Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Overhead longwave infrared hyperspectral material identification using radiometric models

Michael E. Zelinski

SPIE.

Michael E. Zelinski, "Overhead longwave infrared hyperspectral material identification using radiometric models," *J. Appl. Remote Sens.* **12**(2), 025019 (2018), doi: 10.1117/1.JRS.12.025019.

Overhead longwave infrared hyperspectral material identification using radiometric models

Michael E. Zelinski*
Lawrence Livermore National Laboratory, Livermore, California, United States

Abstract. Material detection algorithms used in hyperspectral data processing are computationally efficient but can produce relatively high numbers of false positives. Material identification performed as a secondary processing step on detected pixels can help mitigate false positives. A material identification processing chain for longwave infrared hyperspectral data of solid materials collected from airborne platforms is presented. The algorithms utilize unwhitened radiance data and Nelder–Meade numerical optimization to estimate the temperature, humidity, and ozone levels of the atmospheric profile. Pixel unmixing is done using constrained linear regression and Bayesian information criteria for model selection. The resulting identification product includes an optimal atmospheric profile and a full radiance material model that includes material temperature, abundance values, and several fit statistics. A logistic regression method utilizing the model parameters to improve identification is also presented. Several examples are provided using modeled data at several noise levels. [DOI: 10.1117/1.JRS.12.025019]

Keywords: hyperspectral; longwave infrared; radiometric modeling; detection; identification.

Paper 180123 received Feb. 9, 2018; accepted for publication May 23, 2018; published online Jun. 30, 2018.

1 Introduction

Longwave infrared (LWIR) hyperspectral imagers (HSI) can capture high spectral resolution measurements of the electromagnetic spectrum between 7.5 and 13.5 μm.[1,2] In this spectral region, many gas[3] and solid materials[4,5] have spectral emission/absorption features that are observable by LWIR HSI. Spectral analysts match observed spectral features found in the data with those found in a spectral database of materials. This is often performed using automated material or target detection algorithms.[6] Detection algorithms are designed to be computationally efficient and process spectra quickly; however, they have typical false positive rates of $\sim 10^{-5}$ per pixel for common threshold settings. Detections above the threshold are flagged for examination by a spectral analyst. As the number of sensors increase and the sensors themselves improve, collecting greater numbers of pixels, the number of false positives will begin to overload the current number of spectral analysts. To mitigate the effect of costly false positives, a "material identification" algorithm can be added to hyperspectral data processing chains.[7,8] Material identification performs a more thorough analysis on a single pixel (or region) of interest that passed the detection threshold. It is often more time-consuming than detection algorithms and is not practical to run on a full scene. The resulting product can provide the spectral analyst with more information about the contents of the pixel. This information, which is often quantitative, can also be used to set additional thresholds to suppress false positives found during the detection step.

This approach requires estimation of atmospheric parameters (transmission, downwelling radiance, and upwelling radiance). Temperature emissivity separation algorithms provide a method of acquiring these terms. The in-scene atmospheric compensation (ISAC)

*Address all correspondence to: Michael E. Zelinski, E-mail: zelinski1@llnl.gov algorithm[9] utilizes in-scene blackbodies in an algorithm that provides an estimate of atmospheric transmission and upwelling. This method was developed for Aerospace Corporation's Spatially Enhanced Broadband Array Spectrograph System (SEBASS). SEBASS is a dispersive HSI system that utilizes a liquid helium-cooled focal plane array that has very well-behaved noise structure and few dead pixels. Not all LWIR HSI sensors have these characteristics. Another important limitation with ISAC is that blackbody materials are not found in all datasets (for example, desert scenes). Also, many locations do not have spatially uniform atmospheric profiles across the scene. Because of these limitations, the ISAC algorithm is not appropriate for many LWIR HSI imaging scenarios. Other approaches[10] require measuring the atmosphere by sounding and then using this data in a radiative transport code, such as MODerate resolution atmospheric TRANsmission (MODTRAN)[11] to simulate the atmospheric parameters. This approach has limitations relating to the availability of time/location appropriate sounding data. There are also several methods to "search" for correct atmospheric and model terms. This is done using precomputed look-up tables for the atmosphere[12–14] and spectral emissivity smoothness as a metric for determining appropriate model parameters (solid materials have broader spectral features than atmospheric gasses). The processing chain presented here utilizes some of these concepts in an approach that does not depend on in-scene blackbodies, sounding, or a spatially uniform atmosphere. It should be noted that these traditional algorithms are likely more computationally efficient than the processing chain described in this paper.

This paper demonstrates a physics-based processing chain for performing material identification by unmixing nonwhitened LWIR HSI radiance data. Spectra are unmixed by producing radiance models that match measured scene spectra. The models are comprised of background endmembers and emissivity spectra that are forward modeled to radiance. Models and scene measurements are compared using root-mean-squared (RMS) error.

As mentioned earlier, researchers have used sounding to obtain atmospheric temperature, humidity, and ozone-level profiles. This paper demonstrates how to acquire an estimate of the atmospheric profile using an optimization algorithm. The temperatures and abundances for the material of interest can also be determined. If all parameters in the model are correct, then it should match the measurement. If it does not match, then it is unlikely that the pixel under inspection contains the material of interest.

The processing chain consists of two primary steps: "atmospheric inference" and "radiometric modeling/pixel unmixing." Atmospheric inference is fast enough to be applied to many pixels and potential atmospheres. Its primary function is to find an estimate of the atmospheric profile, as it performs a relatively broad search compared to the radiometric modeling/pixel unmixing step. It also does not operate on the pixel of interest. The radiometric modeling/pixel unmixing step operates on the pixel of interest and aims to produce a radiometric model that includes several fit parameters and statistics related to how well the model fits the pixel of interest. Figure 1 provides the reader with a summary view of the algorithms.

Whether this processing chain is applied to real data or simulated data, there are several important assumptions that need to be stated:

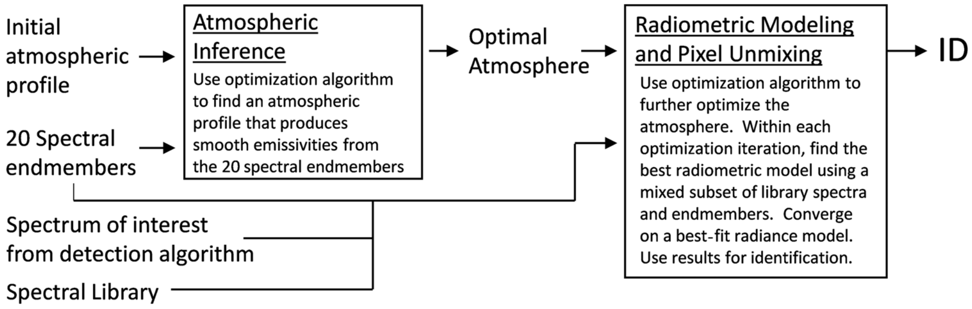
Initial atmospheric profile
20 Spectral endmembers
Atmospheric Inference
Use optimization algorithm to find an atmospheric profile that produces smooth emissivities from the 20 spectral endmembers
Optimal Atmosphere
Radiometric Modeling and Pixel Unmixing
Use optimization algorithm to further optimize the atmosphere. Within each optimization iteration, find the best radiometric model using a mixed subset of library spectra and endmembers. Converge on a best-fit radiance model. Use results for identification.
ID
Spectrum of interest from detection algorithm
Spectral Library

Fig. 1 This flowchart provides a basic view of the data processing pipeline.

1. Sensors accurately and precisely measure radiance. This means that it is desirable to have well-behaved low-noise systems that are radiometrically and spectrally calibrated. The data should not have artifacts relating to spectral–temporal data collection as one might find with a Fourier transform interferometer. Dispersive systems, with low smile and keystone, are favored.
2. MODTRAN simulates atmospheric transmission, upwelling, and downwelling, both accurately and precisely.
3. Lambertian radiance models are appropriate.
4. Spectral libraries accurately characterize materials as they are found in nature.

This paper is organized as follows. Section 2 describes MODTRAN and provides several important guidelines for using it in this study. Section 3 describes an approach for obtaining an optimal atmospheric column parameterization. Section 4 details the Lambertian radiance model used in this study. Section 5 describes the calculations that occur within each iteration of the atmospheric inference and pixel unmixing algorithms. Section 6 describes how logistic regression can be used with multiple model output parameters for identification and reducing false positives. Section 7 uses the data processing chain in multiple simulations and discusses the results. Section 8 provides a conclusion. Throughout the document are several implementation notes that readers should follow if they choose to implement these algorithms.

The radiance unit used here is a micro-Flick (μF), which is a $\mu\mathrm{W}/(\mathrm{cm}^2\,\mathrm{sr}\,\mu\mathrm{m})$.

2 MODTRAN

MODTRAN is a highly capable tool for radiative transport calculations in the Earth's atmosphere at altitudes ranging from below sea level (e.g., Death Valley) to 100 km and wavelengths between 0.2 and 10,000 μm at a spectral resolution of 0.1 cm^{-1}.

This research makes use of MODTRAN's "Card2C1" to define the temperature, humidity, and ozone levels. The optimization code parameterizes the profile at four altitudes. The lowest altitude is at ground level. The second altitude is at 300 m where the atmospheric boundary layer could exist. The third altitude is at the aircraft altitude. If the aircraft has an onboard temperature/humidity/ozone sensor, the data can be used by the algorithm; this will be discussed in Sec. 7. The fourth altitude is at 10 km. Between each of the defined altitudes, seven additional atmospheric layers are computed using linear interpolation for a total of 25 layers.

For the radiance models defined below, the atmospheric transmission and the upwelling radiance values are calculated by positioning the MODTRAN observer at the altitude of the HSI sensor. At this setting, the target temperature should be set to 0 K. The downwelling term is calculated by placing the observer 1 m above the target and setting the target reflectance to 1. The Lambertian reflectance model was used in this work.

Atmospheric band radiance and transmission values vary significantly in narrow wavelength regions—much narrower than the spectral resolution studied here.[9] Therefore, good spectral calibration (band center and shape) is critical. This approach makes use of simulated data, therefore the spectral calibration is known; however, in a real HSI system, wavelength calibration bandcenters should be known to be within 1/10th of a spectral bin across the entire focal plane array (accounting for spectral smile and keystone).

The code used to interact with MODTRAN was written in MATLAB™ and makes use of the MODTRAN class wrapper[15] to set the values in the tape5 file. The class wrapper includes commands to run MODTRAN as well as read the output in the tape7 file.

MODTRAN calculations were performed at 0.1-cm^{-1} spectral resolution between 1400 and 700 cm^{-1}. The resulting atmospheric arrays had 7001 elements. When appropriate, radiometric calculations were done in high spectral resolution (specified in the Figs. 3 and 4). Downsampling was done by integrating under a Gaussian (sigma value of 0.024 μm) spectral line shape for each spectral band. This approach used a simulated sensor with 178 bands and linewidth $0.024\ \mu\mathrm{m}/\mathrm{band}$, covering 8.86 to 13.1 μm. Outside of this region the water vapor features are large and high frequency such that they are not well sampled by MODTRAN at 0.1-cm^{-1} resolution. Downsampling is done efficiently by creating a bandpass array (dimensions 7001×178) that is applied to the high-resolution vectors with a dot product.

Table 1 The constraints used on atmospheric optimizations. "tp_aX," "dp_aX," and "oz_aX" indicate the temperature, dew point, and ozone constraints at altitude "*X*," respectively. The values in the table indicate the boundary around the initialization value. For example, if tp_a1 is 20, then the boundary for the atmospheric inference method would be [5°C, 35°C]. The ozone layer uses the initial value plus or minus the initial value "*i*" divided by 2. This prevents negative ozone values from being used.

Algorithm	tp_a1	tp_a2	tp_a3	tp_a4	dp_a1	dp_a2	dp_a3	dp_a4	oz_a1	oz_a2	oz_a3	oz_a4	Iteration
Atm. Infer.	−15, 15	−16, 16	−16, 16	−16, 16	−16, 16	−16, 16	−16, 16	−16, 16	$-i/2$, $i/2$	$-i/2$, $i/2$	$-i/2$, $i/2$	$-i/2$, $i/2$	60
Unmix	−5, 5	−5, 5	−5, 5	0, 0	−5, 5	−5, 5	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	15
Unmix	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	0, 0	**0, 0.4**	**0, 0.4**	0, 0	0, 0	25
Unmix	−5, 5	−5, 5	−5, 5	0, 0	−5, 5	−5, 5	−5, 5	0, 0	0, 0	$-i/2$, 0.1	0, 0	0, 0	60

Note: Bold numbers indicate hard boundaries that are not relative to the initialization value.

3 Atmospheric Column Parameter Optimization

The atmospheric inference and pixel unmixing methods utilize MODTRAN estimates of the atmospheric spectral transmission, downwelling, and upwelling. MODTRAN atmospheres are parameterized by defining the temperature, dew point, and ozone profiles of the atmospheric column at four altitudes. As will be described in Sec. 5, if these atmospheric terms are known, one can expect smooth emissivities and low error radiance models. Typically, atmospheric sounding is used to measure these parameters. Alternatively, metrics relating to emissivity smoothness and model error can be used with an optimization algorithm to determine a "best" or optimal atmosphere.

Nelder–Meade (a.k.a. "simplex" or "amoeba" algorithm)[14,16] is a common numerical optimization algorithm that does not require an analytical derivative. There are implementations in many coding packages, such as Python or MATLAB™. An open-source constrained version of the algorithm in MATLAB™ allows users to set boundaries on each optimization parameter.[17] This is particularly useful when the objective function has local minima and the user knows the initialization is close to an optimal solution.

Numerical optimization algorithms require initialization points and there are several acceptable ways to do this. One way is to use a set of atmospheric profiles comprised of biased versions the standard atmospheres. Biasing is done by generating a set of atmospheres where the ground temperature is centered at the median scene temperature and biased in increments of 2°C above and below that value. Users should create a set that spans any potential extremes that the atmospheric profile might have. This step can be subjective and rely on the user's intuition of what the potential atmospheric conditions might be. The author, at the time of publication, uses a custom set of atmospheres for this step. Using this set, the atmospheric inference step is run for a single iteration on each atmosphere in the set. This is then repeated using the same set of atmospheres but with varying amounts of tropospheric ozone. The best atmosphere in the set is selected as the initialization point for the atmospheric inference, which is then run for 60 iterations. The output of this is then used as an initialization point for the pixel unmixing method, which is run several times with different initialization constraints (see Table 1). Each successive run aims to improve upon the previous estimate of the atmospheric parameters. The first run improves the estimate of the atmospheric column temperature and humidity. The second run improves the estimate of the tropospheric ozone concentration. The third run fine-tunes the most sensitive portions of the profile. As the optimization algorithm defines new atmospheres, they are saved in a database.

4 Longwave Infrared Lambertian Radiance Model

The LWIR Lambertian radiance model[18] is defined as

$$L = B(T)\varepsilon\tau + (1-\varepsilon)\tau L_d + L_u, \tag{1}$$

where L is the at-sensor radiance (μF), $B(T)$ is the blackbody radiance defined at temperature T, ε is the material emissivity, τ is the transmission of the atmosphere from the ground to sensor, L_d is the downwelling radiance at ground level, and L_u is the upwelling radiance between the ground and sensor. The spectral nature of each component is implied, and the λ subscript has, therefore, been omitted from the equation.

Section 5.2 makes use of a target leaving radiance model that incorporates mixtures of target spectra as well as scene endmembers. This can be defined as

$$L_{\text{leaving}} = \sum_{i}^{M} \frac{f_i(L_{\text{em},i} - L_u)}{\tau} + \sum_{j}^{N} f_j[B(T_j)\varepsilon_j - (1-\varepsilon_j)L_d], \tag{2}$$

where M is the number of endmembers, N is the number of target spectra, $L_{\text{em},i}$ is the i'th scene endmember, and f_i and f_j are the fractional abundances of each component. In this effort, the abundances are constrained such that they sum to 1 and are nonnegative.[19]

5 Calculations Occurring within Each Iteration

As described in the earlier sections, the atmospheric inference and pixel unmixing algorithms utilize iterative optimization to find an atmospheric profile that is close to optimal. At each iteration, tests are done to assess how well model parameters approximate the optimal solution. This section provides details on the calculations done within each iteration.

A key part of this approach is finding the correct material temperature in the radiometric model. Both algorithms used in the paper have separate approaches for finding this temperature for the pixels(s) under inspection. The atmospheric inference method uses an approach inspired by automatic retrieval of temperature and emissivity using spectral smoothness,[12] which determines optimal temperature by examining the smoothness of the calculated emissivity. On the other hand, the pixel unmixing algorithm uses a library-based method, where the material temperature is adjusted to find the lowest "RMS error" between the model and the measurement. The details of both methods are discussed in Secs. 5.1 and 5.2.

5.1 *Temperature Determination for Atmospheric Inference*

A key assumption used here is that most solid materials tend to have smoothly varying emissivity relative to both the sensor's spectral resolution and the spectral features of atmospheric gasses.[14] If Eq. (1) is solved for emissivity and the atmospheric parameters and material temperatures are known, then the calculated emissivity should be smooth for Lambertian materials.

The method of temperature determination used by the atmospheric inference algorithm is shown in Fig. 2. Emissivity vectors are created at a range of temperatures spanning $-30/+80$ K of the median brightness temperature at intervals of 0.1 K. This is expressed in Fig. 2 using a MATLAB™ convention for the *for* loop and interval ":" in "*for* $T = T_{\text{median}} - 30{:}0.01{:}T_{\text{median}} + 80$." A roughness calculation, that is the product of two numbers, is performed on each emissivity vector. The first number is the $\text{abs}[\text{median}(\varepsilon_T - .95)]$, which biases the metric such that high emissivity vectors are favored. The second number is calculated by downsampling the emissivity vector by a factor of 2, taking the difference (shown as "diff" in Fig. 2) along the adjacent elements of the array (analogous to a derivative), raising that vector to the fourth power (this accentuates rough spectral features caused by an incorrect atmosphere but not emissivity variation), and then taking its mean. A minimum value indicates the least rough

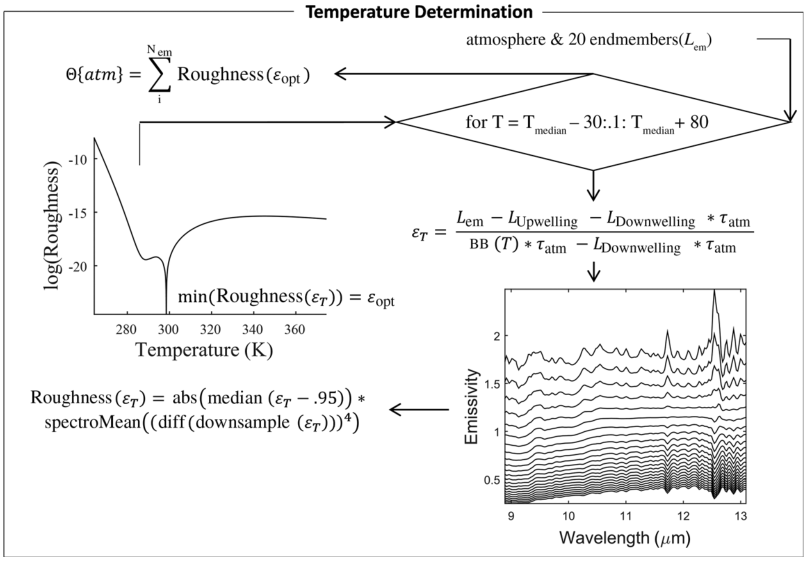
Temperature Determination
atmosphere & 20 endmembers(L_{em})
$\Theta\{atm\} = \sum_{i}^{N_{em}} \text{Roughness}(\varepsilon_{opt})$
for T = T_{median} – 30:.1: T_{median} + 80
$\varepsilon_T = \frac{L_{em} - L_{Upwelling} - L_{Downwelling} * \tau_{atm}}{BB(T) * \tau_{atm} - L_{Downwelling} * \tau_{atm}}$
log(Roughness)
-10
-15
-20
280
300
320
340
360
Temperature (K)
$\min(\text{Roughness}(\varepsilon_T)) = \varepsilon_{opt}$
$\text{Roughness}(\varepsilon_T) = \text{abs}(\text{median}(\varepsilon_T - .95)) * \text{spectroMean}((\text{diff}(\text{downsample}(\varepsilon_T)))^4)$
Emissivity
2
1.5
1
0.5
9
10
11
12
13
Wavelength (μm)

Fig. 2 This flowchart shows the temperature determination method used in the atmospheric inference algorithm. Equation (1) solved for emissivity is used to calculate emissivity spectra at a range of temperatures (shown in right hand plot). A roughness metric is used to select the ideal temperature (left hand plot) at the minimum roughness value. This is repeated for each of the 20 endmembers. The summation of roughness values is used as the cost function in the atmospheric optimization.

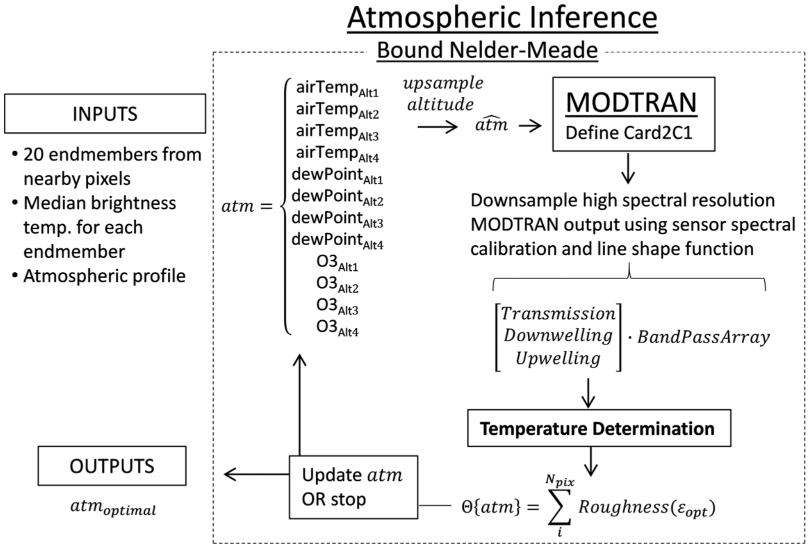
Atmospheric Inference
Bound Nelder-Meade
INPUTS
• 20 endmembers from nearby pixels
• Median brightness temp. for each endmember
• Atmospheric profile
$atm =$
airTemp$_{Alt1}$
airTemp$_{Alt2}$
airTemp$_{Alt3}$
airTemp$_{Alt4}$
dewPoint$_{Alt1}$
dewPoint$_{Alt2}$
dewPoint$_{Alt3}$
dewPoint$_{Alt4}$
O3$_{Alt1}$
O3$_{Alt2}$
O3$_{Alt3}$
O3$_{Alt4}$
upsample altitude
$\widehat{atm}$
MODTRAN
Define Card2C1
Downsample high spectral resolution MODTRAN output using sensor spectral calibration and line shape function
$\begin{bmatrix} Transmission \\ Downwelling \\ Upwelling \end{bmatrix} \cdot BandPassArray$
Temperature Determination
$\Theta\{atm\} = \sum_{i}^{N_{pix}} Roughness(\varepsilon_{opt})$
Update atm OR stop
OUTPUTS
$atm_{optimal}$

Fig. 3 This flowchart shows a high-level view of the atmospheric inference algorithm. Section 5.1 and Fig. 2 provide additional details. The importance of the algorithm to the full processing chain is examined in Sec. 7.2.

(or smoothest) emissivity that is close to 0.95. The product of these two numbers is used in the cost function for the optimization. This process is repeated for all 20 endmembers. Readers should examine Fig. 3, which shows all steps in the atmospheric inference algorithm.

This procedure is computationally fast and can be applied to many pixels, temperature ranges, and atmospheres. In a real-world implementation, it is preferable to have pixels that are at different temperatures and have different emissivities. The 20 endmembers should be collected from an area surrounding the pixel of interest. Depending on the scenario, an appropriate approach might be to define a rectangle or circle with a width of 10 m around the pixel of interest. The hope here is that local endmembers are pure versions of the background spectra found mixed into the target pixel. Ideally, they have similar emissivities, temperatures, and propagate through the same atmosphere. Endmembers can be selected using the maxD[18] algorithm, where the 20 most orthogonal pixel vectors are chosen. Users can employ other endmember selection algorithms or use a different number of endmembers as required by their scenario.

5.2 *Model-Based Temperature Determination and Pixel Unmixing*

A key component of material identification is finding the signal model that matches the measurement. This section describes a method to create radiance models from a subset of library spectra and the 20 local background endmembers. As in Sec. 5.1, this method is also a part of an iterative atmospheric optimization used to determine an optimal set of atmospheric parameters. Here, however, a material of interest is unmixed and several statistics useful for material identification are found.

Unmixing using radiometric models is a multistep process. To aid in understanding this process, readers should first refer to the flowchart in Fig. 4 and then to the flowcharts presented in this section. The first step is to obtain an initial temperature estimate of the material of interest. Using MATLAB™'s "lsqlin"[19] models with all material spectra and background endmembers are fit at a course range of temperatures ($dT = 0.5$ K). Figure 5(a) details this method. The temperature found at this step is used in the following model selection step as shown in Fig. 5(b).

The best-fit model from the initial temperature determination step will likely be overfit. Reducing the number of variables within the model may result in more reliable model statistics. This can be done using Bayesian information criteria (BIC). The definition of BIC in model fitting scenarios where the log-likelihood is being maximized is $\mathrm{BIC} = -2 \cdot \mathrm{loglik} + [\ln(N\,\mathrm{Bands})] \cdot d$,[20,21] where loglik is the log-likelihood, N Bands is the number of samples

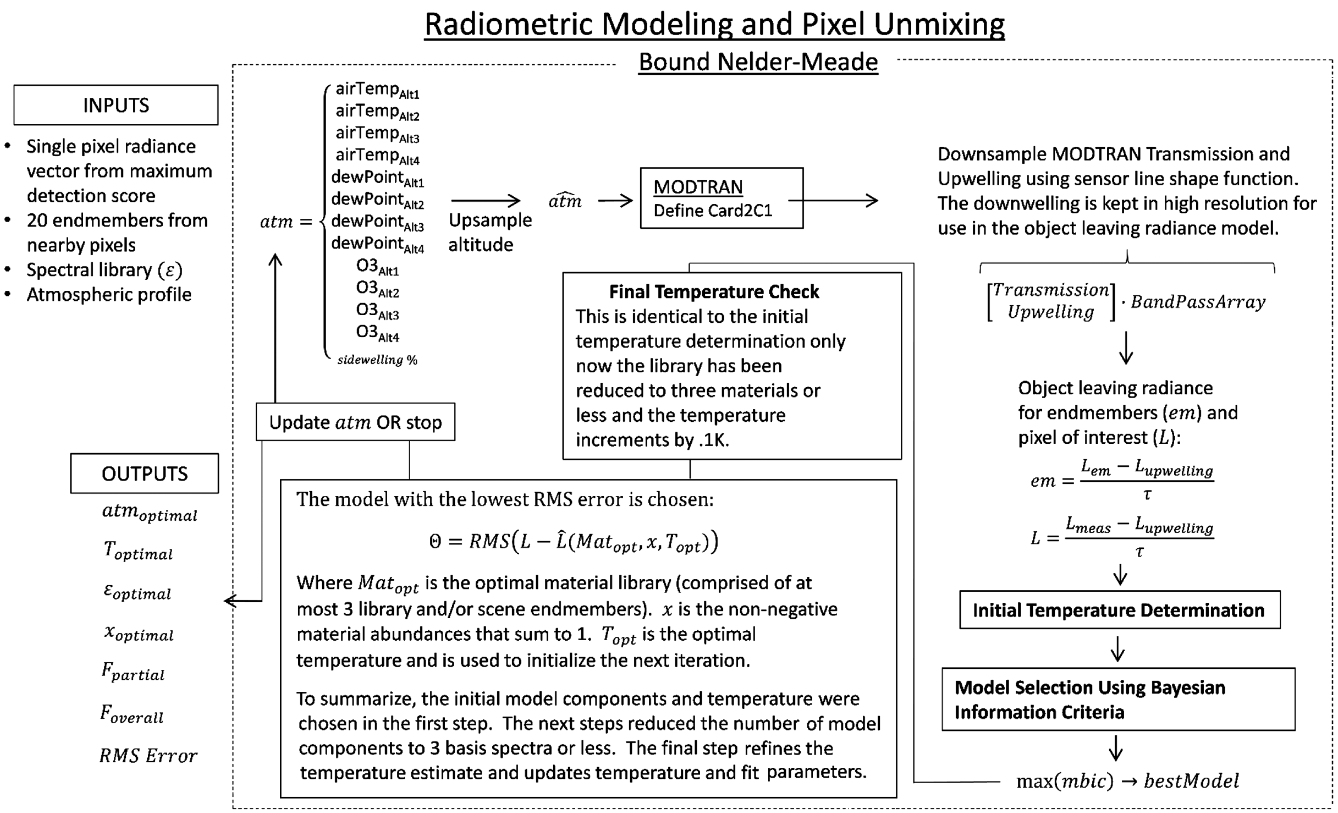
Radiometric Modeling and Pixel Unmixing
Bound Nelder-Meade
INPUTS
Single pixel radiance vector from maximum detection score
20 endmembers from nearby pixels
Spectral library (ε)
Atmospheric profile
OUTPUTS
$atm_{optimal}$
$T_{optimal}$
$\varepsilon_{optimal}$
$x_{optimal}$
$F_{partial}$
$F_{overall}$
$RMS\ Error$
$atm = \{$ airTemp$_{Alt1}$ airTemp$_{Alt2}$ airTemp$_{Alt3}$ airTemp$_{Alt4}$ dewPoint$_{Alt1}$ dewPoint$_{Alt2}$ dewPoint$_{Alt3}$ dewPoint$_{Alt4}$ O3$_{Alt1}$ O3$_{Alt2}$ O3$_{Alt3}$ O3$_{Alt4}$ $sidewelling\ \%$
Upsample altitude
$\widehat{atm}$
MODTRAN
Define Card2C1
Update atm OR stop
Downsample MODTRAN Transmission and Upwelling using sensor line shape function. The downwelling is kept in high resolution for use in the object leaving radiance model.
$\begin{bmatrix} Transmission \\ Upwelling \end{bmatrix} \cdot BandPassArray$
Object leaving radiance for endmembers (em) and pixel of interest (L):
$em = \frac{L_{em} - L_{upwelling}}{\tau}$
$L = \frac{L_{meas} - L_{upwelling}}{\tau}$
Initial Temperature Determination
Model Selection Using Bayesian Information Criteria
$\max(mbic) \rightarrow bestModel$
Final Temperature Check
This is identical to the initial temperature determination only now the library has been reduced to three materials or less and the temperature increments by .1K.
The model with the lowest RMS error is chosen:
$\Theta = RMS(L - \hat{L}(Mat_{opt}, x, T_{opt}))$
Where Mat_{opt} is the optimal material library (comprised of at most 3 library and/or scene endmembers). x is the non-negative material abundances that sum to 1. T_{opt} is the optimal temperature and is used to initialize the next iteration.
To summarize, the initial model components and temperature were chosen in the first step. The next steps reduced the number of model components to 3 basis spectra or less. The final step refines the temperature estimate and updates temperature and fit parameters.

Fig. 4 This flowchart shows a high-level view of the radiometric modeling/pixel unmixing algorithm. Additional details are provided in Sec. 5.2 and in Fig. 5.

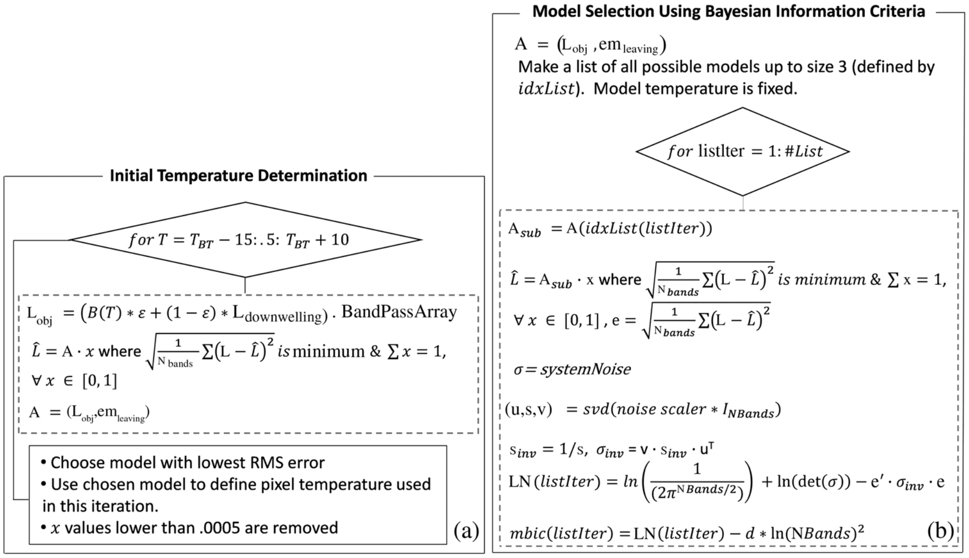
Initial Temperature Determination
$for\ T = T_{BT} - 15{:}.5{:}\ T_{BT} + 10$
$L_{obj} = (B(T) * \varepsilon + (1-\varepsilon) * L_{downwelling})$. BandPassArray
$\hat{L} = A \cdot x$ where $\sqrt{\frac{1}{N_{bands}}\sum(L-\hat{L})^2}$ is minimum & $\sum x = 1$,
$\forall x \in [0,1]$
$A = (L_{obj}, em_{leaving})$
• Choose model with lowest RMS error
• Use chosen model to define pixel temperature used in this iteration.
• x values lower than .0005 are removed
(a)
Model Selection Using Bayesian Information Criteria
$A = (L_{obj}, em_{leaving})$
Make a list of all possible models up to size 3 (defined by $idxList$). Model temperature is fixed.
for listIter $= 1{:}\#List$
$A_{sub} = A(idxList(listIter))$
$\hat{L} = A_{sub} \cdot x$ where $\sqrt{\frac{1}{N_{bands}}\sum(L-\hat{L})^2}$ $is\ minimum$ & $\sum x = 1$,
$\forall x \in [0,1]$, $e = \sqrt{\frac{1}{N_{bands}}\sum(L-\hat{L})^2}$
$\sigma = systemNoise$
$(u,s,v) = svd(noise\ scaler * I_{NBands})$
$s_{inv} = 1/s$, $\sigma_{inv} = v \cdot s_{inv} \cdot u^T$
$LN(listIter) = ln\left(\frac{1}{(2\pi^{NBands/2})}\right) + \ln(\det(\sigma)) - e' \cdot \sigma_{inv} \cdot e$
$mbic(listIter) = LN(listIter) - d * \ln(NBands)^2$
(b)

Fig. 5 (a) A method for determining model constituents and material temperature is provided here. A nonnegativity and sum to 1 constraint is placed on the abundances of the model. Referring to Eq. (2), $\mathrm{em}_{\mathrm{leaving}}$ has dimension [$M \times$ numbands] and L_{obj} has dimension [$N \times$ numbands]. (b) With material temperature held static, a search of models is performed. A modified version of BIC that more strongly preferences simpler models is used. The maximum *mbic* value is chosen as the best model.

(number of bands), and d is the number of variables ($M + N$). Including the number of variables in a summation term has a regularizing effect. A modified version of this equation (*mbic*) is used here for model selection [see Fig. 5(b)]. The primary modification is that the number of variables, d, is multiplied by the squared term $\ln(N\,\mathrm{Bands})^2$, which results in an increased preference for simple models.

The reduced model is then used in a final temperature determination step. This step is identical to the initial temperature determination except that a finer temperature increment is used ($\mathrm{d}T = 0.1$ K). An important implementation note is the final temperature found here is then used as an initialization point for the next iteration of atmospheric optimization.

Readers should note that while it is possible for the algorithm to use multiple material spectra during unmixing, it is not possible to use multiple material temperatures. For many remote sensing scenarios, materials will be at a uniform temperature. Small variations in material temperature produce approximately linear variations in radiance; therefore, the average radiance value captured of subpixel temperature variability is still likely to be useful—this will depend on the magnitude of variability and noise of the system. These issues will not be investigated here.

6 Logistic Regression for Spectral Identification

Traditional detection algorithms require a user-defined threshold to establish which pixels are presented to the analysts. This is also possible with identification algorithms. A useful parameter might simply be the RMS error between the model and measurement. Identification algorithms also offer a different approach, where information can come from multiple sources. A spectral analyst might find it useful to inspect several numerical values before deciding whether a material is present. The detection score (such as the adaptive cosine estimator),[18] RMS error, overall F-statistic, partial F-statistic,[22] target material temperature, number of target materials, and target abundances may all be useful to the analyst. If the analyst views identification results from many detections, patterns might appear that would allow for usage of multiple parameters. One option is to use the fit statistics from known true and false positives to train a logistic regression algorithm[20,21] to define a decision surface that optimally separates the true and false positives. Once the surface is established, future identification results can be tested against this decision surface to determine whether the detection is a true or false

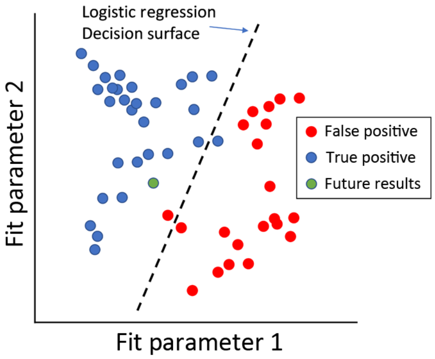

Fig. 6 This is an illustration of logistic regression used to optimally adjust identification thresholds. Each data point represents a pixel that passed the detection threshold and was subsequently processed by the identification chain. The true and false positives are used to train the logistic regression decision surface. The future identification result (green) will be classified by the decision surface; in this case, it will be a true positive.

positive. A two-dimensional illustration of this is provided in Fig. 6. The parameters shown here could be "determined target abundance" versus RMS error, which would reflect that true positive pixels will likely have low RMS error models and high target abundance. Adding additional information, such as partial F-statistics, model size, temperature, etc., can increase the ability of this method to separate true versus false positive detections.

The author has found this to be a powerful approach for eliminating large numbers of false positives in real datasets. If readers choose to employ this method, caution should be taken when using data from multiple sensors. A decision surface established using data from one sensor may not be useful for analyzing data from other sensors with different noise characters. Results using the simulated data will be provided in Sec. 7.6.

7 Simulations and Results

Using synthetic data makes it possible to create and test algorithms under many different conditions. Presenting the processing chain with challenging conditions, such as increased noise or low target abundance, allows users to understand the limitations of the processing chain. This section will present a variety of tests allowing users to understand how well the processing chain determines abundance, target material temperature, and the overall error when compared to the measurement. Simulated false positive detections will also be examined with the algorithm, and their results will be compared to those of the simulated true positives.

NASA's ASTER spectral library has reflectance measurements of many common materials. The material types used here (limestones and roofing materials) were chosen because they are common, have unique spectral features, and do not suggest favoritism toward any applications (mining, defense, etc.). The simulations in this paper utilize:

jhu.becknic.rock.sedimentary.limestone.coarse.limest1.spectrum and
jhu.becknic.manmade.roofing.rubber.solid.0833uuu.spectrum

as the surrogate target and background material, respectively [see Fig. 7(a)]. All true positive pixels are modeled using mixtures of these two materials at different abundances and temperatures. The input target spectral library used in unmixing includes additional ASTER limestones [see Fig. 7(c)] with 11.4-μm features:

jhu.becknic.rock.sedimentary.limestone.coarse.limest1.spectrum,
jhu.becknic.rock.sedimentary.limestone.coarse.limest2.spectrum,
jhu.becknic.rock.sedimentary.limestone.coarse.limest3.spectrum,
jhu.becknic.rock.sedimentary.limestone.coarse.limest4.spectrum,
jhu.becknic.rock.sedimentary.limestone.coarse.limest5.spectrum, and
jpl.nicolet.rock.sedimentary.limestone.solid.fge3.spectrum.

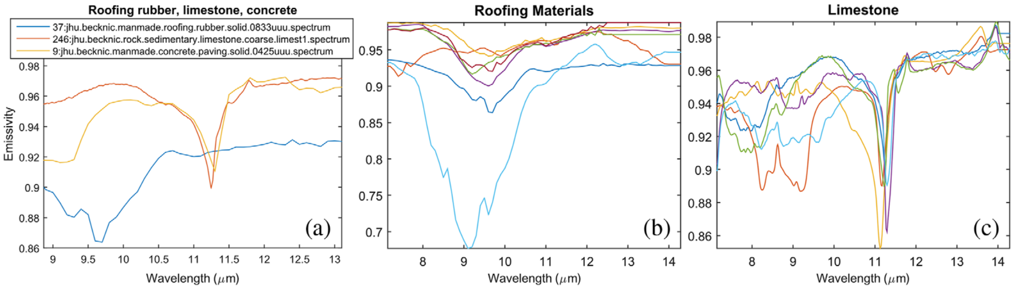

Fig. 7 Plots of spectra used in the experiments are shown here. (a) The limestone and rubber roofing materials used in the pixels of interest are shown here. Also shown is the concrete confuser used in Sec. 7.6. Note the wavelength is bound between 8.86 and 13.1 μm, the spectral region used in identification. Plots (b) and (c) show the background spectra and target (limestone) library spectra, respectively.

The background endmember pixels were created by selecting other common roofing materials also in the ASTER library [see Fig. 7(b)]:

jhu.becknic.manmade.roofing.rubber.solid.0833uuu.spectrum,
jhu.becknic.manmade.roofing.rubber.solid.0834uuu.spectrum,
jhu.becknic.manmade.roofing.shingle.solid.0490uuu.spectrum,
jhu.becknic.manmade.roofing.shingle.solid.0597uuu.spectrum,
jhu.becknic.manmade.roofing.shingle.solid.0672uuu.spectrum,
jhu.becknic.manmade.roofing.shingle.solid.0680uuu.spectrum, and
jhu.becknic.manmade.roofing.shingle.solid.0683uuu.spectrum.

In Sec. 7.6, a confuser concrete material is modeled. The emissivity spectrum for this material is shown in Fig. 7(a).

The 20 background spectra were modeled using a random material temperature selected from a Gaussian distribution with mean 30°C and standard deviation 2.5°C. Target temperatures will be specified in each simulation.

In the following sections, scene radiance spectra are modeled with the middle-latitude summer standard atmosphere, sensor altitude at 5.5 km above sea level, and 4.5 km above ground level.

Ideally, an analyst would like to see a model with low RMS error, realistic temperature, target material abundances above ∼0.3 if it is the only target predictor variable in the model, and high partial F-statistic. Plots of ground leaving radiance, model minus measurement residuals, and estimated emissivity can also be useful to analysts viewing real data.

7.1 *Impact of System Noise on Model Parameters*

This section examines the effect of additive Gaussian noise and target material abundances on several fit parameters. The radiance in the scene measurement was modeled using:

jhu.becknic.rock.sedimentary.limestone.coarse.limest1.spectrum and
jhu.becknic.manmade.roofing.rubber.solid.0833uuu.spectrum

at 25°C and 30°C, respectively. Five sets (a set is 10 scene pixels and 20 endmembers) were created where the pixel of interest was modeled with abundance values of 0.1 to 1 in increments of 0.1. At five additive Gaussian noise levels (standard deviation of 0, 0.5, 1, 1.5, and 2 μF), this amounts to 150 spectra total. The limestone spectrum has a spectral emissivity feature of ∼7% at 11.25 μm. This feature at 25°C and fractional abundance of 0.1 should be about 4 μF in depth prior to atmospheric attenuation.

The full pipeline was applied to all datasets. Figure 8 shows a summary of the algorithm's performance for these tests. The first column shows the RMS error in μF between the modeled and measured radiances. The second column shows the determined target fractional abundance from the radiance model. The third column shows the determined temperature. The dashed lines in the second and third columns are the true values, and deviations from these values are errors.

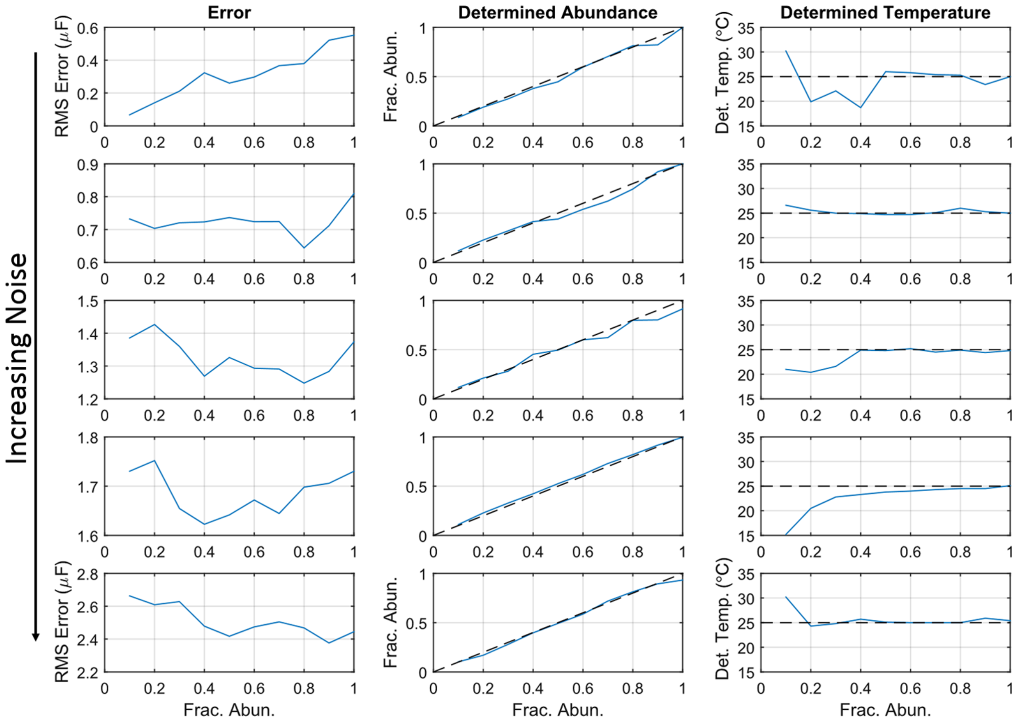

Fig. 8 This shows a demonstration of algorithm performance with additive Gaussian noise with standard deviation of 0, 0.5, 1, 1.5, and 2 μF. Performance is consistent even at higher levels of noise.

Each of the five rows pertains to a different noise level $(0, 0.5, \ldots 2\ \mu\text{F})$. The horizontal axis of each plot is the fractional abundance of the limestone spectra in each modeled target vector. The plots show some disagreement in the RMS error between modeled and measured spectra in the 0-μF noise data. This is expected as the optimized atmospheric profile will not match the original profile and that will be reflected as error in fit and parameter estimates. Other contributing effects are found in the values for the "determined fractional abundance" and the "determined temperature." At higher target abundances, the error consistently increases. It is difficult to know exactly why but one likely explanation is that at lower target abundances (higher background abundances) the background endmembers can explain more of the variance, as they have a variety of temperatures and spectral shapes. The algorithm shows good matching between the modeled and predicted abundance (<10% error). This behavior might change if there is a large discrepancy between predicted and real temperatures, for example, if a predicted temperature is low, a higher abundance might produce a best-fit model as it would compensate for the low temperature. Temperature estimations are between 24°C and 26°C for the models with abundances >50% at all noise levels, with one exception at the zero noise 0.9 abundance level which was 23.4°C. All target fractional abundance estimations are within 0.1 of their true values for all models.

7.2 *Importance of Atmospheric Inference*

Given the number of steps in this processing chain, it is useful to have an understanding of the importance of each step to overall performance. The pixel unmixing can be run without estimating the atmosphere with the atmospheric inference algorithm. The results shown in Fig. 9 utilized the same zero-noise dataset from the first row of Fig. 8. Here, we see that omitting the atmospheric inference causes a reduction in the accuracy of the determined temperature and abundance, as well as a dramatic increase in RMS error. The atmospheric inference algorithm is, therefore, helpful in the retrieval of abundance and temperature and creating low error models.

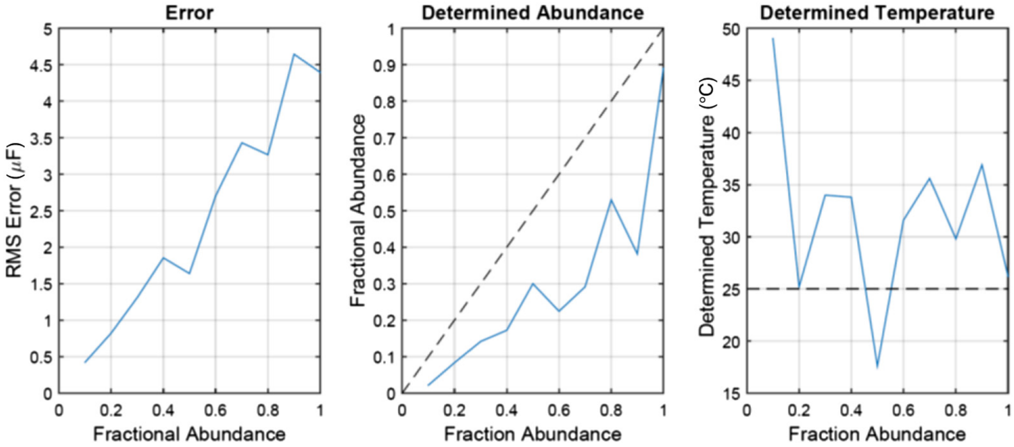

Fig. 9 Retrieval of model abundance and temperature is negatively affected by omitting the atmospheric inference algorithm.

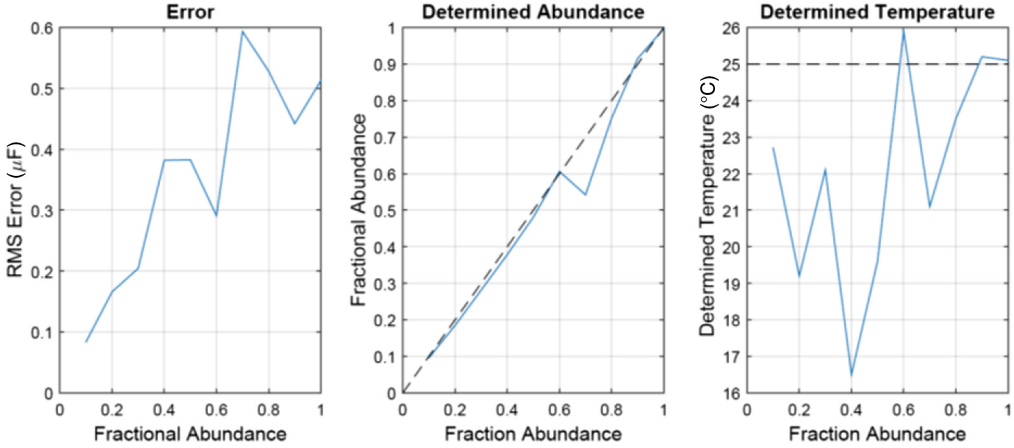

Fig. 10 Effects of omitting calculations specified in rows 2 and 3 of Table 1.

7.3 *Demonstration of Initial Atmospheric Search*

The pixel unmixing calculations specified in the second and third row of Table 1 can be omitted from the processing chain. The reduced pipeline is run on the zero-noise dataset. The results are shown in Fig. 10. The RMS error values for the model are not substantially higher. The determined abundance values are also quite reasonable. The determined model temperature shows much more error than when the full processing chain is used (compare with Sec. 7.1). The calculations defined in rows 2 and 3 of Table 1 act as priming steps for the last run of the algorithm (row 4) to help with determining correct material temperatures.

7.4 *Examining Effects of Object Temperature Relative to Atmospheric Temperature*

One could imagine scenarios where the object temperature is either above or below the ground atmosphere temperature. A simulation was conducted where the limestone target temperature and rubber roof background temperature were increased from 8°C and 13°C to 35°C and 40°C, respectively, in increments of 3°C. The purpose of the simulation was to monitor the behavior of the algorithm as object temperature changed relative to a static atmospheric profile. This simulation used zero-noise data with target fractional abundance of 0.6. The plots in Fig. 11 show that an object having low temperature (8°C) relative to the atmospheric temperature (∼18°C) could create a model with high RMS error. In the low-temperature case, the determined abundance is

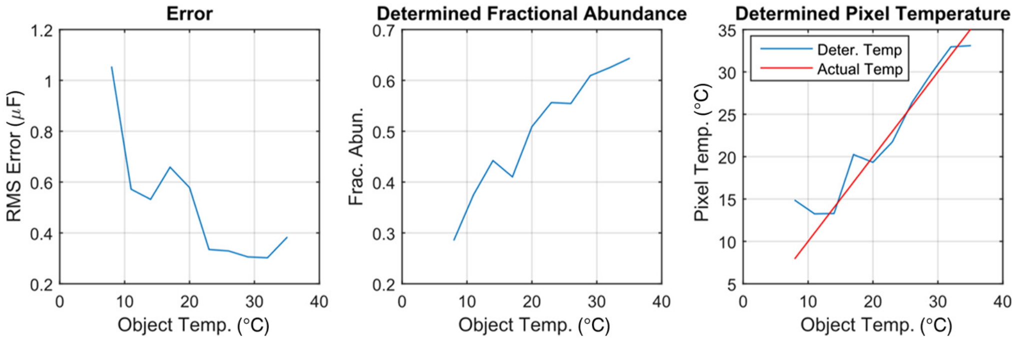

Fig. 11 The lowest temperatures here show higher error rates and larger deviations in determined fractional abundance compared to results shown in Fig. 8.

significantly less than it should be, and the determined temperature has high error. The behavior at higher temperatures resembles results observed in previous simulations.

7.5 *Atmospheric Profile Retrieval*

As described in Sec. 3, the algorithms used here attempt to find optimal atmospheric profiles. This section examines the temperature and humidity profile estimates for the zero-noise dataset with fractional target abundance of 0.6. The results in Fig. 12(a) show little agreement between the determined temperature/dew point profile and that of the middle-summer latitude. The test is repeated using a narrow boundary in the Nelder–Meade optimization at the aircraft altitude—simulating usage of a temperature/humidity sensor onboard the aircraft. The results from this test in Fig. 12(b) show more agreement to the middle-summer latitude atmosphere. However, a degree of error remains. A test for future work using just three altitudes in the atmospheric optimization might show improved results as there are fewer local minima. The author uses four altitudes to guard against atmospheres with strong boundary layer conditions.

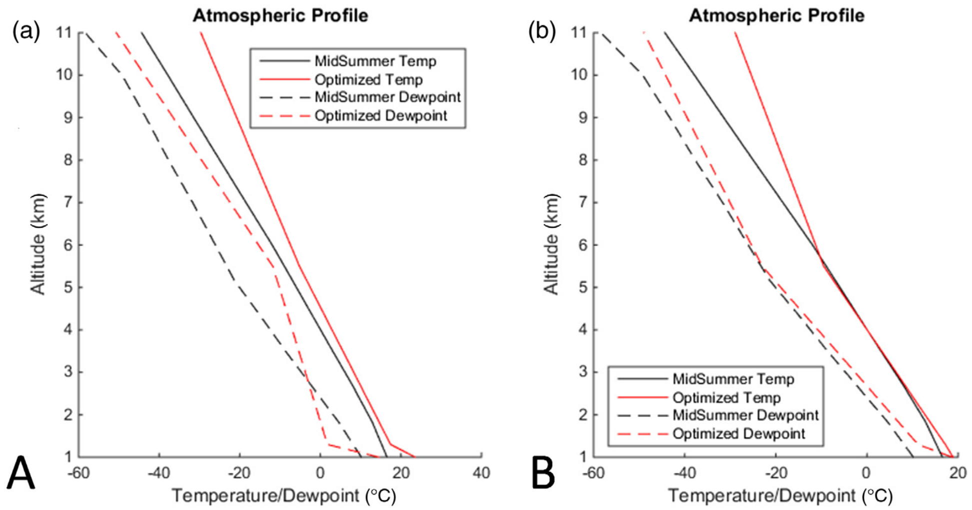

Fig. 12 (a) As can be seen, there is a significant amount of error in this retrieval. The dew point estimate has errors of over 10°C. Examining the dew point closely, there is an undershoot at low altitudes and overshoot at higher altitudes. Therefore, the overall effect here is likely similar to closer approximations of the true atmosphere. This is an example of the algorithm getting trapped in a local minima. (b) There is a slight improvement made by placing a temperature/humidity sensor onboard the aircraft.

7.6 *Examination of Outputs for True and False Positive Detections*

Material detection steps often produce false positive rates $>10^{-5}$.[18,23] The reason for false alarming can be related to the statistical whitening process used in many detection algorithms. Reprocessing the data in radiance space using algorithms, such as the one described in this paper, will allow users to produce physics-based statistics and parameters that can be used to further suppress false positives. This section shows how multiple parameters can be used together to improve system performance.

Four sets of spectra were created:

1. 20 spectra, with abundances of 70% limestone materials (mean temperature 25°C, std 1) and 30% rubber roofing materials (temperature 30°C),
2. 20 spectra, with abundances of 40% limestone materials (mean temperature 25°C, std 1) and 60% rubber roofing materials (temperature 30°C),
3. 20 spectra, with abundances of 70% nonlimestone materials (mean temperature 25°C, std 1) and 30% rubber roofing materials (temperature 30°C), and
4. 20 spectra, with abundances of 40% nonlimestone materials (mean temperature 25°C, std 1) and 60% rubber roofing materials (temperature 30°C).

The nonlimestone materials were comprised of 20 randomly selected materials from the ASTER library. The only nonlimestone material that shared common spectral features to the limestone was jhu.becknic.manmade.concrete.paving.solid.0425uuu.spectrum [see Fig. 7(a)]. Using these 80 spectra, separate low- and high-noise datasets were created using 0.5- and 1.5-μF Gaussian noise, producing a total of 160 spectra.

The full processing chain was used to gather fit results. For many of the nonlimestone materials, the RMS error was very high. The most interesting results occurred when the algorithm processed a false positive that produced a low RMS error. In scenarios such as this, a threshold on the RMS error alone will not be enough to suppress all the false positives.

Figure 13(a) contains the results for the low-noise dataset. Here, we see a near-perfect separation between limestone and nonlimestone pixels when using the RMS error. If the determined target abundance is included, the two groups can be linearly separated. The logistic regression line is drawn as a dashed line here. The two circled nonlimestone results belong to pixels containing the concrete spectrum. Concrete is similar to the limestone but is not a perfect match and is, therefore, still separable. At the higher 1.5-μF noise level, as shown in Fig. 13(b), the concrete pixels are not separable when using the RMS error and determined target abundance.

Plotting fit parameters in two dimensions shows that true and false positives can be separated using linear logistic regression. Using additional fit data to extend the logistic regression to higher dimensions will likely help with separability, possibly separating the concrete. An in-depth discussion of logistic regression has been left out of this paper as there are many resources

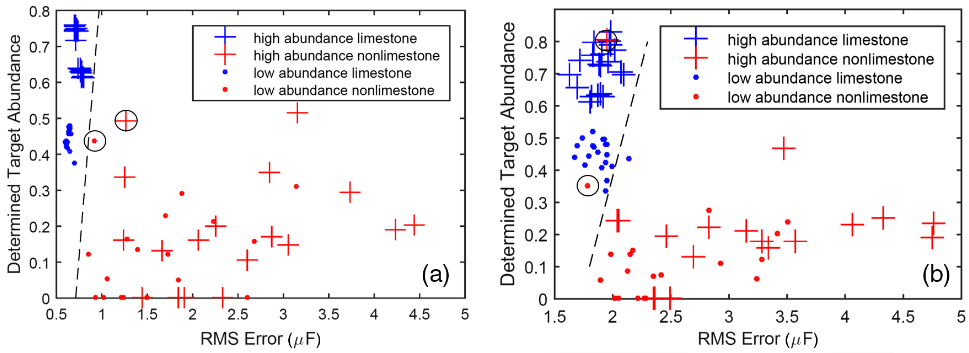

Fig. 13 (a) The results from the low-noise (0.5 μF) data have a clear separation between all target and nontarget materials. (b) The high-noise (1.5 μF) data do not have a clear separation between the target and nontarget materials. The circled red data points belong to spectra modeled with concrete. The dashed line is a logistic regression line that creates a decision surface between the two classes.

available on this topic.[20] General machine learning principles (class balancing, training/test/validation datasets, etc.) should be followed.

8 Conclusion

This paper describes an LWIR HSI solid material identification method. The overall assumption of this approach is that accurate radiometric models can be created for pixels of interest using mixtures of local endmembers and forward modeled library spectra. If a pixel of interest contains the materials found in the model, then the fit between the measurement and model will have an RMS error approaching the instrument noise, and the fit parameters will be close to their true values. If the pixel does not contain the materials found in the model, then the fit between the measurement and model will have a high RMS error, and the model parameters might be unrealistic. False positives can be reduced or eliminated using parameters and statistics from the pixel unmixing step, as input to a logistic regression algorithm that was trained on a known set of true and false positives. These statements are dependent upon the list of assumptions found in Sec. 1.

The simulations provided in Sec. 7 demonstrate the behavior of the processing chain under a variety of conditions. Section 7.1 shows a consistent rise in model error with system noise. For most pixels with target abundance >0.5, the object temperature can be retrieved to within 1°C of the true temperature and abundances can be retrieved to within 0.1 of their true value. Sections 7.2 and 7.3 demonstrate the importance of the atmospheric inference and the initialization runs of the unmixing algorithm. Section 7.4 examines scenarios where the ground is either hotter or cooler than the surrounding air temperature. The algorithm appears to work best when ground temperatures are greater than or equal to the air temperature. When ground temperatures are <10°C than that of the air temperature, the performance drops off significantly. Section 7.5 demonstrates that this pipeline cannot accurately retrieve the atmospheric temperature/humidity profile. But if a temperature/humidity sensor is placed on the aircraft, some improvement can be made in the retrieval of the atmospheric parameters. Section 7.6 demonstrates how multiple fit parameters from the pipeline can be used to separate true and false positives from the detection step. In one case, the false positive material was concrete, which has strong limestone spectral features. At a 0.5-μF noise level, the algorithm was capable of discerning between the limestone and concrete.

An alternate approach to this study would have been to use a real LWIR HSI remote sensing dataset, with in-scene targets, which could allow for characterization the algorithm's performance using receiver operator characteristic curves. In the VNIRSWIR remote sensing community, this is common as datasets from many studies are freely downloadable. An example is the RIT target detection blind test[24] where scientists can benchmark their algorithm's performance. Open-source datasets collected from an airborne platform by a dispersive optic LWIR HSI system are not common. The author chose the modeling approach used here because it is easily reproducible. This approach also allowed for well-controlled simulations where all data parameters were known.

This work was performed on a Dell Precision with an Intel i7 processor. No GPUs were utilized. The algorithm requires about 20 min to process a single spectral vector. During the atmospheric optimization process, atmospheres are saved in a database for subsequent use. Because of the time requirements, this algorithm is not practical for real-time data processing in size, weight, and power-limited environments, such as onboard an aircraft. However, because the required amount of data for identification is only 21-pixel vectors (1 pixel of interest and 20 endmembers), the data could be rapidly transmitted to a computer cluster for parallel processing. There is also additional algorithmic work that could be done to speed up the search for an optimal atmospheric profile; for example, numerical approximations that can be used on a precalculated database of atmospheres to speed up the atmospheric inference and pixel unmixing algorithms. In real datasets collected over a local area, atmospheric conditions in one location could be correlated with atmospheric conditions in another location. In this scenario, users could employ the atmospheric model found for one identification result on another local identification result, either as an initialization for optimization or directly to the pixel. It is also possible to optimize an atmosphere for multiple target pixels simultaneously by summing the pixels' RMS model error values. If users choose this route, care should be taken to ensure that atmospheric conditions are suitable (uniform), as there are scenarios where the atmosphere could be nonuniform over short distances.

References

1. J. A. Hackwell et al., "LWIR/MWIR imaging hyperspectral sensor for airborne and ground-based remote sensing," *Proc. SPIE* **2819**, 102–107 (1996).
2. W. R. Johnson et al., "HyTES: thermal imaging spectrometer development," in *Aerospace Conf.*, IEEE (2011).
3. N. B. Gallagher, B. M. Wise, and D. M. Sheen, "Estimation of trace vapor concentration-pathlength in plumes for remote sensing application from hyperspectral images," *Anal. Chim. Acta* **490**, 139–152 (2003).
4. C. C. Borel, "Iterative retrieval of surface emissivity and temperature for a hyperspectral sensor," in *Proc. for the First JPL Workshop on Remote Sensing of Land Surface Emissivity* (1997).
5. C. C. Borel, "Surface emissivity and temperature retrieval for a hyperspectral sensor," in *IEEE Int. Geoscience and Remote Sensing Symp. Proc.*, Seattle, Washington (1998).
6. D. Manolakis, "Taxonomy of detection algorithms for hyperspectral imaging applications," *Opt. Eng.* **44**(6), 066403 (2005).
7. B. Basener et al., "A detection-identification process with geometric target detection and subpixel spectral visualization," in *3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)* (2011).
8. P. V. Villeneuve, A. R. Boisvert, and A. D. Stocker, "Hyperspectral sub-pixel target identification using least angle regression," *Proc. SPIE* **7695**, 76951V (2010).
9. S. J. Young, R. B. Johnson, and J. A. Hackwell, "An in-scene method for atmospheric compensation of thermal hyperspectral data," *J. Geophys. Res.* **107**(D24), 4774–4793 (2002).
10. M. Cubero-Castan et al., "A physics-based unmixing method to estimate subpixel temperatures on mixed pixels," *IEEE Trans. Geosci. Remote Sens.* **53**(4), 1894–1906 (2015).
11. A. Berk et al., *MODTRAN(TM)-5 Version 3 Revision 4 User's Manual*, Spectral Sciences, Inc., Burlington, Massachusetts (2005).
12. C. Borel, "ARTEMISS—an algorithm to retrieve temperature and emissivity from hyper-spectral thermal imaging data," in *GOMACTech, 28th Annual Conf., Hyperspectral Imaging Session*, Tampa, Florida, LA-UR-027907 (2003).
13. M. Boonmee, "Land surface temperature and emissivity retrieval from thermal infrared hyperspectral imagery," PhD Thesis, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology (2007).
14. C. C. Borel and R. F. Tuttle, "Recent advances in temperature-emissivity separation algorithms," in *IEEE Aerospace Conf.*, Big Sky, Montana, pp. 1–14 (2011).
15. D. J. Griffith, *matlab-modtran-5*, https://www.mathworks.com/matlabcentral/fileexchange/31961-derekjgriffith-matlab-modtran-5 (25 June 2011).
16. W. H. Press et al., *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, Cambridge (2007).
17. J. D'Errico, *fminsearchbnd, fminsearchcon*, https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd--fminsearchcon (22 September 2015).
18. J. R. Schott, *Remote Sensing: The Image Chain Approach*, 2nd ed., Oxford University Press, Oxford, New York (2007).
19. "lsqlin," MATLAB Optimization Toolbox, Version 7.1, https://www.mathworks.com/help/optim/ug/lsqlin.html.
20. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, New York (2009).
21. C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York (2006).
22. M. H. Kutner et al., *Applied Linear Statistical Models*, 5th ed., McGraw-Hill Irwin, Boston, Massachusetts (2005).
23. D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Lab. J.* **14**, 1 (2003).
24. D. Snyder et al., "Development of a web-based application to evaluate target finding algorithms," in *IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS)* (2008).

Michael E. Zelinski graduated with a BS degree in imaging science and MS degree in remote sensing from the Rochester Institute of Technology, Rochester, New York, in 2006 and 2009, respectively. Currently, he works with the Advanced Exploitation Group, Lawrence Livermore National Laboratory, Livermore, California. His research interests include Fourier optics, hyperspectral remote sensing, image system modeling and characterization, image processing, machine learning, and deep learning.