real” image in the Lashansky view will have three distinct gaussian peaks, one for each region.

Inputs to the Lashansky histogram remapping include the mean and standard deviation of the radiance for each of the three regions, and the percent of the image which contains cloud. When performing the remapping, we select a minimum and maximum radiance for the remapping. We chose as a minimum radiance the sky mean radiance minus 6 times the sky standard deviation. For the maximum radiance we chose the cloud mean radiance plus 6 times the cloud standard deviation. Given the minimum and maximum radiance values and the dynamic range of the image in grey levels (2^{20}), we can derive a scaling factor and convert the radiance values to digital grey scale values. The scaling factor is simply the (maximum

radiance – minimum radiance) / number of grey levels. The digital value of the sky mean value then is the sky mean radiance minus the minimum radiance divided by the scale factor.

Since we will need the cumulative distribution function of the original image, we form the histogram of the original image and then convert that to the cumulative distribution function. We do not normalize the cumulative distribution function but rather leave it as a monotonically increasing function with values from 0 to N^2 . Note that the x-axis of the cumulative distribution function runs from 0 to $2^{20}-1$.

Next we need to generate the target cumulative distribution function. Since the desired histogram is a sum of 3 gaussian distributions, the target cumulative distribution function will be a weighted sum of 3 error functions. Since most standard computer error functions ($\text{erf}(x)$) take on values of 0-1 as x goes from 0 to infinity, rather than as x goes from $-\infty$ to $+\infty$, a bit of care must be taken. We scale the target cumulative distribution function to have values from 0 to N^2 .

In order to simplify the remapping process, we invert the target cumulative distribution function into a look-up table of grey scale as a function of cumulative number of pixels, so we do not have to search the target cumulative distribution function for the correct y-axis value. The remapping is now simply done by selecting each pixel in the original image, using its value as the index into the original image cumulative distribution function (to give the number of pixels with that grey value or less), and then using that result as the index into the inverted target cumulative distribution function so we can read off the desired grey scale value. A typical resulting histogram is shown in figure 2.

Figure 2. Remapped histogram

The remapped digital values can be converted to radiometric units by multiplying by our radiometric scale factor and then adding the minimum radiance value. The result is stored as an image having floating point pixel values which are in radiance units.

We obtained values for the mean and standard deviation of the three regions by analyzing imagery collected at several different geographic locations of interest. Various cloud scenes were generated using these numbers.

2.2.3 Motion Smearing

One of the major assumptions of many track algorithms is that the target is being tracked and hence is relatively stationary in the image, while the background appears to move. This background motion can be significant, on the order of several pixels during a frame integration time. This background motion during a frame time will tend to smear the background and soften any edges in the background (while the target will remain stationary and not suffer this edge softening effect). This in turn will make the detection problem easier and hence is an important effect to include in the background scene generation.

Our approach is to Fourier transform the original image generated in the preceding steps. Now in the Fourier domain we apply a smearing operator which effectively smears the image with the amount of background motion. We also generate a new Fourier domain image which is just the original image, shifted by the amount of single frame motion (this will be the input to the next frame smearing). The smeared image is now back transformed to generate a smeared image in real space.

The background motion smearing operator is:

$$S(\vec{k}_{mn}, \vec{a}) = \frac{i}{\vec{k}_{mn} \bullet \vec{a}} \left(1 - e^{i\vec{k}_{mn} \bullet \vec{a}} \right) \quad (4)$$

where $\mathbf{a}$ is the LOS motion per frame and $\mathbf{k}$ is the wave vector. This complex smearing operator is simply multiplied by the Fourier domain image to provide the motion smear. The result is back transformed as just described to get the actual motion smeared image. Energy is conserved in this operation so there are no radiometric adjustments necessary. Note that as $\mathbf{k}\mathbf{a}$ goes to zero, $S(\mathbf{k}, \mathbf{a})$ goes to 1. In order to avoid a division by zero it is necessary to impose a check that if $\mathbf{k}\mathbf{a} = 0$, then $S = 1$ (strictly real).

To simply offset an image by an amount $\mathbf{a}$, we multiply the Fourier domain image by the phase factor $e^{-i\mathbf{k}\mathbf{a}}$. In this notation, both $\mathbf{k}$ and $\mathbf{a}$ are two dimensional vectors and the product is a dot product. For most image sequences we set $\mathbf{a}$ equal to a constant, corresponding to constant LOS motion. Note that the image simply “wraps around”; as a portion of the image is shifted out of the image on one side, it reappears on the other side of the image. We use this and impose the constraint on the image sequence that $n\mathbf{k}\mathbf{a} = m2\pi$ where n is the number of frames in the image sequence and m is an integer. In other words, at the end of an image sequence, the cloud scene has just returned to its original position. This (along with proper target motion constraints), allows us to play an image sequence as a continuous loop without jumps in the background or target positions.

2.3 Target Model and Insertion

Two problems must be addressed when the scene is to include a target. The first is the generation of a target image and the second is how to properly insert the target into the scene. The second problem is important because if the target is not properly inserted into the scene it is possible to easily produce unrealistically strong edges, which can be readily detected by an edge detection algorithm. In this case a track algorithm might work very well on the simulated imagery but fail completely against real world targets.

A silhouette of a representative target is generated using an analytical model (e.g. a CAD type silhouette). The actual target length can be specified as part of the target model. This silhouette can now be rotated and scaled by range as desired. We also assume that the target is a uniform temperature grey body emitter. For a given temperature, the black body radiance is integrated over the sensor waveband and scaled by the emissivity to produce the target in-band radiance. The target is initially imbedded in a zero radiance background.

Recall that the cloud scene was generated with a single pixel IFOV which was 1/4 the sensor single pixel IFOV in each dimension. A target scene having a single pixel IFOV which is 1/16 the sensor single pixel IFOV or 1/4 the cloud scene IFOV is prepared. This grid is aligned with the sensor (and cloud scene) pixel boundaries. This means that for every cloud pixel there will be a 4x4 array of target pixels, and for every sensor pixel there will be a 4x4 array of cloud pixels (Fig. 3). The continuous, scaled target model is now positioned within the target scene grid to sub-pixel (double precision) accuracy and sampled by the target scene grid. The target scene and the cloud scene must now be combined to produce the sensor input scene.

Figure 3. Sensor, Cloud Scene, and Target Scene relative IFOV

A transmissivity map is generated for the cloud scene. The transmissivity is defined as the fraction of energy from the background which is transmitted through the target scene. If a cloud scene pixel is covered with target scene pixels containing no target (i.e. zero radiance), the transmissivity is 1. Similarly, if the cloud scene pixel is completely covered with target pixels from the target scene, the transmissivity is 0. If the cloud scene pixel is only partially covered by target pixels

from the target scene, then the transmissivity is the fraction of the cloud scene pixel which is covered by zero radiance pixels from the target scene. This transmissivity is multiplied by the cloud scene pixel value and used as one of two additive contributions to the sensor input scene. Next the target scene pixels overlaying the cloud scene pixel are summed. We can sum the zero radiance pixels along with the target pixels, because the zero radiance pixels will make no contribution.

Typically, a number of targets are placed at various points in the scene. In addition, the targets are generally designed to move from frame to frame. In order to allow the image sequence to be played as a continuous loop, we require that the targets return to their original position in the imagery, in the last frame of the sequence.

2.4 Atmospheric Propagation

Two effects are captured in the atmospheric propagation model. One is range and the other is atmospheric attenuation. For most of the image sequences, range can be considered approximately constant.

For resolved targets the conversion from target radiance to focal plane irradiance is obtained by multiplying the target radiance by the projected area of a pixel $A_p = (\text{IFOV})^2 * R^2$, and then by the solid angle of the optical system aperture seen from the target, $\Omega_o = \pi D^2 / 4R^2$, where R is the range from the sensor to the target. Note that the range cancels and need not be specified in this calculation. For unresolved targets there is a $1/R^2$ range effect. In either case, this is a simple scalar number which multiplies the radiometric image, transforming scene radiance to sensor irradiance.

For all targets, atmospheric attenuation must be considered. Atmospheric attenuation is computed using atmospheric modeling codes, such as ModTran [5-7], which give attenuation as a function of season, location, elevation angle, range, wavelength pass-band, and various atmospheric conditions. The result is a simple scalar number which is used to multiply the radiometric image.

Since both of these atmospheric effects simply uniformly scale the image intensity, the only effect that they have is on processes dependent on absolute image intensity. These are photon Poisson statistics, and photo-electron to sensor noise electron ratios.

2.5 Optical System Effects

Optical system effects include optical attenuation and point spread function blurring due to a finite aperture in the optics. For a circular aperture, the angular smear will be a gaussian of $4\lambda/\pi D$ radians diameter at the $1/e^2$ points, where λ is the wavelength of light (we choose the center wavelength of the pass-band) and D is the clear aperture diameter. A gaussian of this diameter is convolved with the sensor input scene to introduce the point spread function blurring, resulting from diffraction. Optical attenuation is simply a scalar factor applied to the entire image.

2.6 Detector Sampling

In this step we convert from sensor pixel irradiance to sensor photoelectrons and then to digital counts. The result of the preceding operations is a radiometrically correct input scene for the sensor which contains a minimum of edge effects due to aliasing. This input scene is over-sampled by a factor of 4 in each dimension compared to the sensor (the scene single pixel IFOV is 1/4 the sensor single pixel IFOV). We begin by summing the input scene pixels on a 4x4 grid to convert to sensor pixels. Now the sensor irradiance is converted from Watts/pixel to photons/sec/pixel using a conversion factor of 5×10^{18} photons/sec/Watt. Photon shot noise is introduced at this point using the standard Poisson model in which the standard deviation of the noise is equal to the square root of the number of photons. The result is multiplied by the pixel fill factor and the quantum efficiency to obtain a photo-current. Next we multiply by the frame integration time to obtain the total number of photo-electrons.

Detectors exhibit a dark current which accumulates over time. The rate at which dark current accumulates varies from pixel to pixel, resulting in a spatial non-uniformity. In addition, the amount of dark current exhibits a temporal variation from one frame to the next. Once a frame integration time has elapsed, the charge is read out of each pixel and converted to a voltage, which is then quantized. We assume that the well capacity of the pixels is much greater than the dynamic range of the Analog to Digital (A/D) converter. In this case, the output voltage is shifted prior to the A/D conversion, so that the variation lies within the range of the converter. With this assumption, we can ignore the constant, integration time dependent number of detector generated electrons, and concentrate solely on the spatial and temporal variation. Detector noise is therefore specified as both a spatial and temporal standard deviation in the number of noise electrons.

A gaussian random number generator is used to generate one, temporally constant, spatial noise map. Each entry in this map specifies the number of spatial noise electrons to be added to a given pixel in each image. For the temporal noise, a new

gaussian noise map is generated for each frame. Once spatial and temporal noise electrons are added to each image pixel to give the total number of electrons which will be digitized.

The final step in the scene generation process is to quantize the number of electrons into image grey scale levels. A 12-bit (0 to 4095) A/D converter is used. We assume that an operator in a real system will be able to adjust a gain and offset factor to allow the image sequence to fill the dynamic range of the A/D without saturating. Therefore, we set the full scale (4095) level slightly above the expected largest number of electrons, and the minimum value (0), slightly below the smallest number of electrons.

The preceding procedure is used to generate sequences of imagery in which the clouds are observed to move at constant velocity across the imagery, while the targets move independently. In order to provide a continuous image sequence as input to the tracker, the cloud and target motion are arranged so that at the end of the sequence they have just returned to their original positions.

3. SAMPLE IMAGERY

A sample image is shown below in figure 4. The image shows a typical cloud distribution pattern, with sensor noise effects and four targets. The image wide speckle represents sensor spatial and temporal noise.

Figure 4. Simulated cloud scene with 4 targets.

4. SUMMARY

A scene generator capable of creating accurate scenes of small targets against high clutter cloud backgrounds has been developed. Spatially over-sampled cloud backgrounds are generated using 2 dimensional Wiener spectrums, followed by a histogram re-mapping. The Wiener spectrum produces a cloud background with representative spatial frequency content, while the histogram re-mapping provides radiometric calibration and scene radiance distribution (image statistics). A technique for combining these backgrounds with targets, which accurately preserves the image spatial characteristics, was presented. Target and sensor motion effects using Fourier sub-sampling and combining techniques. Finally, numerous image degradation effects are applied accounting for numerous effects including atmospheric, optical system, detector sampling, noise, calibration, and FPA imperfections.

This scene generation capability can be used to create realistic and real world representative scenes for track algorithm development. Such a simulation test bed capability offers designers an algorithm development environment that cannot practically be duplicated in hardware and test set ups within any reasonable cost. Furthermore, algorithm performance and robustness is vastly improved by allowing algorithm designers to experiment against countless scenarios and combinations of targets, backgrounds, and sensors imperfections.

REFERENCES

- S.N. Lashansky, N. Ben-Yosef, A. Weitz, "Spatial analysis of ground based infrared cloudy sky images", Optical Engineering, **32**, (6) 1993, pp 1272.
1. S. Lashansky, N. Ben-Yosef, A. Weitz, E. Agassi, "Preprocessing of ground-based infrared sky images to obtain the true statistical behavior of the image", Optical Engineering, **30** (12) 1991, pp 1892.

2. S. Lashansky, N. Ben-Yosef, A. Weitz, "Segmentation and statistical analysis of ground-based infrared cloudy sky images", Optical Engineering, **31** (5) 1992, pp 1057.
3. S.N. Lashansky, N. Ben-Yosef, A. Weitz, "Simulation of ground-based infrared cloudy sky images", Optical Engineering, **32** (6), 1993, pp 1290.
4. A. Berk, G.P. Anderson, P.K. Acharya, J.H. Chetwynd, L.S. Bernstein, E.P. Shettle, M.W. Matthew, S.M. Adler-Golden, "Modtran 4 user's Manual", Air Force Research Laboratory, Space Vehicles Directorate, Air Force Material Command, Hanscom AFB, MA 01731-3010, June 1, 1999.
5. A. Berk, L.S. Bernstein, D.C. Robertson, "MODTRAN: A Moderate Resolution Model for LOWTRAN 7", GL-TR-89-0122, AFRL/VSBM, 29 Randolph Rd., Hanscom AFB, MA 0173-3010
6. F.X. Kneizys, E.P. Shettle, L.W. Abreu, J.H. Chetwynd, G.P. Anderson, W.O. Gallery, J.E.A. Selby, S.A. Clough, "Users Guide to LOWTRAN 7," AFGL-TR-88-0177, Geophysics Directorate/GPOS, 29 Randolph Rd., Hanscom AFB, MA 01731-3010, (August 1988) ADA206773.