SensorFX

Users Guide

JRM Technologies
Real Physics • Real Sensors • Real-time

VT MAK
A company of VT Systems

SensorFX

Users Guide

Version 2.5

JRM Technologies
Real Physics • Real Sensors • Real-time

VT MAK
A company of VT Systems

VT MAK
150 Cambridge Park Drive, 3rd Floor
Cambridge, MA 02140 USA

Voice: 617-876-8085
Fax: 617-876-9208

info@mak.com

www.mak.com

Revision VRV-2.5-11-190221

Contents

Preface

Chapter 1. Using SensorFX

Chapter 2. Adding GenesisMC Models to SensorFX

Chapter 3. Setting Sensor Parameters Using the API

Index

Preface

This manual is for persons who will install SensorFX and use it to view simulations. The manual assumes that you are familiar with basic administrative tasks and the graphical window environment for your operating system. SensorFX is a plug-in to VR-Vantage. This manual assumes you are familiar with use of VR-Vantage.

How the Manual Is Organized

This manual has the following chapters:

- Chapter 1, *Using SensorFX* explains how to install and use SensorFX in VR-Vantage. This chapter is the user-level documentation for SensorFX.
- Chapter 2, *Adding GenesisMC Models to SensorFX* explains how to add the files created using GenesisMC to SensorFX. GenesisMC is the application from JRM Technologies that you use to sensorize terrain and models.
- Chapter 3, *Setting Sensor Parameters Using the API* explains how to use the SensorFX API to set sensor parameters.

MAK Products

SensorFX is a member of the VT MAK line of software products designed to streamline the process of developing and using networked simulated environments. The VT MAK product line includes the following:

- VR-Link® Network Toolkit. VR-Link is an object-oriented library of C++ functions and definitions that implement the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS) protocol. VR-Link has built-in support for the RPR FOM and allows you to map to other FOMs. This library minimizes the time and effort required to build and maintain new HLA or DIS-compliant applications, and to integrate such compliance into existing applications.

  VR-Link includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write applications. The executables provide valuable debugging services such as generating a predictable stream of HLA or DIS messages, and displaying the contents of messages transmitted on the network.

- MAK RTI. An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MAK RTI is optimized for high performance. It has an API, RTIspy®, that allows you to extend the RTI using plug-in modules. It also has a graphical user interface (the RTI Assistant) that helps users with configuration tasks and managing federates and federations.
- VR-Forces®. VR-Forces is a computer generated forces application and toolkit. It provides an application with a GUI, that gives you a 2D and 3D views of a simulated environment.

  You can create and view entities, aggregate them into hierarchical units, assign tasks, set state parameters, and create plans that have tasks, set statements, and conditional statements. You can simulate using entity-level modeling, which focuses on the actions of individual people and vehicles, and aggregate-level modeling, which focuses on the interaction of large hierarchical units.

  VR-Forces also functions as a plan view display for viewing remote simulation objects taking part in an exercise. Using the toolkit, you can extend the VR-Forces application or create your own application for use with another user interface.

- VR-Vantage™. VR-Vantage is a line of products designed to meet your simulation visualization needs. It includes three end-user applications (VR-Vantage Stealth, VR-Vantage PVD, and VR-Vantage IG) and the VR-Vantage Toolkit.
  - VR-Vantage Stealth displays a realistic, 3D view of your virtual world, a 2D plan view, and an exaggerated reality (XR) view. Together these views provide both situational awareness and the big picture of the simulated world. You can move your viewpoint to any location in the 3D world and can attach it to simulation objects so that it moves as they do.

  - VR-Vantage IG is a configurable desktop image generator (IG) for out the window (OTW) scenes and remote camera views. It has most of the features of the Stealth, but is optimized for its IG function.
  - VR-Vantage PVD provides a 2D plan view display. It gives you the big picture of the simulated world.
  - SensorFX. SensorFX is an enhanced version of VR-Vantage that uses physics based sensors to view terrain and simulation object models that have been materially classified. It is built in partnership with JRM Technologies.
  - The VR-Vantage Toolkit is a 3D visual application development toolkit. Use it to customize or extend MAK's VR-Vantage applications, or to integrate VR-Vantage capabilities into your custom applications. VR-Vantage is built on top of OpenSceneGraph (OSG). The toolkit includes the OSG version used to build VR-Vantage.
- MAK Data Logger. The Data Logger, also called the Logger, can record HLA and DIS exercises and play them back for after-action review. You can play a recorded file at speeds above or below normal and can quickly jump to areas of interest. The Logger has a GUI and a text interface. The Logger API allows you to extend the Logger using plug-in modules or embed the Logger into your own application. The Logger editing features let you merge, trim, and offset Logger recordings.
- VR-Exchange™. VR-Exchange allows simulations that use incompatible communications protocols to interoperate. For example, within the HLA world, using VR-Exchange, federations using the HLA RPR FOM 1.0 can interoperate with simulations using RPR FOM 2.0, or federations using different RTIs can interoperate. VR-Exchange supports HLA, TENA, and DIS translation.
- VR-TheWorld™ Server. VR-TheWorld Server is a simple, yet powerful, web-based streaming terrain server, developed in conjunction with Pelican Mapping. Delivered with a global base map, you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private, classified networks to provide streaming terrain data to a variety of simulation and visualization applications behind your firewall.
- DI-Guy™. The DI-Guy product line is a set of software tools for real-time human visualization, simulation, and artificial intelligence. Every DI-Guy software offering comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy enables the easy creation of crowds and individuals who are terrain aware, autonomous, and react intelligently to ongoing events. Save time, money and create outstanding simulations with DI-Guy. The DI-Guy product line includes the following products:
  - The DI-Guy SDK. Embed the DI-Guy library in your real-time application and populate your world with lifelike human characters.
  - DI-Guy Scenario™. Author and visualize human performances in a rich, user-friendly graphical environment. Use DI-Guy Scenario as an end visualization application or save scenarios and load them into your DI-Guy SDK enabled application.

  - ECOSim. Enhanced Company Operations Simulation (ECOSim) is a company-level training simulation that teaches leaders how best to deploy troops, UAVs, convoys, and other assets. ECOSim focuses on ease-of-use, rapid scenario generation, runtime operator control, and realistic and reactive human simulation.
  - DI-Guy AI. Generate crowds of autonomous characters to quickly populate your worlds with hundreds and thousands of terrain-aware, collision avoiding DI-Guys. Used as a module on top of DI-Guy Scenario and DI-Guy SDK.
  - Expressive Faces Module. Enable DI-Guy characters to have faces that display emotion, eyes that look in directions and blink, and lips that sync to sound files.
  - DI-Guy Motion Editor. Create or customize motions to your particular needs in an easy-to-use graphical application.
- RadarFX. RadarFX is a client-server application that simulates synthetic-aperture radar (SAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests SAR images from the server. RadarFX includes a sample client application.
- VR-Engage. VR-Engage is a multi-role virtual simulator that lets users play the role of a first person human character, a ground vehicle driver, gunner or commander, or the pilot of a fixed wing aircraft or helicopter. It incorporates the VR-Force simulation engine and the VR-Vantage graphics rendering capabilities.
- WebLVC Server. WebLVC Server implements the server side of the WebLVC protocol so that web-based simulation federates can participate in a distributed simulation. It is part of the WebLVC Suite, which includes the server and several sample applications that work with VR-Forces and VR-Vantage.

How to Contact Us

For SensorFX technical support, information about upgrades, and information about other MAK products, you can contact us in the following ways:

Telephone

Call or fax us at:	Voice:	617-876-8085 (extension 3 for support)
	Fax:	617-876-9208

E-mail

Sales and upgrade information:	info@mak.com
Technical support:	support@mak.com

Internet

MAK web site home page:	www.mak.com
License key requests:	www.mak.com/support/get-licenses
Product version and platform information:	www.mak.com/support/product-versions
For the free, unlicensed MAK RTI:	www.mak.com/support/bonus-material

Post

Send postal correspondence to:	VT MAK 150 Cambridge Park Drive, 3rd Floor Cambridge, MA, USA 02140

When requesting support, please tell us the product you are using, the version, and the platform on which you are running.

Document Conventions

This manual uses the following typographic conventions:

`Monospaced`	Indicates commands or values you enter.
`Monospaced Bold`	Indicates a key on the keyboard.
`Monospaced Italic`	Indicates command variables that you replace with appropriate values.
Blue text	A hypertext link to another location in this manual or another manual in the documentation set.
`{ }`	Indicates required arguments.
`[ ]`	Indicates optional arguments.
`|`	Separates options in a command where only one option may be chosen at a time.
(|)	In command syntax, indicates equivalent alternatives for a command-line option, for example, `(-h | --help)`.
/	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.
Italic	Indicates a file name, pathname, or a class name.
sans Serif	Indicates a parameter or argument.
È	Indicates a one-step procedure.
Menu → Option	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as: Choose **File → Save**.
	Click the icon to run a tutorial video in the default browser.
	Indicates supplemental or clarifying information.
	Indicates additional information that you must observe to ensure the success of a procedure or other task.

Directory names are preceded with dot and slash characters that show their position with respect to the SensorFX home directory. For example, the directory *vrvantagex.x2.5/doc* appears in the text as *./doc*.

Mouse Button Naming Conventions

An instruction to click the mouse button, refers to clicking the primary mouse button, usually the left button for right-handed mice and the right button for left-handed mice. The context-sensitive menu, also called a popup menu or right-click menu, refers to the menu displayed when you click the secondary mouse button, usually the right button on right-handed mice and the left button on left-handed mice.

Third Party Licenses

MAK software products may use code from third parties. This section contains summary license information and where required, specific license documentation required by these third parties.

The following libraries are covered by the GNU Lesser General Public License (LGPL) license:

- CIGI
- DevIL
- GEOS
- GStreamer
- LibIconV
- OPCODE
- OpenAL
- OSG
- OsgEarth
- Pthread
- Qt
- Qwt
- LibWebSockets.

The following libraries are covered by the ZLib license:

- Bullet Physics
- Zlib
- LibPNG
- LibSDL
- Minizip.

The following libraries are covered by the MIT license:

- Expat
- FreeGLUT
- GDAL
- GeoTiff
- HarfBuzz
- LibCURL
- LibSquish
- LibXML
- LUA
- LuaBind
- Proj.

The following libraries are covered by the BSD license:

- FreeType
- GLEW
- JPEG
- LibTiff
- Protobuf
- PCRE.

The following libraries are covered by the commercial licenses:

- SpeedTree
- Sundog Triton
- Sundog SilverLining
- FlexLM
- Gameware
- JRM Technologies SenSimRT and SigSimRT
- GLStudio
- OpenFlight
- CDB API.

The following libraries are covered by custom licenses:

- NVidia CG Toolkit
- FreeImage (FreeImage Public License)
- Boost (Boost Software License)
- Poco (Boost Software License).

COLLADA DOM is covered by the SCEA Shared Source license.

Thread Building Blocks is covered by the GPL license:

SQLLite is in the public domain.

Boost License

VR-Link, and all MAK software that uses VR-Link uses some code that is distributed under the Boost License. All header files that contain Boost code are properly attributed. The Boost web site is: www.boost.org.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libXML and libICONV

VR-Link and all MAK software that uses VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MAK Products. MAK has made no modifications to these libraries. For more information about these libraries please see the following web sites:

- The LGPL license is available at: http://www.gnu.org/licenses/lgpl.html.
- Information about IconV is at: http://www.gnu.org/software/libiconv/.
- Information about LibXML is at: http://xmlsoft.org/.

1. Using SensorFX

This chapter introduces SensorFX, explains how to install it, and explains how to use it.

1.1. Introduction to SensorFX

SensorFX is a plug-in to VR-Vantage. SensorFX was created by JRM Technologies. It provides physics-based sensor views for Long Wave Infrared (LWIR), Midwave Infrared (MWIR), and night vision goggles (NVG). In physics-based sensor views, the rendering of terrain and simulated objects takes into account the material of the object, air and material temperature, ambient light, and the characteristics of the sensor. Therefore, SensorFX provides much more realistic sensors than the CameraFX sensors included in VR-Vantage, which are more akin to putting a colored filter on a camera.

To take advantage of physics-based sensors, the terrain and simulated objects must be "materially classified". That is, their composition must be specified and included in the terrain and model definitions. The ability to classify materials is not part of SensorFX or VR-Vantage. You must use the GenesisMC™ software from JRM.

SensorFX can apply material classification on demand for terrains loaded using osgEarth earth files.

Material classifications have been provided for the *Ala Moana.mtf* and *VR-Village.mtf* terrain databases. Scenarios that use these terrains will probably reference entities and effects that have not been material classified, so they may not look correct. The material-classifications are notional only to provide some indication of capability, but not reflect actual vehicle signatures.

The terrains *MAK Earth - MCOD (online).mtf* and *MAK Earth - MCOD with Lights (online).mtf* are configured for material classification on demand.

- SensorFX supports only one observer at a time being fully sensorized. If you open additional windows that use different observers, for example, inset views, they will show a sensor view similar to that provided by the VR-Vantage CameraFX sensors. (However, note that when you run the SensorFX version of VR-Vantage , the CameraFX sensors are not directly available as user options.)
- SensorFX terrains use much more texture memory than non-sensorFX terrains. If the amount of memory your terrain requires exceeds the capability of your video card, performance can drop dramatically. You can check the performance of your video card by enabling VR-Vantage's performance statistics display.
- When the SensorFX plug-in is loaded, VR-Vantage disables indirect rendering.

1.2. Installing SensorFX

On Windows, SensorFX is provided as a standard Windows installer and a data package. To install it, run the installer. Be sure to install it into the version of VR-Vantage for which it was compiled, for example, VR-Vantage 2.5, VC 12.

On Linux, SensorFX is supplied as a compressed tar file. To install it, uncompress it and untar it in the VR-Vantage installation directory. Do the same with the data package.

1.2.1. Installing the SensorFX License

SensorFX is licensed separately from VR-Vantage. To request a license, contact keys@mak.com. For information about installing the license, please see Chapter 2 in *VR-Vantage Users Guide*, which is in the *./doc* directory.

1.2.2. Uninstalling SensorFX

If you uninstall SensorFX and want to continue using VR-Vantage, you must reinstall the VR-Vantage application. This is because the SensorFX installer overwrites some files that VR-Vantage needs and these files get removed when SensorFX is uninstalled.

1.3. Starting VR-Vantage with SensorFX

VR-Vantage with SensorFX has its own startup shortcut. This allows you to run VR-Vantage with SensorFX or the standard VR-Vantage applications without SensorFX.

È To start VR-Vantage with SensorFX, on the Start menu, choose **MAK Technologies → VR-Vantage with SensorFX**.

SensorFX does not support geocentric terrains if they are part of a terrain that does not include an osgEarth terrain patch. For more information, please see "Using SensorFX with Geocentric Terrains," on page 1-27.

1.4. Using SensorFX

When SensorFX is installed, you choose the sensor to use by choosing an observer mode that has a sensor as part of its configuration, just as you do with the native CameraFX sensors. (For details, please see Section 12.2, "Enabling Sensors," in *VR-Vantage Users Guide*.) SensorFX includes the following observer modes:

- Stealth. Uses the Visual mode sensor. It displays the scene using the visual spectrum that is viewable by the human eye.
- MWIR (Out-the-Window). This sensor uses mid-wave infrared. MWIR is also called intermediate infrared (IIR): 3-8 µm. In guided missile technology the 3-5 µm portion of this band is the atmospheric window in which the homing heads of passive IR heat seeking missiles are designed to work. They home in on the infrared signature of the target aircraft, typically the jet engine exhaust plume. The default MWIR mode is white hot. You can change it to black hot on the Display Settings dialog box, Sensor Settings page.
- LWIR (Out-the-Window). This sensor uses long wave Infrared: 8–15 µm. This is the thermal imaging region, in which sensors can obtain a completely passive picture of the outside world based on thermal emissions only and requiring no external light or thermal source such as the sun, moon or infrared illuminator.
- NVG (Out-the-Window). This sensor mimics night-vision goggles, which allow images to be produced in levels of light approaching total darkness.

SensorFX works with multiple observers. However secondary observers do not provide materially classified effects. They provide the equivalent of CameraFX effects. If you have multiple windows that use the same observer, they all use SensorFX. (For information about CameraFX, please see Chapter 12, *Using Sensors*, in *VR-Vantage Users Guide*.) If you have multiple observers, you can specify which observer should be the SensorFX observer with the SENSORFX-_OBSERVER_NAME environment variable.

Clouds and precipitation are not correlated between the sensor modes and visual or no sensor modes.

When you switch observer modes, there may be a delay of several seconds before the new sensor takes effect.

1.4.1. Creating New Sensors

You can create new sensors with settings that vary from the installed IR and NVG sensors. The process for creating new sensors is generically the same as for creating new sensors with CameraFX. You add a new sensor on the Sensor Settings page and then you configure it. (For details, please see "Configuring SensorFX Sensors," on page 1-6.)

To add a sensor:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Sensor Settings page.

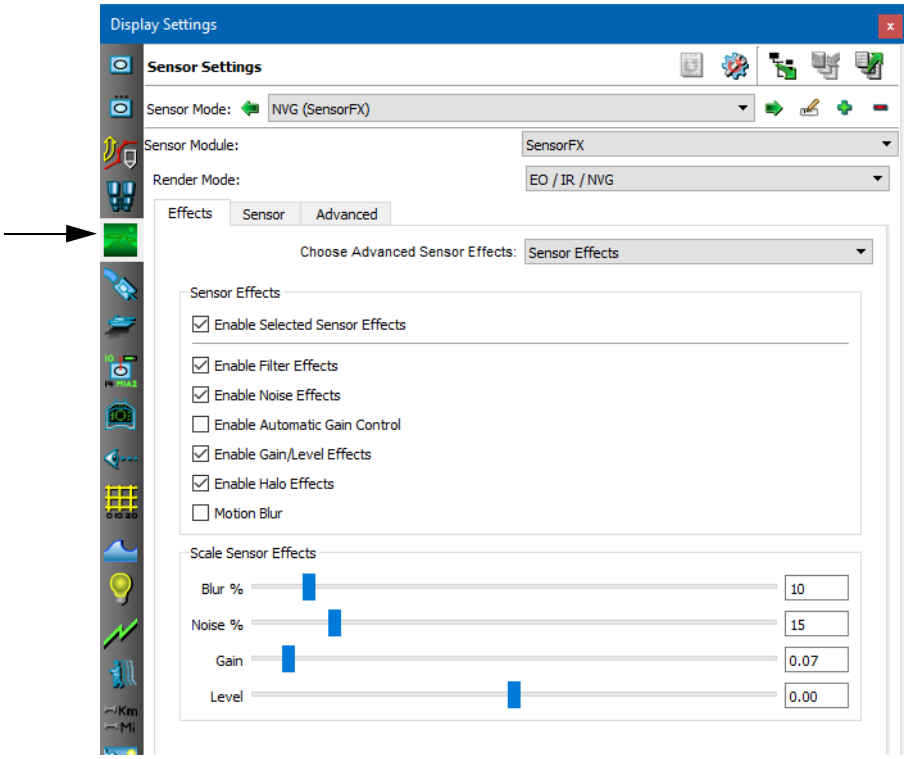

Figure 1-1. Sensor Settings page

3. Select a sensor mode that is similar to the one you want to create.
4. On the Sensor Mode line, click the Add button (✚). A new sensor called sensor_*number* is added. It copies the settings of the sensor you selected. (To rename the sensor, click the Rename button (✎) and enter a new name.)
5. Change the settings for the new sensor as desired.

1.5. Configuring SensorFX Sensors

The SensorFX sensors are configured on the Display Settings dialog box, Sensor Settings page. Configuration options are organized into tabs as follows:

- Effects.
  - Sensor Effects. Table 1-1 lists options for IR (MWIR and LWIR) and NVG.
  - Optics (Table 1-2)
  - Detector (Table 1-3)
  - Electronics (Table 1-4).
- Sensor (Table 1-5).
- Advanced (Table 1-6).

Table 1-1: Effects tab, Sensor Effects

Option	Description
Enable Selected Sensor Effects	Enables or disables the selected options on this page.
Enable Filter Effects	Enables filtering.
Enable Noise Effects	Enables noise.
Enable Automatic Gain Control	SensorFX automatically adjusts the scene to provide the best contrast and brightness given time-of-day, weather, and other scene parameters. It overrides the Gain and Level settings.
Enable Gain/Level Effects	Enables the settings in the Scale Sensor Effect group box.
Enable Halo Effects	Show halos around bright points of light such as head lights and street lamps. (NVG Only)
Enable Blackhot	Changes IR from white hot to blackhot. (IR Only)
Motion Blur	Enables motion blur effects.
Blur	Affects the focus of the scene.
Noise	Affects the graininess of the scene.
Gain	Affects the contrast in the scene.
Level	Affects the brightness of the scene.

Table 1-2: Effects tab, Optics Effects

Option	Description
Aperture Shape	Choose elliptical or rectangular.
Aperture Aspect Ratio	Ratio between horizontal and vertical Instantaneous Field of View (IFOV). Changes are most noticeable when blur is high.
Aperture Diameter	Specified in millimeters. It is used in diffraction calculations. Lower values produce more diffraction spreading (blur).
Blur Spot Diameter	This quantity [mrad] is a measure of the amount of spherical aberration, and thus the focusing power of the lens. The more this number varies from the focal point the more blur in the scene.
F-Number	Non-editable field. It is calculated based on the FOV and aperture diameter.
Focal Length	Non-editable field. It is calculated based on the FOV and aperture parameters. Units are cm.
Halo Threshold	The point at which halo effects are used. A halo is shown around bright points of light such as vehicle headlights or street lights.
Halo Radius	Determines the overall maximum size of the halo.
Halo Intensity	Determines the maximum brightness of the halo.

Table 1-3: Effects tab, Detector Effects (Sheet 1 of 2)

Option	Description
Dwell Time	Dwell time is the time [μsec] over which the post-detector electronics integrates the detector output voltage. A value of zero denotes no integration. Increasing dwell time results in higher signal-to-noise ratio.
Temporal NEdR	This is the “noise-equivalent delta-temperature”. Any real IR sensor will produce self-noise, which adds a constant to the effective apparent temperature of the scene, and may be generated from a variety of effects. Noise converted to a DeltaT. If 0 then no noise impacting the temperature.
% Fixed Pattern	Unwanted signal component that is usually constant or very slowly changing with time. May vary spatially and so could potentially be confused with true fringes. Examples of fixed pattern noise are the pixel to pixel gain variations cosmetic defects in CCD detectors parasitic interference caused by unwanted reflections from various air/glass surfaces.

Table 1-3: Effects tab, Detector Effects (Sheet 2 of 2)

Option	Description
% Poisson	The Percent Poisson Noise is the statistical fluctuations in charge carrier number. It is also known as "shot" noise. Percent Poisson Noise is more important for optical wavelengths than infrared because in the short wavelengths the energy per photon is higher and so there are fewer photons per unit power of illumination.
% 1/F	Also known as "flicker" or "pink noise", this occurs in any biased detector and is a thermo-mechanical effect related to material inhomogeneity or transistor recombination.
Horizontal Detector Pitch	The horizontal distance between detector centers [mm], used in determining focal length.
Vertical Detector Pitch	The vertical distance between detector centers [mm], used in determining focal length.
Background Temperature	The effective temperature [degK] of the ambient background. Can usually be set to nominal air or ground temperature.
Detector Pixel Fill	Term of measurement of FPA performance, which measures how much of the total FPA is sensitive to IR energy. Because the FPA is made of numerous individual detector cells, the total amount of sensitivity is measured by the pathways used to separate the cells and transmit signals. The higher the fill factor, the higher the ratio of sensitivity.
FPA Operability	This indicates the effectiveness of the detector and 100% would be fully functional and as the operability percentage drops the image quality drops.
Noise Frequency Exponent	[Also known as "beta", this is the exponent of the 1/f noise power spectrum.
Noise Frequency Knee	The frequency above which 1/f noise is overshadowed by thermal or statistical noise from other processes.

Table 1-4: Effects tab, Electronics Effects

Option	Description
Frame Rate	The frame rate parameter is used to determine the change in temporal noise per frame.
Pre-Amp Cuton Frequency	Frequency at which amplification of the signal occurs after conversion to a voltage. Good values to use for this parameter are 1 – 10.
Post-Amp Cutoff Frequency	Frequency at which amplification of the signal ceases. Good values to use for this parameter are 50 – 500.
Manual Gain	This is the manual gain value and also the maximum gain used by the automatic gain control. The Effects section will scale the gain from 0 – 100% of this value.
Manual Level	This is the manual level and is added to the value from the Effects section, so if you are using a slider for that, then this would be set to zero. This could probably be left out of a GUI if the slider was being used.
Max AGC Gain Min AGC Gain	Maximum and minimum values for automatic gain control.

Table 1-5: Sensor tab

Option	Description
Lower Band	Lowest wavelength of band in microns.
Upper Band	Highest wavelength of band in microns.
Min Temp	If Automatic Radiance Scaling is not set, then this determines the range used to encode the response textures generated from the mcm textures (for IR).
Max Temp	If Automatic Radiance Scaling is not set, then this determines the range used to encode the response textures generated from the mcm textures (for IR).
Max Light Level	Sets the maximum light level [uW/cm2] which the NVG sensor can detect without saturating. (Visual sensor and NVG only)
Light Pool Gain	(NVG only) Adjusts the brightness of light pools under street lights.
Intensity Modulation Gain	Together with the geometry-model-specific IMOD files, this governs the extent to which detail variance present in the RGB texture for the entity or terrain models is translated into additional variance in the NVG scene. Typical value: 0 – 1.00.

Table 1-6: SensorFX Advanced tab

Option	Description
Non-uniformity On	The non-uniformity feature applies a gain and/or an offset to each pixel at the end of the sensor effects processing.
Non-uniform Gain File	Specifies the gain to apply.
Non-uniform Offset File	Specifies the offset to apply.
Min Gain Max Gain Min Offset Max Offset	The Min/Max Gain and Min/Max Offset parameters can be adjusted to increase/decrease the amount of variation.
Scanning Direction	The direction of the Sensor scanning direction.
Scanline Spacing	The Scanline Spacing can be changed to make it more blocky.
Color Mapping On	A monochrome format with value interpolated from user-defined LUT scaling curve.
Color Mapping Type	1) User-Defined ColorMap: A monochrome format with value interpolated from user-defined LUT scaling curve. 2) LADAR ColorMap: A format designed to emphasize blue at low input values, green in mid-range values, and red at high values. 3) NVG ColorMap: A format characterized by low blue and red channels (until saturation, when they switch to high values), and a constantly-increasing scaled green channel. 4) LogScale ColorMap: A monochrome format based on a logarithmic scaling of the input.
User Defined File	A user-defined color mapping file.
Pre-Aperture	If selected, record the image before applying sensor effects.
Image(s) File Path	Location to store recorded images.

1.5.1. Using Active Sensor Regions (Dynamic Hot Spots)

SensorFX supports active sensor regions. An active sensor region changes its sensor effect as part of a model, such as an engine or gun barrel heats up or cools down. Models that support active sensor regions have a *.ar* file. The following list is a partial list of models have active sensor regions:

- SH-60 Sea Hawk helicopter
- USS Harry S. Truman
- T-72 main battle tank
- DB_AH-64D_LONGBOW_ARMED_GREEN_V7.0
- DB_Mi-24D_ARMED_CAMO_V7.0_RUSSIA
- DB_UH-60L_DESERT_CAMO_V7.0
- Mi-24D_HIND_ARMED_CAMO_V7.0_RUSSIA
- MI-28_Havoc
- UH-60L_BLACK_HAWK_DESERT_CAMO_V7.0
- Y345-Tug Boat
- Leclerc-FR
- M113
- T-80UM1
- ZSU-23-4
- BMW320DBlue
- ChevroletTahoeSilver
- fire_engine_LOD1
- Ford_Contour_Sedan
- GMCYukonGold
- JeepCommanderWhite
- LandRoverDiscoveryWhite
- M-923_5TONS_TRUCK_DESERT_V7.0
- M1025_M2_HMMWV_DESERT_V7.0
- MercedesMSilver
- MI-28
- MiniCooperBlue
- police_car_LOD1
- VBL
- VolksWagenGolfWhite.

When using a DIS or HLA connection, these models activate a dynamic region when the power_plant_status appearance bit is set. Additionally, the T72's tracks grows hotter when the tank is moving and the gun barrel heats up as the result of firing its main gun.

Sensor effects that happen over time require the Ephemeris model (clock) to be active.

If you are using CIGI, you can activate dynamic sensor regions using a Component Control or Short Component Control message. Active sensor regions are named (the names can be found in *.ar* files in the data directories for the models included). CIGI Mappings are provided to map integer values to the names of sensor regions found in the example models. The Component Control message contains a Component class of Entity(0); the instance ID is the ID of the entity to change; the Component State is Sensor State(6), and the component data is a long value of '0' or '1'. '1' turns an active region on; '0' turns it off. The gun barrel on the T72 is an exception. It is impulsed-based. You should send a component data value of '1' each time the gun is fired. '0' is not used.

1.5.2. Specifying when Models Get Displayed

By default, when you are in a sensor mode, such as NVG, simulation object models do not get displayed until SensorFX has processed the materials for the model to determine how to display it. This may cause a delay of several seconds before the model is displayed. If you want models to get shown immediately, you can change this behavior with the SENSORFX_DISABLE_MODEL_DISPLAY_DELAY environment variable. To have models displayed immediately, set this variable to 1. To wait until the materials have been processed, set it to 0. If models are shown before their materials have been processed, they will usually be all black until processing is complete.

1.6. Configuring the Environment

Sensors are sensitive to environmental factors such as lighting and temperature.

To change the environment settings:

1. Choose **Settings** → **Scene**. The Scene Settings dialog box opens.
2. Select the SensorFX Environment Settings page (Figure 1-2).

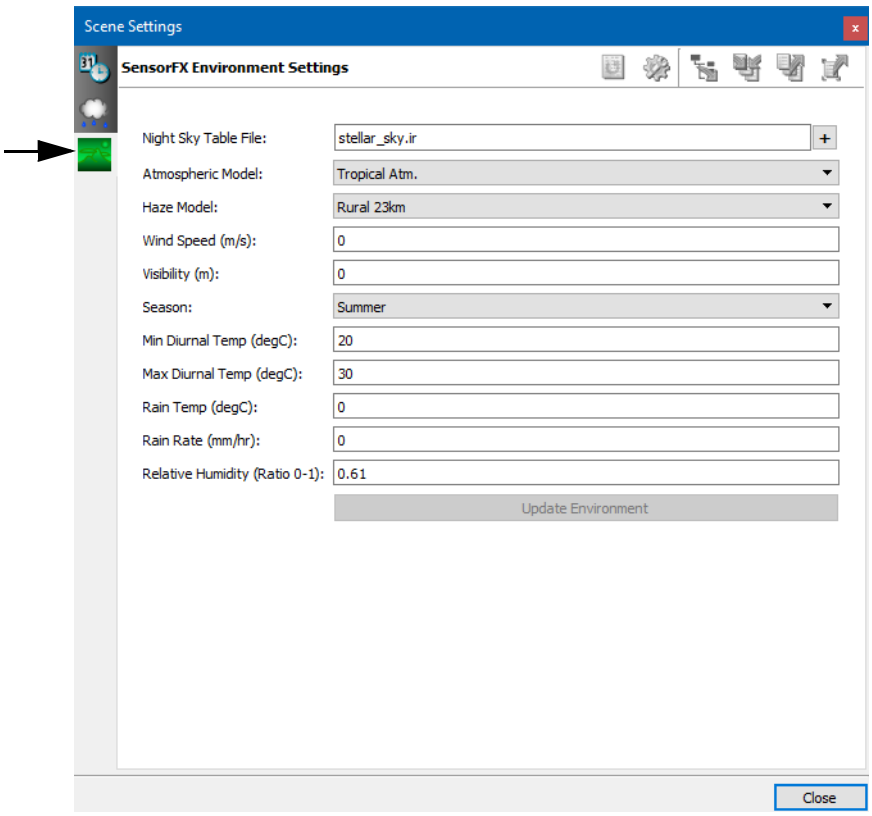

Figure 1-2. SensorFX Environment Settings page

3. Change settings as desired. Table 1-7 describes these options.

Table 1-7: SensorFX Environment tab (Sheet 1 of 2)

Option	Description
Night Sky Table File	Contains planetary and stellar irradiance info.
Atmospheric Model	An atmospheric model is a mathematical model constructed around the full set of primitive dynamical equations which govern atmospheric motions. It can supplement these equations with parameterizations for turbulent diffusion, radiation, moist processes (clouds and precipitation), heat exchange, soil, vegetation, surface water, the kinematic effects of terrain, and convection. Select an atmospheric model from the drop-down menu. Select the model that best describes the simulated scene. User Specified: You can define custom Atmospheric models and identify the model by name in the "User Specified Atmosphere" field.
Haze Model	The amount of haze in the atmosphere. Select an option from the list.
Wind Speed	This is used in the thermal temperature calculations and would likely be constant for a short scenario.
Visibility	This overrides the Modtran visibility when not using a user defined atmosphere file. The units are meters. A setting of 0.0 instructs Modtran to use the default visibility for the chosen atmosphere, and is a recommended setting.
Season	Sets the Season used by the Modtran environment calculations. Choose Summer, Winter, or AutoSelect to use the date.
Min Diurnal Temp Max Diurnal Temp	These specify the temperature range of the air through out a diurnal cycle and would not need to change during a short run scenario, but is important to set correctly even for such scenarios.
Rain Temp [degC]	This setting affects objects in the scene, not the actual temperature of rain. Increasing the temperature leads to a brighter scene.

Table 1-7: SensorFX Environment tab (Sheet 2 of 2)

Option	Description
Rain Rate [mm/hr]	The precipitation rate per hour. Increasing the rain rate reduces the contrast in the scene.
Relative Humidity	The amount of water vapor in a mixture of air and water vapor. It is defined as the partial pressure of water vapor in the air-water mixture, given as a percentage of the saturated vapor pressure under those conditions. The relative humidity of air thus changes not only with respect to the absolute humidity (moisture content) but also temperature and pressure, upon which the saturated vapor pressure depends. Relative humidity is often used instead of absolute humidity in situations where the rate of water evaporation is important, as it takes into account the variation in saturated vapor pressure.

4. Click Update Environment. It can take 30 seconds or longer for the changes to affect the scene.

1.7. Configuring Light Points

In addition to the light point configuration described in Section 35.4, "Configuring Light Points," in *VR-Vantage Users Guide*, you can configure the light points for SensorFX. Once a light point has been assigned a light point type from the *Lights_DefaultValues_3_2.csv* file, it will have a specific bulb type assigned to it. This is the last parameter in the CSV file for a particular light point type. The default is 1, but SensorFX comes with 12 bulb types preconfigured for use (in *./data/sensorFX/atm/mm_lsources.dat*). Here are general descriptions of how the 12 types behave in IR and NVG.

Table 1-8: Preconfigured bulb types

Name in mm_lsources.dat	Bulb Type Index	IR behavior	NVG behavior
Mercury_Pure	1	Air temperature	Bright
HPSodium	2	Air temperature	Bright
Mercury_Vap	3	Warmer than air	Bright
ECE_H7	4	Warmer than air	Dimmer than 3
500T14/8	5	Very Hot	Bright
500T14/8	6	Very Hot	Bright
500T14/8	7	Very Hot	Dimmer than 6
500T14/8	8	Very Hot	About the same as 7

Table 1-8: Preconfigured bulb types

Name in mm_lsources.dat	Bulb Type Index	IR behavior	NVG behavior
500T14/8	9	Very Hot	About the same as 8
500T14/8	10	Very Hot	Dimmer than 9
500T14/8	11	Very Hot	About the same as 10
500T14/8	12	Very Hot	Dimmer than 11

In addition to the preconfigured types specified in Table 1-8, an advanced use case would be to add new types to *mm_lsources.dat*. Each line represents a light bulb type, with tab separated parameters starting with the name on the left. Here are descriptions of the parameters in the order that they are specified in the line.

Table 1-9: Bulb parameters

Parameter Name	Parameter Description
ID	String name of the light bulb type.
posN[m]	Unused by SensorFX.
posE[m]	Unused by SensorFX.
posD[m]	Unused by SensorFX.
dirN[unitless]	Unused by SensorFX.
dirE[unitless]	Unused by SensorFX.
dirD[unitless]	Unused by SensorFX.
T[degK]	Floating point value that is the effective black-body temperature of filament in degrees Kelvin.
P[W]	Floating point value that is the source total spectrally-integrated power.
etha[unitless]	Floating point value between 0 and 1 that is the source efficiency (that is, fraction of input power P converted into EO-band light field).
cutoff[deg]	Unused by SensorFX.
area[cm2]	Floating point value that is the effective area of light source (bulb). If < =0 then the default value of 400cm2 is used.
thickness[cm]	Floating point value that is the thickness of bulb glass or other material through which excess heat is conducted.

1.8. Using SensorFX with Remote Display Engines

You can use SensorFX with remote display engines.

È To run a remote display engine that uses SensorFX, on the Start menu, choose **MAK Technologies → VR-Vantage Display Engine with SensorFX**

1.8.1. Changing the Sensor on a Remote Display Engine

Ordinarily, to change the sensor being used, you change to an observer that has the sensor you want. However, you cannot change the observer on remote display engines. If you want to change the sensor on a remote display engine, you can do so by changing the sensor for the remote display engine's channel. (You can change the sensor for any channel this way, not just remote display engines.)

To change the sensor for a remote display engine:

1. Choose **View → Display Engine Configuration Editor Panel**. The Display Engine Configuration Editor opens.
2. Select the channel for the remote display engine. By default, the Sensor attribute has the value From Observer (Figure 1-3).

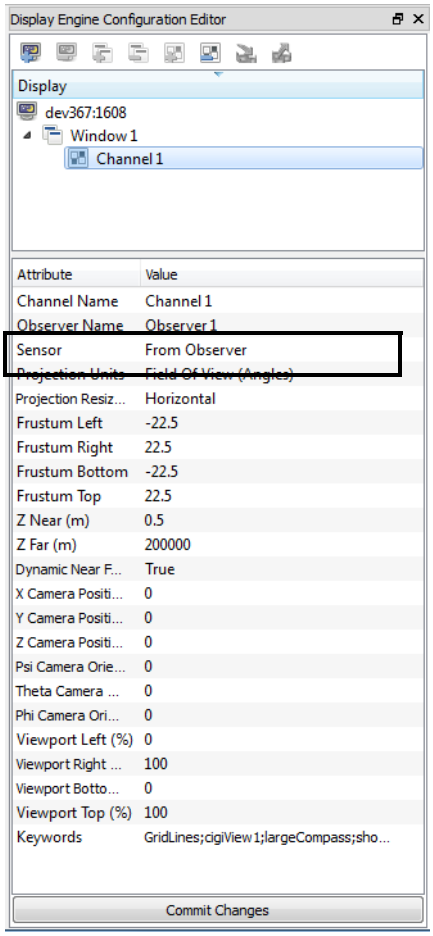

Figure 1-3. Display Engine Configuration Editor

3. Click the value to make it editable.
4. Select a sensor from the list (Figure 1-4).

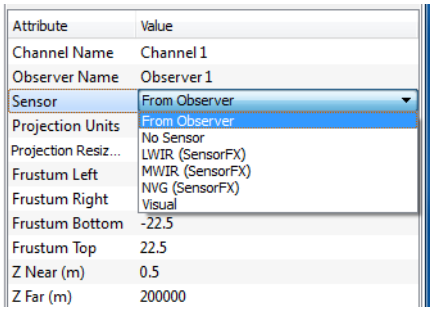

Figure 1-4. Sensor list

5. Click Commit Changes. The sensor in the remote display engine will change.

1.9. Material Classification on Demand

SensorFX material classification on demand (MCOD) provides automatic material classification using vector shape-files, and combines them with an intensity-modulated image to produce material encoded textures used in the real-time sensor rendering. The material encoded textures use JRM's Eigen material technique, which provides excellent rendering capabilities with the ability to mimic an unlimited number of material systems through the combination of multiple eigen materials.

Eigen materials can be thought of as a basic material group that can be combined to form specific material systems. These textures are in a continuous parameter space that is not dependent on time-of-day and other environmental conditions and can be Mipped/interpolated and compressed. The textures can be used to simulate all the different sensor bands (NVG, IR, and so on) without being regenerated.

The process is done automatically as the data is paged in from the server, and once the material textures have been generated, the texture array images are saved to the standard osgEarth file cache ready for fast loading on subsequent runs.

This section explains how to define generation of a material classified texture in an earth file.

The MCOD process:

- Processes any number of vector shape-files as raster images for each LOD. These shape-files represent material features (for example, roads, building footprints, water, and so on.)
- Combines the image textures to produce a per-pixel intensity modulation that maintains the spatial variation.
- Generates the material classified texture layers that are processed in real-time to calculate sensor wavelength-specific output using JRM's sensor suite.

MCOD:

- Supports an unlimited number of material types by the use of texture arrays. Individual Eigen materials are stored as distinct components in each of the texture array layers.
- Stores individual materials as distinct components in each of the texture array layers.
- Specifies material components using osgEarth earth files. A feature component's color is specified using the standard style symbol set. This section describes how to use the styles necessary to define features plus any custom styles particular to MCOD.
- Caches files using the standard osgEarth pipeline to accelerate run-time load performance.

The MCOD plug-in uses standard osgEarth terminology plus the following terms:

- texture layer. Corresponds to an index of an output texture array (0 to number of available layers-1).
- channel. An image color component (for example, R,G,B or A).
- default material. The material to use in the areas that are not defined by a shape-file.
- Intensity Modulation (IM). An optional channel to hold a gray scale representation of the visual imagery.
- feature level. Extends the osgEarth concept of level, but is particular to features, such as shapefiles.

The following code is a template for setting up an MCOD layer in an earth file:

```
<map name="Name of the Map" type="geocentric" version="2">
  <!-- [Optional] Visible Image layers go here -->

  <!-- [Optional] Elevation source can be specified here -->
  <image name="MaterialClassificationOnDemand" driver="mcod"
  visible="false" shared="true">
    <!-- optional file cache id -->
    <cacheid>mcod</cacheid>
    <!-- defines the depth of the texture array, must be 1 or more -->
    <num_texture_layers>2</num_texture_layers>

    <!-- for now texture compression MUST be set to none -->
    <texture_compression>none</texture_compression>
    <!-- assign default material channel (RGBA) and texture layer -->
    <default_material_channel>#0000ff00</default_material_channel>
    <default_material_texture_layer>0</default_material_texture_layer>

    <!-- assign intensity modulation color channel (RGBA) and
      texture layer -->

  <intensity_modulation_channel>#000000ff</intensity_modulation_channel

  <intensity_modulation_texture_layer>0</intensity_modulation_texture_la
  yer>

    <!-- specify minimum level where features (shapefiles) will begin
      to render -->
    <min_feature_level>10</min_feature_level>
    <!-- [Optional] image layers - to compute intensity modulation -->
    <image>
      ...
    </image>
    <image>
      ...
    </image>
    <!-- features layers - contain material shapefiles to render -->
     <features>
      ...
    </features>
    <features>
      ...
    </features>
    <!-- define rendering styles, by name, for the features -->
```

```
      <styles>
        <style type="text/css">
          ...
        </style>
      </styles>
    </image>
  </map>
```

Layer flags (osgEarth standard):

- driver="mcod". Specifies the MCOD plug-in.
- visible="false". Layer will not be rendered directly.
- shared="true". Textures are made available to shaders.

MCOD specific variables:

- num_texture_layers. The number of texture layers defining the creation of the output texture array.
- default_material_channel. Channel (color component, RGBA order) that will contain the default material. in the example above, default material channel is defined as #0000ff00 indicating that the 8-bit Blue channel will contain the default material.
- default_material_texture_layer. Texture array id [0 to num_texture_layers-1] for the default material.
- intensity_modulation_channel. Channel (color component, RGBA order) that will contain intensity modulation. In the example above, the intensity modulation channel is defined as #000000ff, indicating that the 8-bit Alpha channel will contain IM.
- intensity_modulation_texture_layer. Texture array id [0 to num_texture_layers-1] for intensity modulation.
- min_feature_level. Similar to the osgEarth variable min_level, but is specific to features (material shapefiles). Default: 0.
- splat_combine_mode. Specifies how the procedural imagery materials should be combined with the MCOD layer's materials and what to use as the intensity modulation. This can be one of the following options:
  - NO_COMBINE. Default value for splat_combine_mode. Use this mode if there is not a procedural imagery/splatting layer that can be combined with the MCOD layer.
  - IM_SPLAT. Uses the visual procedural image layer as the intensity modulation for the SensorFX sensor modes. This mode requires that both the procedural imagery material layer and the MCOD layer use the same *.ms* file. See *./userData/servers/MAK Earth - MCOD - Splat - Im Splat (online).earth* for an example of how to configure this mode.
  - IM_IMAGE_LAYER. Uses an image layer as the intensity modulation for the SensorFX sensor modes. This mode can use two separate *.ms* files, one for the procedural imagery material layer and one for the MCOD layer. See *./userData/servers/MAK Earth - MCOD - Splat - Im Image Layer (online).earth* for an example of how to configure this mode.

The following code shows the MCOD setup in *MAK Earth - MCOD (online).earth*:

```
<image name="features"
         driver="mcod"
         opacity="1.0"
         visible="false"
         shared="true"
         min_filter     = "LINEAR_MIPMAP_LINEAR"
         mag_filter     = "LINEAR"
         max_range      = "100000000">
  <cacheid>mcod</cacheid>

  <!-- specify the minimum level that features (shapefiles) will begin to
  render -->
  <min_feature_level>10</min_feature_level>

  <num_texture_layers>1</num_texture_layers>

  <texture_compression>none</texture_compression>
  <!-- assign default material channel (RGBA) and texture layer -->
  <default_material_channel>#0000ff00</default_material_channel>
  <default_material_texture_layer>0</default_material_texture_layer>

  <!-- assign intensity modulation color channel (RGBA) and texture layer
  -->
  <intensity_modulation_channel>#000000ff</intensity_modulation_channel
  <intensity_modulation_texture_layer>0</intensity_modulation_texture_layer
  >
```

1.9.1. Associating Styles and Features

Styles specify how features are rasterized into a layer. The feature rasterization is implemented using the osgEarth framework that wraps the AggLite renderer. Styles defined in the earth file control raster characteristics such as color and stroke width.

MCOD features must be associated with style. (Please see osgEarth documentation for information on setting up feature styling.) In addition to the osgEarth styling support, MCOD also supports a simple styling configuration by matching the feature name to the style name.

The following code is from *MAK Earth - MCOD (online).earth*:

```
<features name="buildings-1" driver="wfs">
  <url>http://vr-theworld.com/vr-theworld/features/wfs</url>
  <build_spatial_index>true</build_spatial_index>
  <typename>65</typename>
</features>

<features name="roads-1" driver="wfs">
  <url>http://vr-theworld.com/vr-theworld/features/wfs</url>
  <build_spatial_index>true</build_spatial_index>
  <typename>55</typename>
</features>

<styles>
  <style type="text/css">
    roads-1 {
      stroke:             #ff0000;
      stroke-width:       50m;
      stroke-min-pixels:  0;
      altitude-clamping:  terrain-drape;
      texture-layer:      0;
    }
    buildings-1 {
      fill:          #00ff00;
      stroke:   #00ff00;
      stroke-min-pixels:  0;
      stroke-width:  0px;
      texture-layer:      0;
    }
  </style>
</styles>
```

This example has the MCOD specific style variable texture-layer, a texture array id [0 to num_texture_layers-1] for the feature.

1.10. Streaming Sensorized Imagery

SensorFX supports streaming imagery. It is set up through the earth file. Create an image section with a name that contains emat. For example:

```
<image name="streaming emat2" driver="gdal" lod_blending="true"
  shared="true" visible="false">
  <shared_sampler>jrmuSensorTex2</shared_sampler>
  <shared_matrix>jrmuSensorTexMatrix2</shared_matrix>
  <url>../../Data/Terrain/DATABASENAME/imagery/emat2</url>
</image>

<image name="streaming emat3" driver="gdal"  lod_blending="true"
  shared="true" visible="false">
  <shared_sampler>jrmuSensorTex3</shared_sampler>
  <shared_matrix>jrmuSensorTexMatrix3</shared_matrix>
  <url>../../Data/Terrain/DATABASENAME/imagery/emat3</url>
</image>
```

Emat2 and emat3 should be in separate directories. They need to be geoTiff versions of the emat textures so that osgEarth has the location information to correctly place the textures.

To specify an ms file, do the following:

```
<external>
  <msfile>../../Data/Terrain/DATABASENAME/databaseEmat.ms</msfile>
</external>
```

This ms file can be used with streaming imagery, MCOD, or osgEarth terrains that are not material classified at all.

For information about ms files and emat files, please see Chapter 2, *Adding GenesisMC Models to SensorFX*.

The example HawaiiTifExample has a sample earth file called *HawaiiTifExample.earth* and an *.mtf* file called *HawaiiTifExample.mtf*. It shows how to set up the earth file and the terrain directory for using geotiff's directly.

1.11. Using SensorFX with Geocentric Terrains

If you load a geocentric terrain without using an earth file, SensorFX does not work properly. You can work around this problem by creating an earth file that does not do anything and creating an MTF file that loads the earth file and the terrain.

The earth file should look like this:

```
-->
<map name="MAK Earth" type="geocentric" version="2">
<options>
<terrain color = "#00000000" />
</options>
</map>
```

1. In SensorFX, load the earth file as a terrain server.
2. Load the geocentric terrain as a terrain patch.
3. Save the terrain as an MTF terrain.

By default, the observer will be in space, so create a saved view of the terrain at a useful location.

2. Adding GenesisMC Models to SensorFX

This chapter explains how to integrate materially classified models generated by GenesisMC into SensorFX.

2.1. Introduction to Adding Models Processed by GenesisMC

GenesisMC is an application created by JRM Technologies that processes models and creates the files needed by SensorFX to render them based on material classification. This chapter assumes you have generated material classified versions of your models using GenesisMC and explains how to set up the files for SensorFX to use them.

2.2. Model File Setup

SensorFX uses the following types of files to load the materially classified version of a model:

- Material classification file (*.emat*).
- Material system file (*.ms*).
- Intensity modulation file (*.imod*) (optional).
- Active regions file (*.ar*) (optional).

The *.emat*, *.ms*, and *.imod* files are generated by GenesisMC. The *.ar* file is created by the model developer or someone who understands the location of hotspots on the model.

SensorFX uses the names of the textures and models to look for the material classified configuration files. For example, the base set of files for the SH-60 SEA-HAWK (in *./data/Vehicles/RotaryWing/SH-60 SEA-HAWK*) before material classification is:

- DB_ROTOR_4PALES_128_rgba.meif
- DB_SH60L_1024_DESERT_CAMO_NML_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_RFL_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_SPC_rgb.meif
- DB_SH60L_1024_rgb.meif
- DB_SH-60L_DESERT_CAMO_V7.0.medf
- DB_SH60L_FLOOR_256_rgb.meif
- DB_SH60L_INTERIOR_512_rgb.meif
- DB_SH60L_SHADOW_256_rgba.meif
- DB_SH60L_WINDOW_512_RFL_rgba.meif
- DB_SH60L_WINDOW_512_rgba.meif
- DB_SH60L_WINDOW_512_SPC_rgba.meif

2.2.1. The *.emat* Files

EMAT files are materially classified versions of the visual RGB textures. One of the RGB textures for the SH-60 is DB_SH60L_1024_rgb.meif. After processing the textures, GenesisMC creates three EMAT textures for *DB_SH60L_1024_rgb.meif – DB_SH60L_1024.emat1.dds*, *DB_SH60L_1024.emat2.dds*, and *DB_SH60L_1024.emat3.dds*. SensorFX does not use the emat1 file. In order to save texture memory, SensorFX uses the *.imod* file to fulfill the same purpose.

Copy the **.emat2.dds* and **.emat3.dds* files to the same directory as the RGB textures so that SensorFX can find them. The complete list of *.emat* files for the SH-60 is as follows:

- DB_ROTOR_4PALES_128.emat2.dds
- DB_ROTOR_4PALES_128.emat3.dds
- DB_SH60L_1024.emat2.dds
- DB_SH60L_1024.emat3.dds
- DB_SH60L_1024_DESERT_CAMO.emat2.dds
- DB_SH60L_1024_DESERT_CAMO.emat3.dds
- DB_SH60L_FLOOR_256.emat2.dds
- DB_SH60L_FLOOR_256.emat3.dds
- DB_SH60L_INTERIOR_512.emat2.dds
- DB_SH60L_INTERIOR_512.emat3.dds
- DB_SH60L_WINDOW_512.emat2.dds
- DB_SH60L_WINDOW_512.emat3.dds.

2.2.2. Material System File (*.ms*)

SensorFX uses the material system file to map a channel of the emat texture to a particular material. There might be multiple material system files in the GenesisMC output directory. The one that SensorFX needs ends in *Emat.ms*. The contents of the file will be nine lines that define six materials. In the SH-60 example, this file is *DB_SH-60L_DESERT_CAMO_V7.0Emat.ms*.

There are two ways to point to a material system file. The simplest way is to give the file the same name as the model file name minus the extension and add *Emat.ms*. This file must be in the same directory as the model file. In the SH-60 example, the model filename is *DB_SH-60L_DESERT_CAMO_V7.0.medf*. Therefore, the material system file is *DB_SH-60L_DESERT_-CAMO_V7.0Emat.ms*.

The other way to point to a material system file is in a model definition. For details, please see "The msFilename Parameter," on page 2-4.

2.2.3. Intensity Modulation File (*.imod*)

An intensity modulation file or *.imod* file is an optional file that defines how much of the RGB texture to mix with the Sensor response in the final result. It allows the intensity modulation to be defined on a per-material and per-sensor band level. The file must have the same name as the material system file, except that the extension must be *.imod* instead of *.ms*. This file must be in the same directory as the material system file. In the SH-60 example the material system file is *DB_SH-60L_DESERT_CAMO_V7.0Emat.ms* which corresponds to intensity modulation file *DB_SH-60L_DESERT_CAMO_V7.0Emat.imod*.

2.2.4. Active Region File (*.ar*)

The active region file is an optional file that lets you define which materials are dynamic. (Active regions are also called hotspots.) The file defines a material for the off state and another material for the on state. If the model does not have any active regions then it does not need an active region file.

The active region file is a text file that you create. It is not created by GenesisMC.

See *DB_SH-60L_DESERT_CAMO_V7.0Emat.ar* for an example of an active region file. For a more complex example, see *./data/Vehicles/Tracked/T72/T72aEmat.ar*.

The *.ar* file must have the same name as the *.ms* file it corresponds to, except that the extension is changed to *.ar*. For example *DB_SH-60L_DESERT_-CAMO_V7.0Emat.ar* corresponds to *DB_SH-60L_DESERT_-CAMO_V7.0Emat.ms*.

2.3. Model Definitions for Materially Classified Models

If you have generated material classification files for a model that is already configured in VR-Vantage, all you have to do is put the required files in the model directory as described in previous sections. You do not have to edit the model definition.

If you have created a new model, you must create a model definition, as described in Section 32.1, "Creating and Editing Model Definitions," in *VR-Vantage Users Guide*.

SensorFX supports the following optional parameters that you can add to schemas and model definitions: msFilename, powerPlantActiveRegions, movingActiveRegions, and weaponFireActiveRegions.

2.3.1. The msFilename Parameter

The msFilename parameter lets you point to a material system file that does not use the naming convention described in "Material System File (*.ms*)," on page 2-3. Possible reasons for this are if there are two models that need to use the same material system file or there are multiple variations of the same model that need different material systems.

The msFilename parameter is not available by default. You must be added to the model's schema before you can use it. Please see Section 31.5, "Adding a Parameter to a Schema," in *VR-Vantage Users Guide* for information about how to add a parameter to a schema.

The parameter must be called msFilename and have its type set to string.

To determine which schema to edit, look at the schema listed for the model definition for the model you are editing. The most likely schemas are Entity and Watercraft.

2.3.2. The powerPlantActiveRegions Parameter

The power plant active regions are active regions that should be turned on when the model's power plant is turned on and turned off when the model's power plant is turned off.

In the active region file, the first column of each line specifies the name of an active region. The SH-60 has two active regions, EngineGlass and EnginePaint. In order to map them to the power plant active state in VR-Vantage, the model definition needs a parameter added called powerPlantActiveRegions. The value of that parameter is a comma-separated list of active regions that should map to the power plant state. For the SH-60, the value for powerPlantActiveRegions is EngineGlass,EnginePaint. This maps the EngineGlass and EnginePaint active regions to the power plant state.

The default value for the powerPlantActiveRegions parameter is `powerPlant` if the parameter is not specified for a given model definition. If the model definition specifies a powerPlantActiveRegions parameter, then it overrides the default and `powerPlant` is no longer a valid active region unless it is also specified in the parameter value.

2.3.3. The movingActiveRegions Parameter

The moving active regions are active regions that should be turned on when the model is moving and turned off when the model is stationary. *./data/Vehicles/Tracked/T72/T72aEmat.ar* is an example of an active regions file that has a moving active region. The Track active region corresponds to the material on the tracks of the tank. Therefore it must be mapped to the moving active region in the model definition. The movingActiveRegions parameter in the model definition for the T-72 has a value of Track. It maps the Track active region to the moving state of the tank.

The default value for the movingActiveRegions parameter is `moving` if the parameter is not specified for a given model definition. If the model definition specifies a movingActiveRegions parameter, then it overrides the default and `moving` is no longer a valid active region unless it is also specified in the parameter value.

2.3.4. The weaponFireActiveRegions Parameter

The weapon fire active regions are active regions that should be triggered when the model fires its weapon. *./data/Vehicles/Tracked/T72/T72aEmat.ar* is an example of an active regions file that has a weapon fire active region. The Artillery active region corresponds to the material on the weapon of the tank. It must be mapped to the weapon fire active region in the model definition. The **weaponFireActiveRegions** parameter in the model definition for the T-72 it has a value of Artillery. It maps the tank's main large gun active region to the weapon fire event in VR-Vantage.

The default value for the **weaponFireActiveRegions** parameter is `weaponFire` if the parameter is not specified for a given model definition. If the model definition specifies a **weaponFireActiveRegions** parameter then it overrides the default and `weaponFire` is no longer a valid active region unless it is also specified in the parameter value.

2.4. Configuring Particle Systems

Particle systems are configured slightly differently than other models in SensorFX. Instead of using emat textures to define the materials that are used, they use PIP files to define how the particle system should be handled in sensor modes. In order for SensorFX to know that the model is a particle system instead of a regular model, the MS file must end in *.specialeffect.ms* instead of *.ms*. These MS files are searched for by taking the model name, removing the extension and replacing it with *.specialeffect.ms* (for example, *smoke.osg* would look for *smoke.specialeffect.ms*).

Once SensorFX determines that there is a *.specialeffect.ms* file, it looks for the corresponding *.pip* file. First it looks for the same filename as the MS file except without the *.ms* extension (for example, for *smoke.specialeffect.ms* it would look for *smoke.specialeffect.pip* in the same directory as the MS file). If it does not find that PIP file, it looks for a keyword that it can match to a default PIP file. The default PIP files are in *./data/SensorFX/particleEffects*.

Continuing our example, assume the *smoke.osg* model has a *smoke.specialeffect.ms* file. SensorFX looks for *smoke.specialeffect.pip*. If it does not find it, it uses *smoke.pip*, because it matches the smoke keyword in *smoke.specialeffect.ms*.

Table 2-1 lists the default PIP files.

Table 2-1: Default PIP files

Keyword	PIP file
dust	*dust.pip*
smoke	*smoke.pip*
flame	*flame.pip*
plume	*plume.pip*
flare	*flare.pip*

3. Setting Sensor Parameters Using the API

This chapter explains how to set sensor parameters using the VR-Vantage Toolkit.

3.1. Sensor Attribute Names and Types

Table 3-1 lists attribute names and types for settings SensorFx sensor parameters using the setParameter() functions in the *DtMasterSensorController* class. The exampleSensorSettings example project shows how to use these parameters to control the SensorFX settings through the API. These also correspond to the attribute value pairs in *default_Sensors.sfx* for SensorFX.

Table 3-1: Attributes and types for sensor parameters (Sheet 1 of 8)

Attribute	Type	Description
ApertureShape	Integer	Specifies the shape of the sensor lens aperture. Elliptical = 0 Rectangular = 1.
ColorMapType	Integer	16-bit output image format options. Accepted values are: ◆ 1 - User-Defined ColorMap: A monochrome format with value interpolated from user-defined LUT scaling curve. ◆ 2 - Pseudo ColorMap: A format designed to emphasize blue at low input values, green in mid-range values, and red at high values. ◆ 3 - NVG ColorMap: A format characterized by low blue and red channels (until saturation, when they switch to high values), and a constantly-increasing scaled green channel. ◆ 4 - LogScale ColorMap: A monochrome format based on a logarithmic scaling of the input.

Table 3-1: Attributes and types for sensor parameters (Sheet 2 of 8)

Attribute	Type	Description
ModtranAtmModel	Integer	An atmospheric model is a mathematical model constructed around the full set of primitive dynamical equations that govern atmospheric motions. It can supplement these equations with parameterizations for turbulent diffusion, radiation, moist processes (clouds and precipitation), heat exchange, soil, vegetation, surface water, the kinematic effects of terrain, and convection. Accepted values are: ◆ 1 - Tropical ◆ 2 - MidLatitudeSummer ◆ 3 - MidLatitudeWinter ◆ 4 - SubArcticSummer ◆ 5 - SubArcticWinter ◆ 6 - USStandard ◆ 7 - UserDefinedAtm
ModtranHazeModel	Integer	Specifies the Modtran haze model to use for the atmospheric calculations. Available values are: Accepted values are: ◆ -1 - NoAerosol (but might have clouds) ◆ 0 - NoHaze (No aerosol or clouds) ◆ 1 - Rural_23km ◆ 2 - Rural_5km ◆ 3 - NavyMaritime ◆ 4 - Maritime_32km ◆ 5 - Urban_5km ◆ 6 - Tropospheric_50km ◆ 7 - UserDefined (uses the atm state file for definition) ◆ 8 - Fog_200m ◆ 9 - Fog_500m ◆ 10 - Desert
ModtranSeason	Integer	Set the season of the year that should be used by the ModTran environment calculations. ◆ AutoSelect = 0 ◆ Summer = 1 ◆ Winter = 2
RenderMode	Integer	Sets the render mode to use for this sensor. Set to 0 for visual sensors and 2 for EOIR sensors (NVG or IR).

Table 3-1: Attributes and types for sensor parameters (Sheet 3 of 8)

Attribute	Type	Description
ScanMaskDirection	Integer	Sets the direction of the Sensor scanning. ◆ Horizontal = 0 ◆ Vertical = 1 ◆ Full Image Scan = 2.
ScanMaskLineSpacing	Integer	Scanline spacing can be changed to make it blockier. Value between 1 and 512.
ColorMapEnable	Boolean	Enables the color mapping effect which converts the grayscale intensities to colored values.
OptimizedFilterEnable	Boolean	Enables a performance optimized filtering technique for the blur calculations.
ScanMaskUserDefined	Boolean	Enables the application of a scan mask user defined file to the image.
ScanningSensorEnable	Boolean	Enables the scan mask to the image.
agcOn	Boolean	Enables automatic gain control. gainOn must be set to True for this to be fully enabled.
blackHotOn	Boolean	Enables black hot to be applied to the image. When this is False, the image is displayed in white hot.
filterOn	Boolean	Enables blur to be applied to the image.
gainOn	Boolean	Enables gain and level effects to be applied to the image and must be set to True for automatic gain control to be enabled.
halosOn	Boolean	Enables halos to be applied to the scene. This effect is only applied if the sensor mode is NVG.
motionBlurOn	Boolean	Enables motion blur to be applied to the scene.
noiseOn	Boolean	Enables noise to be applied to the scene.
sensorEffectsOn	Boolean	Enables or disables all the sensor effects. If set to False all other sensor effects are disabled.
AGCMaxGain	Float	Sets the maximum gain value that can be used by the automatic gain control.
AGCMinGain	Float	Sets the minimum gain value that can be used by the automatic gain control.
ApertureAspectRatio	Float	Allows for a non-circular/non-square aperture to be specified as a ratio of the width and height.
ApertureWidth	Float	Diameter of the aperture in millimeters.

Table 3-1: Attributes and types for sensor parameters (Sheet 4 of 8)

Attribute	Type	Description
BackgroundTemp	Float	Temperature used to determine radiance of the noise delta in IR = Radiance(Tback + NEdT) – Radiance (Tback) (Degrees Kelvin; typically 293K.)
BlurSpotDiameter	Float	Measure of the amount of spherical aberration and thus the focusing power of the lens. (millirads)
DwellTime	Float	Time over which the post-detector electronics integrates the detector output voltage. A value of zero (0) denotes no integration. Increasing dwell time results in higher signal-to-noise ratio and thus a higher quality image/scene. Dwell time is measured in microseconds. This also affects the motion blur as the higher the dwell time, the more that previous frames affect the current frame.
FInverseNoisePercent	Float	The Percent 1/f Noise, also known as “flicker” or “pink noise” occurs in any biased detector and is a thermo-mechanical effect related to material inhomogeneity or transistor recombination. For more information about calculating noise, please see “Calculating Noise,” on page 3-9.
FilterAdjustment	Float	Specifies the amount of blur to apply to the image. A value of 1.0 will use the rest of the blur configuration settings directly. Values between 0.0 and 1.0 reduce the amount of blur accordingly.
FixedPatternNoisePercent	Float	An unwanted signal component that is usually constant or very slowly changing with time. Examples of Fixed Pattern Noise include pixel to pixel gain variations, cosmetic defects in the CCD detectors, and parasitic interference caused by unwanted reflections from various air or glass surfaces. For more information about calculating noise, please see “Calculating Noise,” on page 3-9.
FrameRate	Float	Frame rate of the sensor. Typically between 30 and 60hz.
GainAdjustment	Float	Sets the percentage of the manual gain value to apply to the image.
HaloIntensity	Float	Represents the maximum brightness of the Halo.

Table 3-1: Attributes and types for sensor parameters (Sheet 5 of 8)

Attribute	Type	Description
HaloRadius	Float	Measured in pixels, and represents the maximum size of the Halo.
HaloThreshold	Float	Determines how bright a light point needs to be before it creates a halo. A higher halo threshold means less haloing and a lower halo threshold means more light points will create halos.
HorizontalFOV	Float	Refers to how far it can look left and right without physically moving, similar to the way you can scan a room with your eyes without moving your head.
HorizontalPitch	Float	The horizontal distance between detector centers, in millimeters. Used in determining focal length.
IntensityModulationGain	Float	Together with the geometry-model-specific IMOD files, this governs the extent to which detail variance present in the RGB texture for the entity or terrain models will be translated into additional variance in the IR scene. Between 0.0 and 1.0
LevelAdjustment	Float	Sets the amount of level to apply to the image in addition to the manual level value.
LightPoolGain	Float	Scales the dynamic lighting intensity for EO and NVG sensor modes.
LowerBand	Float	Lowest wavelength of band in microns.
ManualGain	Float	Sets the amount of gain to apply to the image if the gain adjustment value is set to 1.0.
ManualLevel	Float	Sets the amount of level to apply to the image if the level adjustment value is set to 0.0.
MaxDiurnalTemperature	Float	Specifies the maximum temperature of the air throughout a diurnal cycle and would not need to change during a short run scenario, but is important to set correctly even for such scenarios. (Degrees Celsius)
MaxLightLevel	Float	Sets the maximum light level [uW/cm2] that the EO/NVG sensor can detect without saturating.
MaxTemp	Float	(Upper Temp Limit): If Automatic Radiance Scaling is not set, then this determines the range used to encode the response textures generated from the mcm textures (for IR).

Table 3-1: Attributes and types for sensor parameters (Sheet 6 of 8)

Attribute	Type	Description
MinDiurnalTemperature	Float	Specifies the minimum temperature of the air throughout a diurnal cycle and would not need to change during a short run scenario, but is important to set correctly even for such scenarios. (Degrees Celsius)
MinTemp	Float	(Minimum Temp Limit): If Automatic Radiance Scaling is not set, then this determines the range used to encode the response textures generated from the mcm textures (for IR).
ModtranVisibility	Float	Specifies the visibility, in meters, to use for the Modtran calculations when there is not a user defined atmosphere file specified. If set to 0 then it will use the haze model to determine the visibility.
MtfMotionBlurSize	Float	Sets the motion blur spot size (millirads). Currently this is not implemented as a motion blur but is added to the other MTFs.
NoiseAdjustment	Float	Specifies the amount of noise to apply to the image. Between 0.0 and 1.0.
NoiseEquivalentRadiance	Float	The Noise-Equivalent delta Radiance is applied when using an NVG/EO sensor. NEdR is another means of specifying the noise level. It takes into account the effect of external factors on overall uncertainty in output levels. This quantity is used instead of NEdT when outside of the IR regime. Typically between 0.00003 and 0.0003
NoiseEquivalentTemp	Float	The Noise-Equivalent delta Temperature. All IR sensors will produce self –noise, which will add a constant to the effective apparent temperature of the scene and may be generated from a variety of effects.
NoiseFrequencyExp	Float	Known as “beta” is the exponent of the 1/f noise power spectrum.
NoiseFrequencyKnee	Float	The frequency above which 1/f noise is overshadowed by thermal or statistical noise from other processes. (Hz)
PercentDeadPixel	Float	Specifies the percentage of the total number of pixels that are dead.
PercentPixelFill	Float	Between 100 and 0 with lower values increasing the amount of blur.

Table 3-1: Attributes and types for sensor parameters (Sheet 7 of 8)

Attribute	Type	Description
PoissanNoisePercent	Float	The Percent Poisson Noise is the statistical fluctuation in charge carrier number. This is also referred to as “shot” noise. For more information about calculating noise, please see “Calculating Noise,” on page 3-9.
PostCutoffFreq	Float	The frequency at which amplification of the signal ceases. (Hz)
PreCutonFreq	Float	The frequency at which amplification of the signal occurs after conversion to a voltage. (Hz)
RainRate	Float	Specifies the rate of precipitation, in mm/hr.
RainTemp	Float	Specifies the temperature of the precipitation. (Degrees Celsius)
RelativeHumidity	Float	The amount of water vapor in a mixture of air and water vapor. It is defined as the partial pressure of water vapor in the air-water mixture, given as a percentage of the saturated vapor pressure under those conditions. The relative humidity of air thus changes not only with respect to the absolute humidity (moisture content), but also temperature and pressure, upon which the saturated vapor pressure depends. Relative humidity is often used instead of absolute humidity in situations where the rate of water evaporation is important, as it takes into account the variation in saturated vapor pressure. Floating point value between 0.0 and 1.0.
ScanMaskMaxGain	Float	Sets the maximum gain to apply to the image based on the scan mask.
ScanMaskMaxLevel	Float	Sets the maximum level to apply to the image based on the scan mask.
ScanMaskMinGain	Float	Sets the minimum gain to apply to the image based on the scan mask.
ScanMaskMinLevel	Float	Sets the minimum level to apply to the image based on the scan mask.
UpperBand	Float	Highest wavelength of band in microns.
VerticalFOV	Float	Refers to how far it can look up and down without physically moving, similar to the way you can scan a room with your eyes without moving your head.

Table 3-1: Attributes and types for sensor parameters (Sheet 8 of 8)

Attribute	Type	Description
VerticalPitch	Float	The vertical distance between detector centers and is used in determining focal length.
WindVelocity	Float	Specifies the wind speed to use for the Modtran calculations, in meters/second.
AtmosphereStateFile	String	Specifies a user defined atmosphere state file to use for the Modtran atmospheric calculations.
ColorMapUserDefined-RadFile	String	File defining a custom color mapping.
NightSkyTableFile	String	Specifies the file to use for the planetary and stellar irradiance information.
ScanMaskGainFilename	String	An image file to use as a mask for adjusting the gain of the final image.
ScanMaskLevelFilename	String	An image file to use as a mask for adjusting the level of the final image.

3.1.1. Calculating Noise

The default noise is varying Gaussian noise. The FInverseNoisePercent, FixedPatternNoise-Percent, and PoissanNoisePercent parameters indicate the percent of the noise to combine with the default noise. In other words if these parameters are 0, then the noise is 100% varying Gaussian. If not, the percentage of the noise that is varying Gaussian is whatever is left over after each of these noise sources are applied.

Index

L

M

N

O

P

R

S

T

U

W

Link - Simulate - Visualize

VRV-2.5-11-190221

150 CAMBRIDGE PARK DRIVE, 3RD FLOOR CAMBRIDGE, MA 02140 617.876.8085 www.mak.com