# SENSOR: a tool for the simulation of hyperspectral remote sensing systems

Anko Börner, Lorenz Wiest, Ralf Reulke, Rolf Richter

German Aerospace Center (DLR), Germany

Peter Keller, Michael Schaepman, Daniel Schläpfer

Remote Sensing Laboratories (RSL), University of Zurich, Switzerland

## ABSTRACT

The consistent end-to-end simulation of airborne and spaceborne earth remote sensing systems is an important task and sometimes the only way for the adaptation and optimization of a sensor and its observation conditions, the choice and test of algorithms for data processing, error estimation and the evaluation of the capabilities of the whole sensor system.

The presented software simulator SENSOR (Software ENvironment for the Simulation of Optical Remote sensing systems) includes a full model of the sensor hardware, the observed scene, and the atmosphere inbetween. The simulator consists of three parts. The first part describes the geometrical relations between scene, sun, and the remote sensing system using a ray tracing algorithm. The second part of the simulation environment considers the radiometry. It calculates the at-sensor radiance using a pre-calculated multidimensional lookup-table taking the atmospheric influence on the radiation into account. Part three consists of an optical and an electronic sensor model for the generation of digital images. Using SENSOR for an optimization requires the additional application of task-specific retrieval algorithms.

The principle of the end-to-end-simulation approach is explained, all relevant concepts of SENSOR are discussed, and first examples for its use are given. The verification of SENSOR is demonstrated. This work is closely tied to APEX, an airborne imaging spectrometer funded by the European Space Agency.

**Keywords:** hyperspectral, simulation, optimization, sensor, APEX

## 1. INTRODUCTION

The consistent end-to-end simulation of data of optoelectronic remote sensing systems has significant importance for the development, optimization, calibration, test, and application of such instruments and the interpretation of their data products. Generally, the interactions between the sensor, the observed scene, its environment, and the retrieval algorithms are much too complicated to be described by simple relations between the input and output quantities of the entire system. All relevant parts have to be understood as modules of one inter-dependent system in order to estimate the sensitivity of the results to different input parameters and disturbing factors.

Designing a remote sensing mission requires the adaptation of individual hardware parameters of a sensor system, observation conditions, and retrieval algorithms to certain scientific or commercial goals. In general, two approaches are used for this purpose. First, simple physical models are applied. This is the best way if there is a straight relationship between input and output quantities (e.g., the signal-to-noise ratio is directly proportional to the solid angle of the instantaneous field of view and the aperture size). Secondly, hardware testbeds can be used to determine an optimal system configuration (e.g. to evaluate the effect of using a mechanical instead of an electronic shutter).

On the other hand, if the input-output relationship is complex (e.g. non-linear) or the testbed facilities are not feasible because of unavailable human, technical, or financial resources, then the simulation approach is the only one which can support the

detailed system design, the optimization and evaluation of mission parameters (e.g. orbit selection) and the data processing procedures.

This paper presents the software simulator SENSOR (**S**oftware **EN**vironment for the **S**imulation of **O**ptical **R**emote sensing systems). Due to its modular structure it is possible to simulate a wide range of optoelectronic systems for airborne and spaceborne earth observation missions.

The basic concept behind SENSOR is to integrate all relevant knowledge about the hardware of the remote sensing system (e.g. optical distortions, dark signal), the radiation source, the atmosphere, and the observed scene (e.g. terrain shape) into a single model. In combination with data processing algorithms, SENSOR represents a powerful tool treating all mentioned parts as elements of a complex, unified system. The main objectives of applying SENSOR are:

- Evaluation of system specifications,
- Optimization of hardware parameters and observation conditions,
- Test of processing facilities,
- Choice, adaptation, and optimization of retrieval algorithms concerning computing time and accuracy,
- Accuracy and error estimation of data products, and
- Cost-versus-quality trade-offs.

The implementation and further development of the SENSOR approach was boosted significantly by the airborne imaging spectrometer APEX (Itten et al., 1997) in order to evaluate the impact of specification changes during the construction phase and for the test of processing facilities.

APEX was initiated by ESA and will be built as a joint Swiss-Belgian project. APEX is an airborne pushbroom imaging spectrometer featuring approximately 300 bands in the wavelength range between 400 and 2500 nm. One of the major technological achievements is the increased spectral resolution of 5 nm for APEX. Another goal is the use of pushbroom technology at a high number of up to 1000 pixels per line in order to simulate spaceborne instruments. Since the performance of APEX will mainly be driven by user requirements, the simulator for APEX is capable of monitoring the change of the specifications in terms of uncertainty statements in dedicated data products.

SENSOR pursues a similar philosophy as other end-to-end simulation approaches, but due to its complexity, SENSOR considers effects which are often ignored. For instance, in comparison to SENSAT (Richter, 1990a) a digital elevation model (DEM) and DEM related quantities, e.g. sky view factor, are considered. The main difference to a simulator proposed by Reulke (1995) is the integration of a complete atmospheric radiative transfer model.

The following section introduces SENSOR and its key concepts, then some validation results are reported, followed by a conclusion with a few examples.

## 2. SENSOR: AN END-TO-END-SIMULATION TOOL

Figure 1 depicts the general flow diagram of SENSOR. All essential input parameters, the processing modules, and the output (often being input parameters for the next processing step), are given from left to right. SENSOR consists of three main parts: (1) determination of the geometric relation between the remote sensing system, the radiation source, and the scene, (2) simulation of the influence of the atmosphere and calculation of the at-sensor radiances, and (3) description of the sensor hardware composed of optical and electronic components.

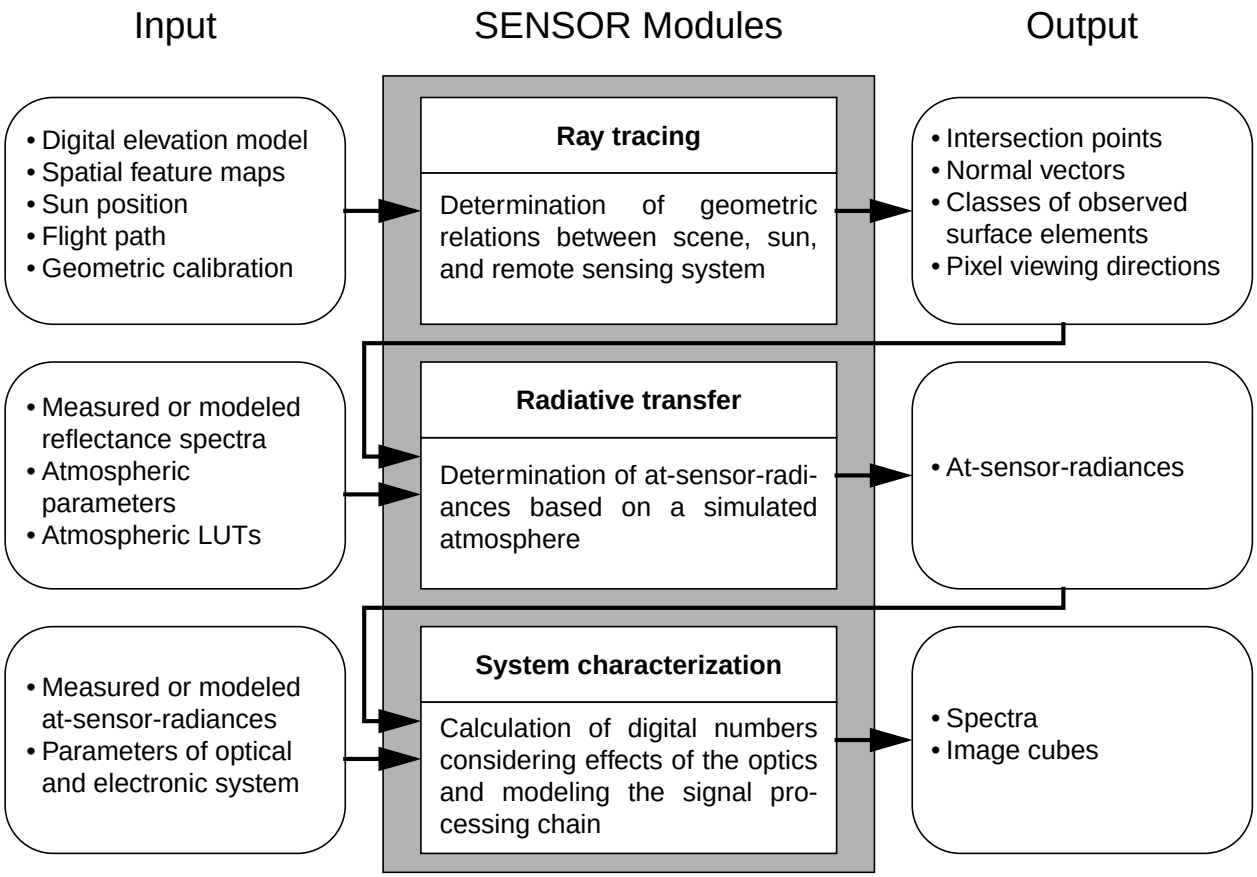


*Fig. 1: Scheme of the software environment for the simulation of optical remote sensing systems (SENSOR)*

### 2.1 Ray tracing

The task of the first SENSOR module is to determine the geometric relation between the observed scene (in general characterized by a digital elevation model), the radiation source (e.g. the Sun) and the remote sensing system by applying a ray tracing algorithm.

A flight over a DEM is simulated. Information about the sensor position (*x, y, z*) and attitude (roll, pitch, yaw), and the geometric calibration data can be provided by data of real sensor systems or can be simulated. Using this information, so-called pixel rays are defined. They describe the viewing direction vector of each detector element for all scan lines of the simulated flight. The next step is the determination of the observed objects and their properties. The class of the observed object (e.g. “wood”) is determined by a spatial feature map belonging to the DEM which gives a link to a spectral library including lambertian and bi-directional input reflectance spectra (simulated or ground-truth values). Other surface properties, like slope and aspect, are products of SENSOR’s ray tracing procedure or, like temperature and sky view factor, are given by additional spatial feature maps. Starting from a certain detector element, the pixel ray is traced until a surface element is hit. The following section gives an idea of this so-called ‘ray tracing’ in a DEM (see Figure 2).

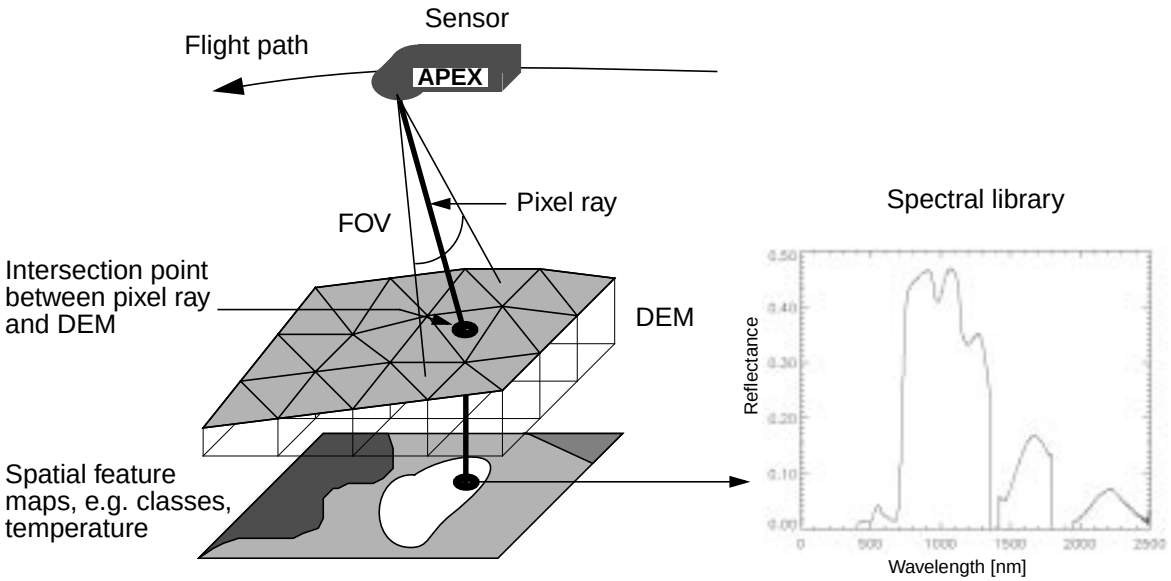


*Fig. 2: Ray tracing in a digital elevation model*

The DEM is described by a regular grid of triangles. The size of the DEM elements and the ground sampling distance of the sensor should be approximately the same. The task of ray tracing is to find the triangle seen by a certain detector element. This leads to finding the intersection point between the pixel ray and the appropriate triangle. In most cases, it is computationally far too expensive to check all triangles whether they were hit by a pixel ray or not, therefore fast algorithms are required.

After defining the pixel ray (by assigning each sensor pixel a certain position and a viewing direction), the intersection points are calculated between the pixel ray and two bounding planes defined by the maximum and minimum height of the DEM. Both intersection points form a sub-array within the DEM. The intersection point with the DEM lies within this sub-array. The next step is to find the triangles which lie on the foot print of the pixel ray (Figure 3). All these DEM elements have to be tested with respect to a valid intersection with the current pixel ray. In order to speed up the ray tracing procedure a prediction algorithm was implemented. It uses the high correlation between the location of the intersection points of two neighboring pixel rays, e.g. if the pixel *n* of scan line number *m* saw a certain DEM element, then it is very probable that pixel *n* of scan line number *m*+1 saw the same or a neighboring DEM element.

The speed of the procedure depends on the geometry of the remote sensing system (e.g. field of view), the flight motion, and the DEM parameters (height differences, resolution, steepness of slopes). It is applied once per spatial pixel, independent of the number of spectral bands. For example: Ray tracing algorithms run about 20 minutes on a Sun Ultra 60 for a simulated APEX scene of 1000 spatial pixels and 1500 lines in a mountainous region (Rigi mountain, Switzerland, height differences of about 1300 m within an area of 10 km $\times$ 10 km).

The output data of the ray tracing procedure are:

- Parameters concerning the geometry: intersection points with DEM, normal vectors of the observed DEM elements, pixel viewing directions for all spatial pixels of all image lines,
- Parameters concerning the properties of the observed objects: object classes, sky view factors, temperatures.

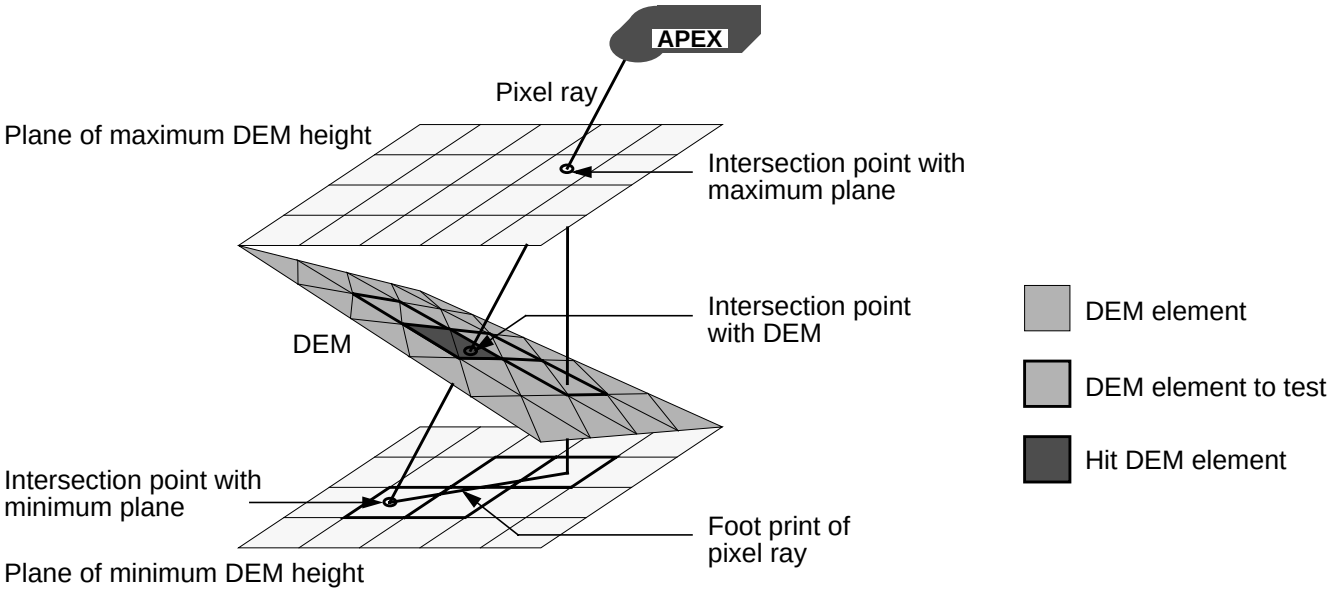


*Fig. 3: Principle of ray tracing in a digital elevation model*

### 2.2 Radiative transfer

After the determination of the geometric properties, the at-sensor radiance is calculated for each pixel of each image line in each spectral band. The radiative transfer describes the influence of the Earth's atmosphere on the solar irradiance. SENSOR uses results from the radiative transfer code MODTRAN (Berk et al., 1989), because this code offers a large wavelength range from the ultraviolet to the thermal region and great flexibility in varying viewing directions.

Internally, SENSOR applies a sub-band model. These sub-bands are the smallest spectral units. All calculations are done on this level. Each spectral band of the sensor consists of a number of sub-bands. The spectral sampling interval and the resolution of the sub-bands are set to 1 nm, exploiting MODTRAN's internal minimum spectral resolution of 1 $cm^{-1}$ which corresponds to a spectral resolution of about 0.625 nm at the uppermost wavelength being relevant for APEX (2500 nm).

At first, the reflectance of the observed object is needed for each sub-band. This information is available after ray tracing using the connection between object classes and spectral libraries. The reflectance spectrum must be given at the same sampling distance as the sub-bands. SENSOR is able to handle both lambertian and bidirectional reflectance distribution functions (BRDF) provided by spectral libraries.

An alternative way to import reflectance values is to access a reflectance data cube containing reflectance values for each spatial pixel. Such a data cube can be produced by resampling atmospherically corrected data sets of other, spectrally similar hyperspectral remote sensing systems (e.g., Schläpfer, 1999a), such as AVIRIS (Vane and Goetz, 1988).

The next step is the calculation of the at-sensor radiance of each sub-band. MODTRAN allows this calculation for a given band and a set of input parameters describing the view and illumination geometry, atmosphere, etc. To speed up the simulation's processing time, pre-computed lookup-tables (LUT) are used. These LUTs are created using an in-house developed tool. It automatically creates MODTRAN tape5 input files, sets their input fields according to the specified set of input parameter tuples and processes the customized MODTRAN tape7 output files into binary LUTs. Currently, an input parameter tuple consists of atmosphere and aerosol type, visibility, surface and sensor altitude, view zenith angle, relative azimuth angle, and solar zenith angle. A typical range of input parameter tuples for APEX is shown in Table 1. For a given set of input parameter tuples the overall computation time for the set of LUTs strongly depends on the computer and MODTRAN version used. For the example given in Table 1, a low-end computer (Pentium II, 333MHz) running MODTRAN 4.0 takes several days, creating 667 MB of LUT data.

| LUT input parameter | Number of entries | Values or Range |
|---|---|---|
| Atmosphere Type | 1 | Mid-latitude Summer |
| Aerosol Type | 1 | Rural |
| Visibility | 3 | 10, 15, 23 km |
| Surface Altitude | 11 | 0 - 2000 m in steps of 200 m |
| Sensor Altitude | 1 | 7500 m |
| View Zenith Angle | 9 | 140 - 180˚ in steps of 5˚ |
| Relative Azimuth Angle | 7 | 0 - 180˚ in steps of 30˚ |
| Solar Zenith Angle | 4 | 15 - 60˚ in steps of 15˚ |
| Wavelengths | 2101 | 400 - 2500 nm in steps of 1 nm |

*Table 1: A typical range of input parameter tuples for APEX*

It is advantageous and offers greater flexibility not to store the total spectral at-sensor radiance $L$ [W $cm^{-2}$ $sr^{-1}$ $m^{-1}$] for each tuple in the LUTs, but to consider $L$ as a sum of six spectral at-sensor radiance contributions (see Figure 4), which can be taken from modified MODTRAN output:

- $L_g$ Ground-emitted radiance
- $L_{pth}$ Thermal path radiance
- $L_{psc}$ Solar scattered path radiance
- $L_{dir}$ Directly reflected radiance
- $L_{dth}$ Thermal diffuse radiance
- $L_{dsc}$ Solar scattered diffuse radiance

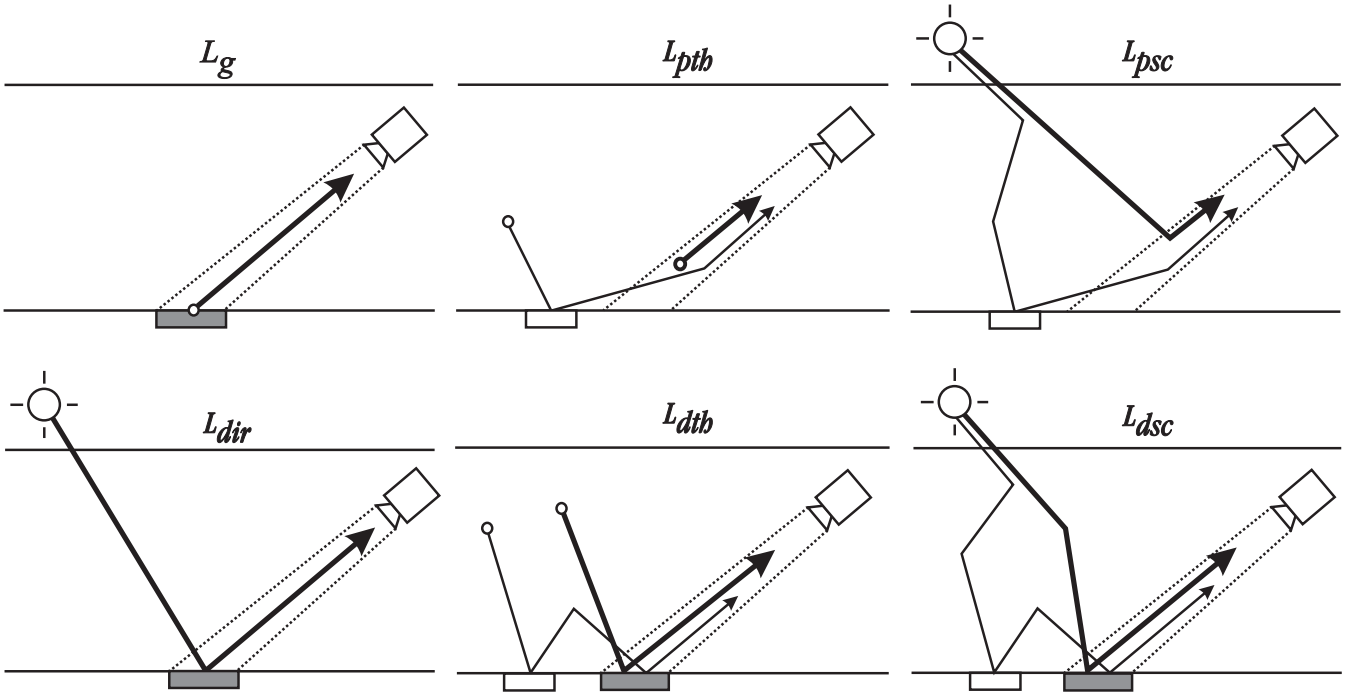


*Fig. 4: Contributions to the at-sensor radiance. Dark and white surface patches distinguish between a reflectance* $\rho$ *in the immediate field of view of the sensor pixel and an ambient reflectance* $\rho_a$*, respectively.*

MODTRAN treats the Earth's surface as flat smooth sphere. In order to simulate more realistic environments these radiance contributions need to be modified according to the surrounding terrain (slope, aspect, sky view, etc.).

In addition, all six at-sensor radiance contributions are implicitly surface reflectance dependent. To keep the size of the LUTs as small as possible, this dependency was modeled by functions linear in reflectance to include first-order reflectance effects (Richter, 1990b; Wiest and Reulke, 1999). The resulting set of equations is:

$$L_g = B(T_s)\ (1-\rho)\ \tau \tag{1}$$

$$L_{pth} = L_{pth0} + g_{pth}\ \rho \tag{2}$$

$$L_{psc} = (L_{psc0} + g_{psc}\ \rho)\ d \tag{3}$$

$$L_{dir} = L_{dirdown}\ \rho\ \tau\ \frac{\cos\theta_{ill}}{\cos\theta_{sz}}\ d\ b \tag{4}$$

$$L_{dth} = (L_{dth0down} + g_{dth}\ \rho_a)\ \rho\ \tau\ S \tag{5}$$

$$L_{dsc} = (L_{dsc0down} + g_{dsc}\ \rho_a)\ \rho\ \tau\ S_{hay}\ d \tag{6}$$

$$L = L_g + L_{pth} + L_{psc} + L_{dir} + L_{dth} + L_{dsc} \tag{7}$$

$B(T_s)$ is the blackbody radiance of the surface with temperature $T_s$. $\tau$ is the transmission from ground to sensor. $\rho$ is the surface reflectance in the instantaneous field of view (IFOV), $\rho_a$ is the ambient surface reflectance in the vicinity of the IFOV. When set not equal to $\rho$, this value can be used to simulate adjacency effects. $\theta_{ill}$ is the illumination angle between the surface normal and the sun direction, $\theta_{sz}$ is the solar zenith angle. $d$ is the square of the Earth-Sun distance in astronomical units (all LUT data are calculated for an Earth-Sun distance of 1 astronomical unit). $b$ is a binary factor which is set to zero if the surface in the IFOV lies in cast- or self-shadow, and 1 elsewhere. The values $\theta_{ill}$ and $b$ are delivered by the ray tracing section of the simulation. $S$ and $S_{hay}$ are sky view factors (explained below). $L_{pth0}$ and $L_{psc0}$ are the thermal and solar scattered at-sensor path radiances for a reflectance of zero. $L_{dirdown}$ is the downwelling direct-reflected radiance at the surface, $L_{dth0down}$ and $L_{dsc0down}$ are the downwelling thermal and solar scattered diffuse radiances at the surface. $g_{pth}$, $g_{psc}$, $g_{dth}$, and $g_{dsc}$ are quantities to model the linear reflectance dependence.

The ten values $L_{pth0}$, $g_{pth}$, $L_{psc0}$, $g_{psc}$, $L_{dirdown}$, $L_{dth0down}$, $g_{dth}$, $L_{dsc0down}$, $g_{dsc}$, and $-ln(\tau)$ (corresponding to the optical depth) are stored in a LUT. They belong to a specific input parameter tuple. By storing these values instead of a single total spectral at-sensor radiance in the LUT, the LUT's size is increased by a factor of ten. However, this approach offers much greater flexibility when using different scenarios of terrain shape and surface reflectances, because time-consuming re-runs of MODTRAN are avoided.

**Reflectance Parametrization**

To compute $g_{pth}$, $g_{psc}$, $g_{dth}$, and $g_{dsc}$, two MODTRAN runs are required at different reflectances. We chose values of $\rho = 0$ and 0.3. Our investigations show that the reflectance parametrization is a good approximation for all radiance contributions. Figure 5 illustrates the agreement of the modeled total at-sensor radiance compared with direct MODTRAN calculations for a typical scenario and several constant surface reflectances. The deviations in the short wavelength range are strongly dominated by the $L_{psc}$ contribution. This is due to strong multiple Rayleigh and aerosol scattering and multiple photon reflection at the surface.

At first glance, the reflectance dependency model for the $L_{psc}$ term needs refinement by allowing second order or higher functions in reflectance, leading to an increase of LUT size. However, this is not imperative: a comparison of 156 surface reflectance spectra covering agricultural, vegetation, mineral, and water specimens (Bowker et al., 1985) showed that in the wavelength range of 400 - 650 nm practically all spectra - with the exception of snow and ice - exhibit reflectance values below 0.5.

According to Figure 5, the overall agreement of the modeled at-sensor radiance with MODTRAN calculations for reflectances below 0.5 in this wavelength interval is better than 99%. The figure also shows that modeled at-sensor radiances for reflectances larger than 0.5 at wavelengths larger than 650 nm (being present, e.g. in vegetation reflectance spectra) agree better than 99% with MODTRAN calculations.

This leads to the conclusion that although the modeled at-sensor radiance exhibits deviations from MODTRAN it can be well applied to natural targets over the entire visible and near-infrared range - with the exception of snow and ice - due to the small reflectances they exhibit in the short wavelength range.

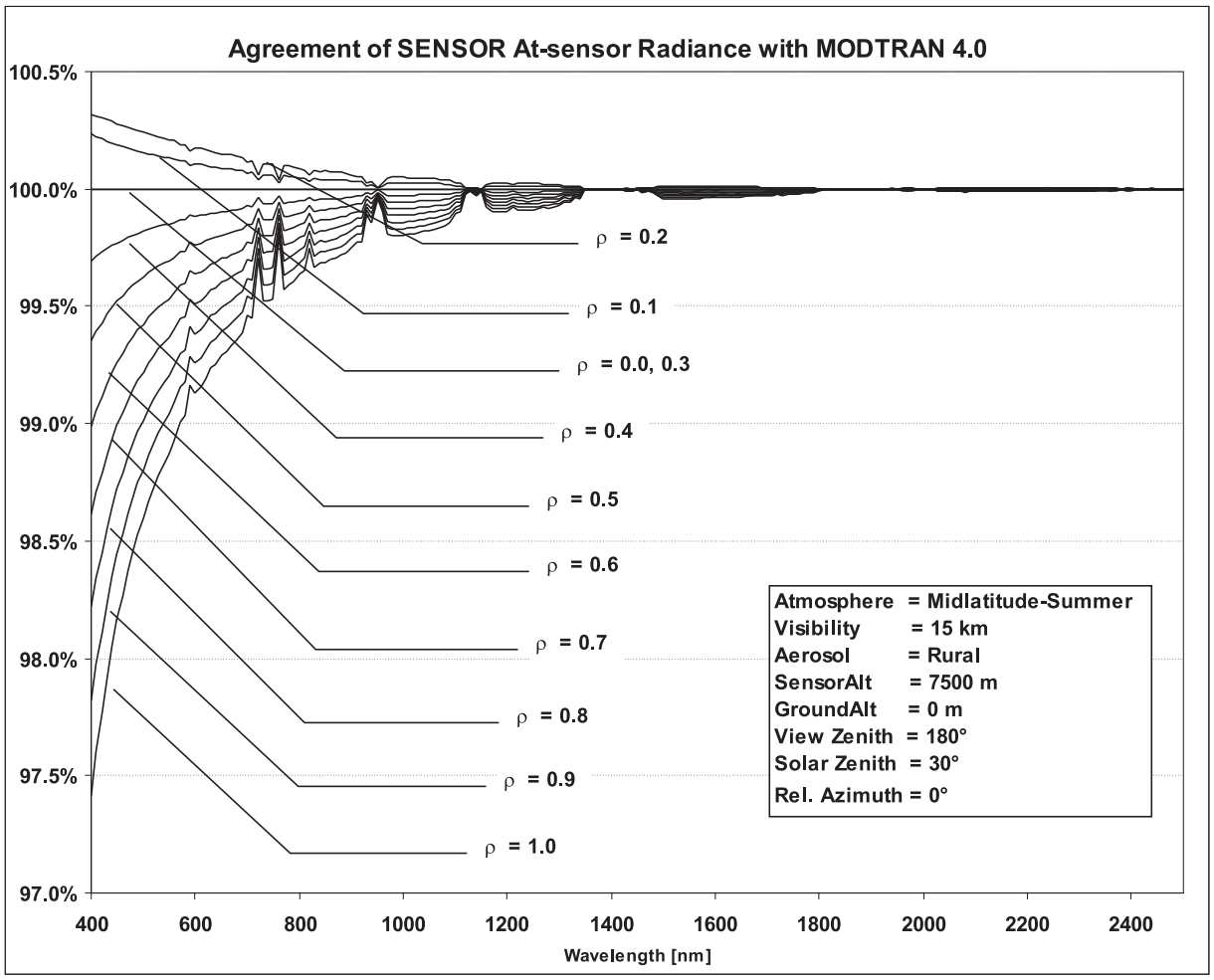


*Fig. 5: Agreement of the modeled total at-sensor radiance with direct MODTRAN calculations for several reflectances (midlatitude-summer atmosphere, rural aerosol, 15 km visibility, sensor altitude 7500 m, surface altitude 0 m, 30 degrees solar zenith, 0 degrees relative azimuth, nadir view).*

**Sky view factor**

The two diffuse radiances, taken from modified MODTRAN output, are considered isotropic and valid only for a surface in the IFOV with unobstructed view of the sky hemisphere. To improve simulation realism they are scaled by sky view factors $S$ and $S_{hay}$ to compensate for the obstruction of the horizon by surrounding terrain. The sky view factor $S$ is computed by the horizon-line algorithm (Dozier et al., 1981; Sandmeier, 1995) that scans the DEM for a given surface location along a number of radial azimuthal lines (typically 32) to determine the unobscured part of the sky hemisphere. The sky view factor for each DEM element needs to be computed only once and can be reused whenever the same DEM is used for a simulation. By applying optimized algorithms it takes 2 minutes for a $701 \times 481$ element DEM on a Pentium II, 333MHz computer. In order to fur-

ther benefit from the computation of the sky view factor it is currently investigated to exploit this information to simulate reflected terrain radiance.

Following a model by Hay and McKay (1985) a modified sky view factor $S_{hay}$ is applied to $L_{dsc}$ to account for the anisotropic diffuse radiance contribution of the circumsolar disc.

**Interpolating and Navigating the LUTs**

Each LUT is associated with a specific input parameter tuple. The radiance values of any other input parameter tuple are interpolated from the LUTs. Three interpolation methods are implemented within SENSOR (in decreasing speed but increasing complexity and accuracy): (1) nearest-neighbor, (2) first-order, and (3) multilinear interpolation.

The nearest-neighbor method (1) picks the LUT of the tuple most similar to the input parameter tuple. There is no interpolation between LUTs. Method (2) linearly interpolates radiance values from LUTs by considering just two neighbor values of the same parameter. Method (3) treats this task as a multilinear interpolation problem in a 6-dimensional parameter space (Original tuple parameter number minus atmosphere and aerosol type parameter, which are not interpolated) which makes it necessary to access $2^6$ LUTs per interpolation. This operation has to be repeated for each pixel and sub-band.

An efficient caching algorithm reduces computation time by a factor of about 4 by temporarily saving interpolation coefficients derived from the last-accessed LUTs in a separate memory store (cache). Any subsequent interpolation operation between input parameter tuples similar but not necessarily equal to the previous ones, accesses the interpolation coefficients being kept in the cache without repeating their time-consuming computation. Tests indicate that method (3) is only about 2 to 3 times slower than method (1).

With the presented radiative transfer module a spectral at-sensor radiance value can be assigned to each sub-band.

### 2.3 System characterization

This module describes the hardware of the remote sensing system considering the aspects of signal and system theory (Jahn and Reulke, 1995). It is divided into an optical and an electronics part. The aim is the calculation of digital numbers from the at-sensor radiance given either by the radiative transfer module of SENSOR or by radiance values provided by other hyperspectral remote sensing systems. The following parameters and effects influencing the conversion into digital numbers are considered:

- spectral response function for each band with a resolution of 1 nm,
- optical transmission given for each spatial and spectral pixel,
- photon noise,
- quantum efficiency,
- pixel response non-uniformity,
- dark signal,
- white noise,
- analogue-digital conversion,
- hardware parameters, e.g. size of entrance aperture, integration time,
- point spread function, and
- blurring by flight motions.

Each spectral band is characterized by a spectral response function. It describes the sensitivity of the band with regards to the energy in a certain range of the electromagnetic spectrum. In order to model a continuous response function, SENSOR defines the response function of each band as a set of discrete samples (sampling distance 1 nm corresponding to one sub-band). The first step is the calculation of the number of generated photons arriving at the detector for each sub-band by the equation

$$n_p = A_e \, \Omega \, t_{int} \int_{\lambda_1}^{\lambda_2} R_0(\lambda) \, \frac{\lambda}{h \, c} \, L(\lambda) \, d\lambda, \tag{8}$$

where $n_p$ is the number of photons, $A_e$ the area of the entrance aperture, $\Omega$ the solid angle of the instantaneous field of view, $t_{int}$ the integration (exposure) time, $R_0(\lambda)$ the response function of the entire remote sensing system, $\lambda$ the wavelength, and $L(\lambda)$ the spectral at-sensor radiance. The response function of the entire sensor system including the transmissivity of the optics must be given for each pixel of the detector CCD (spatial and spectral direction) and is the result of a calibration process. All other parameters are fixed during the simulated flight. Assuming constant values for the spectral dependent quantities $R_0(\lambda)$ and $L(\lambda)$ within a sub-band $i$ results in the number of photons $n_p(i)$.

The photon noise $n_{np}(i)$ must be considered. The standard deviation of this poisson-distributed value is given by $\sqrt{n_p(i)}$. Hence, the number of electrons freed by the photons is a random number calculated by

$$n_e(i) = (n_p(i) + n_{np}(i))\ \eta(i)\ , \tag{9}$$

where $n_e(i)$ is the number of electrons and $\eta(i)$ the wavelength dependent quantum efficiency. Most approaches end at this stage, ignoring important noise sources and processing steps. In contrast to these models, SENSOR includes parameters characterizing the hardware of the remote sensing system. A quasi-convolution in the spectral direction is performed by summing up the generated electrons of the sub-bands into one spectral instrument band.

Behind the optics, an electronic data processing chain is modeled, which includes certain noise sources (e.g. pixel response non-uniformity and dark signal) and an analog-digital-converter. All sources of noise are assumed to be gaussian-distributed (for large values the poisson distribution can be described as gaussian) and can be summarized in the system noise quantity $n_n$. Its standard deviation $\sigma$ is determined by

$$\sigma = \sqrt{(n_{np}\ \eta)^2 + {n_{nel}}^2 + {n_{nadc}}^2}\ , \tag{10}$$

with the noise electrons $n_{np}\ \eta$ caused by photon noise, by the electronics noise $n_{nel}$ and by the quantization noise $n_{nadc}$ due to the analog-digital-conversion.

The output is one single digital number per spatial and spectral pixel and per image line. The conversion of the band-integrated at-sensor radiances $\bar{L}$ (which are no longer spectrally dependent) into digital numbers corresponds to an inverse radiometric calibration obeying the equation

$$\bar{L} = DN\ c_1 + c_0\ , \tag{11}$$

where $c_0$ and $c_1$ are system and campaign specific calibration coefficients provided by the manufacturer or calibration laboratory. The goal of the simulation approach is to simulate these calibration coefficients for each pixel. Additionally, it is possible to skip the calculations applying the equations (8) to (10) and to use band-wise coefficients ($c_0$, $c_1$) for the radiance-to-digital-number conversion instead.

The influence of the point spread function (PSF) of the system and blurring caused by the flight motion during integration time is considered by applying convolution kernels across and along flight track. This is performed for each spectral image, considering neighboring pixels. The final result is a simulated image data cube.

SENSOR is able to deal with problems occurring especially when hyperspectral systems are simulated. A few examples are:

- A mixed pixel signal can be simulated by dividing one spatial pixel into $n \times n$ sub-pixels with their digital number outputs summed at the end, assuming linear mixing processes.
- Smile and frown effects can be simulated using different data sets of the geometrical calibration for each spectral and spatial pixel. This results in different intersection points in the ray tracing procedure, different observed objects, and consequently in a “distorted” image.

### 2.4 Optimization process

The optimization or evaluation concept using SENSOR is outlined in Figure 6. A set of system parameters provided by the scientific team and the industrial partners is fed into SENSOR. The simulation outputs an artificial image data cube. This data

cube will be processed by a number of retrieval algorithms. The output data generated this way can be compared to user driven requirements, e.g. data quality and uncertainties, feasibility considerations, and cost analysis. This leads to an evaluation of the input parameters and, if necessary, to an update of the sensor design.

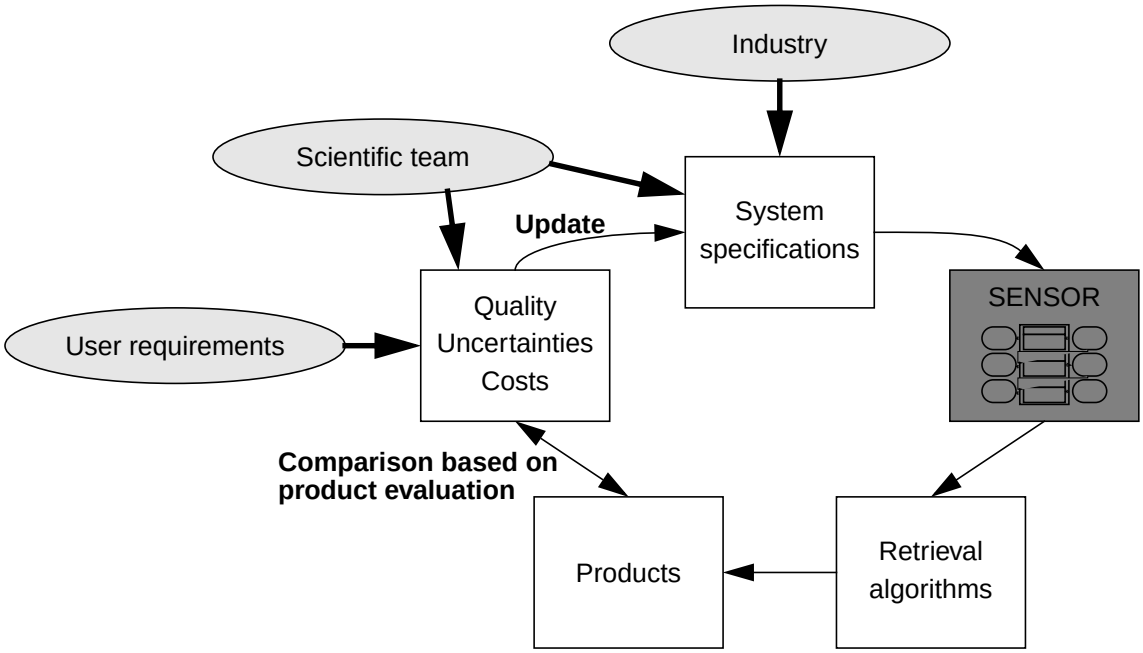


*Fig. 6: General scheme for an optimization or evaluation process using SENSOR*

A few examples show which projects were done or which are currently in progress applying SENSOR. They give an idea of the variety of possible applications for this tool:

- optimization of the stereo angle of CCD line scanners (Börner, 1996),
- test of the APEX processing and calibration facility (Schläpfer et al., 1999b),
- choice of color channels of LH Systems Airborne Digital Sensor.

## 3. SIMULATED DATA

### 3.1 Example

Figure 7 shows an APEX image cube simulated by SENSOR. The image gives an unnatural impression due to a coarse subdivision of object classes. The first and last lines are the in-flight calibration data. Before and after each data acquisition, the sensor pixels are illuminated by an internal calibration sphere and a few image lines will be scanned with a closed shutter. The black pixels along the left and right border are used for dark current measurements (Schaepman et al., 1999). Currently, it takes about 20 hours on a Sun Ultra 60 to simulate one entire APEX image cube with a size of 1000 pixels $\times$ 1500 image lines $\times$ 300 spectral bands.

In order to illustrate SENSOR's capabilities, the digital output of the simulation was calculated in dependence of a number of varying input parameters. Referring to Figure 1, the output of the simulation (digital numbers) was obtained by changing parameters at all stages of the simulation, starting from the spectral properties of a target, continuing with a significant atmospheric parameter, and finishing with the system hardware itself (see Figure 8).

First, the reflectance of an target was changed. The reflectance values of a measured vegetation spectrum were multiplied by 0.9 and 1.1, respectively, to simulate measurement uncertainty due to a spatial non-uniformity. The goal was to show the effect of such an uncertainty to the digital output of the SENSOR simulation.

Secondly, the influence of varying a typical atmospheric parameter was investigated. The visibility was changed to 10, 23 and 40 km. The resulting image spectrum for one pixel is depicted.

Thirdly, two instruments noise sources (dark signal and white noise) were set to a value being one hundred times larger than currently expected (see equation (10)). The digital output is plotted in comparison to the nominal values.

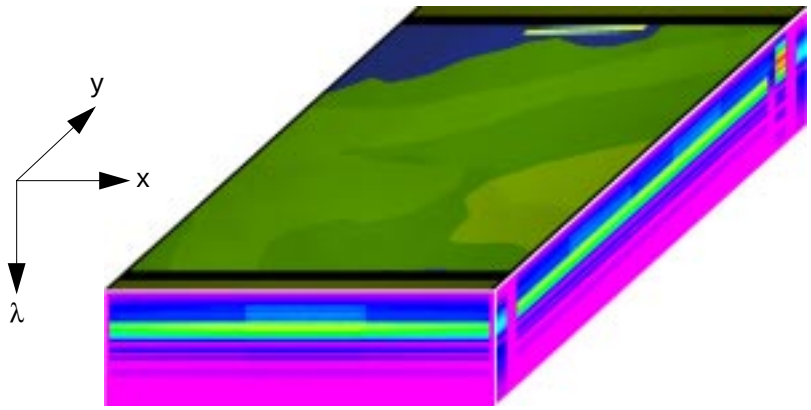


*Fig. 7: SENSOR simulated APEX data cube*

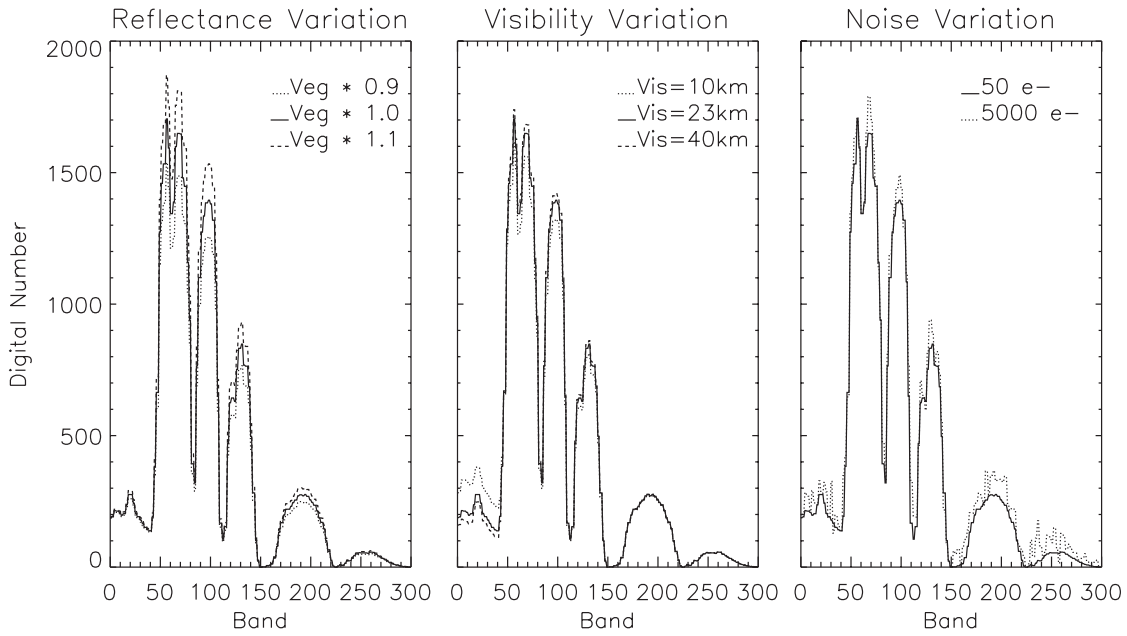


*Fig. 8: SENSOR simulated image spectra. Digital output in dependence of input reflectance (left), visibility (center) and hardware noise levels (right) with respect to a reference vegetation spectrum (solid line).*

### 3.2 Verification

It is an essential prerequisite that the consistency of SENSOR-simulated data is verified. The verification of the geometric simulation is performed easily by comparing the original elevation model with all calculated intersection points between pixel rays and DEM.

In addition to the comparison between direct MODTRAN output and the at-sensor radiance simulated by SENSOR (see (2.2)) the following procedure is carried out to estimate the radiometric accuracy of SENSOR simulated data:

- Simulation of a HyMap (Cocks et al., 1998) flight over an area (all input parameters are known exactly, e.g. reflectance spectrum of a selected target, calibration coefficients) leading to an image cube,
- Application of an independent MODTRAN based atmosphere correction program (ATCOR4 Richter, 1999) to the simulated data leading to a reflectance spectrum again.

In an ideal simulation the original input reflectance and the retrieved output reflectance spectra should be the same. The comparison of a wavelength independent input spectrum ($\rho = 0.5$) and a real vegetation spectrum and their corresponding output spectra are shown in Figure 9.

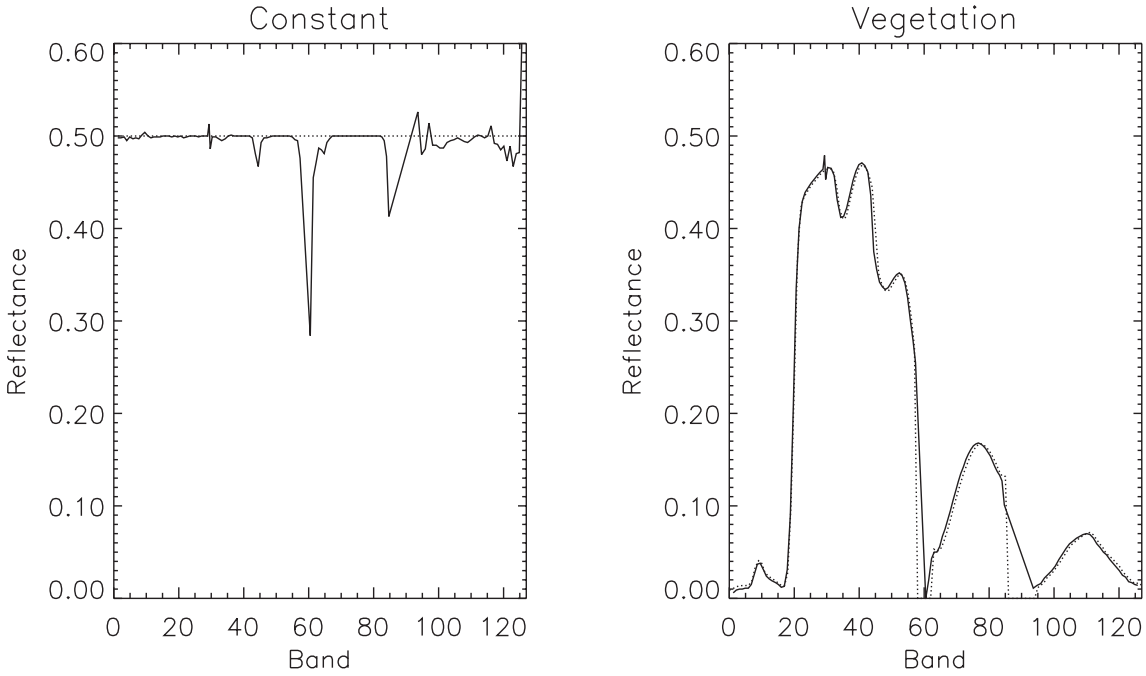


*Fig. 9: Comparison of original (dotted line) and reconstructed (solid line) spectra of a constant reflectance of 0.5 (left) and a vegetation spectrum (right). The outliers at bands 64, 95, and 127 in the left plot are artifacts caused by small digital values obtained from the at-sensor radiance. The deviations at wavelengths larger than 1.8 m(corresponding to HyMap band 90) are caused by different spectral resolutions of SENSOR and ATCOR4.*

By using a worst-case example (tropical atmosphere, 15 km visibility, sensor altitude 7.5 km, nadir sensor viewing zenith, rural aerosol, 30˚ solar zenith), the comparison between the input and the output spectra to assess overall accuracy yields a difference of the means smaller than 0.5% and a standard deviation smaller than 1.0% for both spectra. The bands 64, 95, and 127 were ignored, since the small simulated digital numbers lead to large discretization error. Detailed studies showed, that the biggest deviations we have to deal with are about 5% in absolute reflectance within one sub-band within water vapor absorption bands.

## 4. CONCLUSIONS

The complex end-to-end simulation tool SENSOR has been presented. It describes the sensor hardware itself, the observed scene, and the atmosphere. With this tool, the interactions between parameters of the complex model, retrieval algorithms, and any output, such as data accuracy and costs, can be evaluated. In principle, it is possible to model any optoelectronic remote sensing system since due to its modular structure, merely the hardware part has to be changed. SENSOR differs from other approaches by its high complexity. All parts of the complex sensor-environment system are considered and advanced features are implemented, e.g. ray tracing, fast and flexible access to atmosphere LUTs, sky view factor, point spread function, and noise sources.

## 5. OUTLOOK

SENSOR is an on-going project, for which the following features will be investigated in order to enhance its accuracy and applicability:

- modeling of terrain reflected radiances,
- expand SENSOR for use in the infrared wavelength range,
- compression of the LUTs,
- including additional optical effects, e.g. straylight models, and
- including additional electronic effects, e.g. temperature dependent noise sources and cross talk.

In the near future, SENSOR will be applied to the design and support of imaging remote sensing systems, e.g. the verification of APEX hardware components. We hope SENSOR will give significant contributions to such tasks.

## 6. REFERENCES


Berk A., L.S. Bernstein, D.C. Robertson, 1989. MODTRAN: A Moderate Resolution Model for LOWTRAN7, Report GL–TR–89–0122, Geophysics Lab., pp. 38, Bedford, USA

Börner A., 1996. The Optimization of the Stereo Angle of CCD-Line-Scanners, ISPRS Vol. XXXI, Part B1, Commission I, pp. 26-30, Vienna 1996

Bowker David E., Davis Richard E., Myrick David L., Stacy Kathryn, and Jones William T, 1985. Spectral Reflectances of Natural Targets for Use in Remote Sensing Studies. NASA Reference Publication 1139

Cocks T., Jenssen R., Stewart A., Wilson I., and Shields T., 1998. The HyMap airborne hyperspectral sensor: the system, calibration and performance. Proceedings of the 1st EARSeL Workshop on Imaging Spectroscopy, Zurich, Switzerland, pp. 37-42

Dozier J., Bruno J. and Downey P., 1981. A Faster Solution to the Horizon Problem. Computers and Geosciences, 7, pp. 145-151

Hay J.E. and McKay D.C., 1985. Estimating Solar Irradiance on Inclined Surfaces: A Review and Assessment of Methodologies. International Journal of Solar Energy, pp. 203-204

Itten K.I., Schaepman M., De Vos L., Hermans L., Schläpfer D., and Droz F., 1997. APEX - Airborne PRISM Experiment: A New Concept for an Airborne Imaging Spectrometer, In: Third International Airborne Remote Sensing Conference and Exhibition, Vol. 1, pp. 181 - 188, 7-10 July, Copenhagen, Denmark

Jahn H. and Reulke R., 1995. Systemtheoretische Grundlagen optoelektronischer Sensoren; Akademie Verlag, Berlin, pp. 298

Reulke N., 1995. Simulation und Optimierung optoelektronischer Systeme am Beispiel der Bestimmung von Wolkengeschwindigkeit und -höhe; DLR-Forschungsbericht 95-45, pp. 107

Richter R., 1999. Atmosperic/Topographic correction for wide FOV airborne imagery: model ATCOR4, Version 2.0, DLR Report DLR-IB 552-05/99, German Aerospace Center, Institute for Optoelectronics

Richter R., 1990a. Model SENSAT: A Tool for Evaluating the System Performance of Optical Sensors, Proc. SPIE, 1312:286–297

Richter R., 1990b. A fast atmospheric correction algorithm applied to Landsat TM images, International Journal of Remote Sensing, 1990 Vol. 11, No. 1, pp. 159-166

Sandmeier S.R., 1995. A Physically-Based Radiometric Correction Model. Dissertation, Universität Zürich, Zürich.

Schläpfer D., Boerner A., and Schaepman M., 1999a: The Potential of Spectral Resampling Techniques for the Simulation of APEX Imagery based on AVIRIS Data. Summaries of the Eighth JPL Airborne Earth Science Workshop, JPL, Pasadena (CA), 99-17:377-384.

Schläpfer D., Schaepman M., Bojinski S., Börner A., 1999b. Calibration and Validation Concept for the Airborne PRISM Experiment (APEX), Canadian Journal of Remote Sensing, pp. 13

Schaepman M., Schläpfer D., Börner A., Bojinski S., Itten K., 1999. A new Airborne Hyperspectral Imager for the Simulation of ESA's Land Surface Processes and Interactions Mission, ISSSR, *in print*

Vane G. and Goetz A.F.H., 1988: Terrestrial Imaging Spectroscopy. Remote Sens. Environ., Elsevier Science Publishing Co., Inc., New York, Nr. 24, pp. 1-29.

Wiest L. and Reulke R., 1999. Radiometric Simulation and Verification of the Line Scanner Camera WAAC. Proc. Fourth Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing Vol.II, pp. 24-31