Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Distributed adaptive framework for multispectral/hyperspectral imagery and three-dimensional point cloud fusion

Robert S. Rand
Timothy Khuon
Eric Truslow

SPIE.

Robert S. Rand, Timothy Khuon, Eric Truslow, "Distributed adaptive framework for multispectral/hyperspectral imagery and three-dimensional point cloud fusion," *Opt. Eng.* **55**(7), 073101 (2016), doi: 10.1117/1.OE.55.7.073101.

Distributed adaptive framework for multispectral/ hyperspectral imagery and three-dimensional point cloud fusion

Robert S. Rand,[a,*] **Timothy Khuon,**[a] **and Eric Truslow**[b]
[a]National Geospatial-Intelligence Agency, 7500 GEOINT Drive, Springfield, Virginia 22150, United States
[b]Northeastern University, Department of Electrical and Computer Engineering, 360 Huntington Avenue, Boston, Massachusetts 02115, United States

Abstract. A proposed framework using spectral and spatial information is introduced for neural net multisensor data fusion. This consists of a set of independent-sensor neural nets, one for each sensor (type of data), coupled to a fusion net. The neural net of each sensor is trained from a representative data set of the particular sensor to map to a hypothesis space output. The decision outputs from the sensor nets are used to train the fusion net to an overall decision. During the initial processing, three-dimensional (3-D) point cloud data (PCD) are segmented using a multidimensional mean-shift algorithm into clustered objects. Concurrently, multiband spectral imagery data (multispectral or hyperspectral) are spectrally segmented by the stochastic expectation–maximization into a cluster map containing (spectral-based) pixel classes. For the proposed sensor fusion, spatial detections and spectral detections complement each other. They are fused into final detections by a cascaded neural network, which consists of two levels of neural nets. The success of the approach in utilizing sensor synergism for an enhanced classification is demonstrated for the specific case of classifying hyperspectral imagery and PCD extracted from LIDAR, obtained from an airborne data collection over the campus of University of Southern Mississippi, Gulfport, Mississippi. [DOI: 10.1117/1.OE.55.7.073101]

Keywords: spectral–spatial fusion; hyperspectral imagery; point cloud data; LIDAR; neural networks; stochastic expectation–maximization; mean shift.

Paper 151622P received Nov. 25, 2015; accepted for publication Jun. 20, 2016; published online Jul. 19, 2016.

1 Introduction

Multiband spectral data (multispectral or hyperspectral) and point cloud data (PCD) are typically processed and interpreted separately from each other. These data types are seldom combined for purposes of scene segmentation, detection, classification, or object recognition. For low-dimensional spectral data such as multispectral imagery (MSI), a scene can be segmented, e.g., using the stochastic expectation–maximization (SEM) algorithm.[1] For high-dimensional data, such as hyperspectral imagery (HSI), many methods for spectral segmentation, detection, and classification have been investigated, including[2–13] to determine the class of a two-dimensional (2-D) pixel, based on its spectral signature. Given the large number of bands and consequent sparse information existing in HSI, dimension reduction methods are sometimes applied to the data, such as by using principal component analysis (PCA).[13] Unfortunately, spectral-based segmentation and classification methods respond to material types, not object shapes, and so this approach by itself is often not sufficient to effectively segment and classify a scene into objects.

Much progress has been made in recent years to incorporate spatial information into the process. More sophisticated methods of dimension reduction have been proposed, such as an approach based on tensor modeling[14] that seeks to retain spatial information that is otherwise missing in traditional PCA, as well as another method that makes use of local Fisher's discriminant analysis that reduces dimensionality but preserves its multimodal structure.[15] A number of spectral–spatial segmentation and classification methods have been proposed, including spectral–spatial methods that use a support tensor machine (STM),[16] spectral–spatial methods based on Markov random fields (MRFs),[17–20] spectral and spatial classification using support vector machines and morphological profiles,[21] a partitional clustering technique involving pixel wise SVM and spatial postregularization,[22] evidence theory using Dempster's rule on joint spatial–spectral features,[23] and unsupervised deep learning with convolutional neural networks applied to HSI.[24] Conversely, research on spatial classification[25,26] can be used to determine the class of a group of points based on its shape, size, position, and intensity, but it cannot determine the material (subclass) of each pixel.

The goal of this research is to investigate an approach that can overcome some of the limitations arising from using only one data modality and achieve improved classification results using the spatial and spectral fusion of information from multisensor data, specifically PCD and multiband imagery (MSI or HSI).

Our proposed method for spectral–spatial classification is a framework that differs from the spectral–spatial methods mentioned above,[16–24] because it includes the use of three-dimensional (3-D) information from PCD, coming from, say, a LIDAR instrument. Without such information segmentation and classification is fundamentally limited to separate or identify objects based on spectral information in the case

[*]Address all correspondence to: Robert S. Rand, E-mail: Robert.S.Rand@nga.mil

Domestic imagery was collected in accordance with a valid Proper Use Memorandum approved by NGA, and is not directed at U.S. persons.

of Refs. 1–15 or spectral–spatial (2-D shape) information in the case of Refs. 16–24. Consequently, without such information, it is often difficult to determine its super class or object class. The proposed framework makes use of a spatial classification method based on Refs. 25 and 26 that determines the class of a group of points based on its shape, size, position, and intensity.

Our approach is to segment and classify objects based on both material composition and 3-D shape information. Spatial and spectral classification results that have been derived independently from PCD and spectral multiband imagery are conflated and conflicts between the decisions made by the independent processes are resolved. A neural-based approach is proposed to conflate classification information from PCD and MSI/HSI multiband data. The front-end algorithms in the framework are unsupervised; thus algorithm training by an analyst is not required until the final stage of the process. An underlying assumption is made that the PCD and MSI/HSI data sets are registered to a common ground reference frame. Our hypothesis is that the proposed framework is capable of generating a classification map that delineates objects based on similarities in 3-D geometry and material composition, while simultaneously reducing false alarms by resolving the geometric and spectral ambiguities that often occur when using a single source of imagery.

Our experiment considers the specialized case of PCD extracted from LIDAR and spectral data from HSI. The approach is demonstrated using data from an unclassified airborne collection of LIDAR and HSI data in November 2010 over a college campus in Gulfport, Mississippi. During this campaign, LIDAR and HSI sensors were mounted on the same airplane. Consequently, the registration process was relatively simple.

We mention a few comments about terminology: for purposes of this paper, a number of terms are defined in this section. An "object" is a set of pixels in an image that belong together and represent a single physical entity. This object could be composed of one material or be an aggregate of multiple materials. An "unsupervised process" requires no human intervention for training an algorithm, whereas a "supervised process" requires intervention. The supervised process is trained by a human analyst such that the labels associated with the regions are typically associated with classes of materials or objects that are of interest to an analyst. "Segmentation" is an unsupervised process that partitions an image into multiple segments according to statistical properties of the data. During the segmentation process, sets of pixels are typically assigned a numeric label, but these labels need not represent a particular type of object or material, and have yet to be assigned to a class. We use the term "clustering" interchangeably with this process, whenever we are attempting to find the structure in data and organize these data into groups whose members are similar in some way, where each group is called a "cluster." We define the term "classification" as a supervised process to segment an image into regions that are assigned labels associated with an object or material class. This process could be implemented algorithmically by one of many decision-theoretical approaches, or manually by an analyst who assigns classes to the labels provided by a segmentation process. Throughout this paper, "distributed system" refers to events within the system that are happening concurrently and independently, possibly replicating or having different functionalities, possibly asynchronous, diversely interconnected; "adaptive system" referred to an ability to adjust the system by means of incremental training to support as well as to reject previously learning.

2 Description of Algorithms

The proposed framework is implemented through three processes: spatial segmentation of 3-D PCD, spectral segmentation of multiband (MSI or HSI) data, and spectral/spatial fusion of the segmentation results. In the first process, PCD is segmented based on the multidimensional mean-shift segmentation algorithm, where the results are clustered based on the geometry of points in the point cloud. In the second process, any of a number of unsupervised spectral-only[1,7–12] or spectral–spatial methods[16–24] could be applied to HSI data to segment the data into a classification map of material compositions based on spectral information. We select the SEM as our choice algorithm to spectrally segment the data, because it is theoretically rigorous and more sophisticated than older segmentation methods, such as k-means[13] or ISODATA,[13] and because it can also especially be applied to MSI. The choice of SEM is also a conservative gesture, because we would expect results on HSI to improve with some of the recently introduced spectral–spatial methods.[16–24] Note that the first two processes are independent and can be implemented in parallel.

The SEM algorithm works best in a low-dimensional space;[1–4] therefore, a dimension reduction method is important whenever HSI is the source imagery. We selected PCA[13] as our method to project the most significant information in the high-dimensional HSI space into a lower dimensional space. This process results in a smaller multiband image as input to the SEM that contains the majority of information in the original HSI image. Alternative newer dimension reduction methods[14,15] could have been used, but we chose PCA because of its overall acceptance in the community and widespread availability.

In the third process, spectral/spatial fusion, the spatial detections and spectral detections complement each other. They are fused into a classification map based on geometry and material type by a supervised cascaded neural network, which consists of two levels of neural nets. The first level (sensor level) consists of two neural nets: a spectral neural net (SN1) and a spatial neural net (SN2). The second level is a single neural net, which is the fusion neural network (FNN). The algorithm flow is shown in Fig. 1.

On the front side of the spectral/spatial fusion network, the spatial detections (clusters) are fed into the spatial sensor net without assigning specific classification labels. The spectral detections are similarly fed into the spectral sensor net without specific labels. On the back side of the network, supervised training is performed on a small number of objects that are recognized in a scene by a human analyst. The detection decisions from both sensor nets are fed into the FNN for the final classification.

2.1 *Spatial Segmentation of PCD*

A PCD exists in a 3-D space (x, y, z). Each point is represented by the vector $\mathbf{r}_i = r_{ix}\mathbf{x} + r_{iy}\mathbf{y} + r_{iz}\mathbf{z}$, where $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are orthonormal unit vectors. The set of all points in the

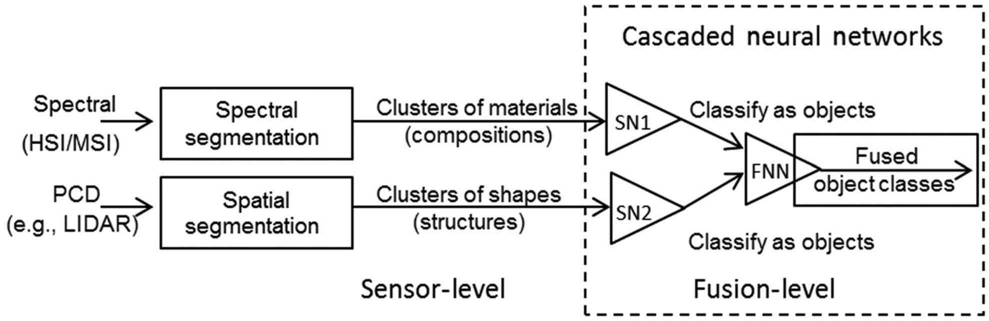

Fig. 1 Algorithm flow for the proposed distributed adaptive sensor fusion architecture. A bottom-up approach is used where spectral segmentation is applied to multiband spectral imagery (such as HSI or MSI data) and spatial segmentation is applied to PCD (such as obtained from LIDAR). The groups from the segmentation are then given class labels and fed into the cascaded neural networks.

point cloud is denoted by $\mathbf{r}_i \in \Omega$ for $\forall\ i \in [1, N]$, where N is the number of points in the cloud of the observed space Ω.

A combined mean-shift and dispersion-based approach is proposed for the spatial segmentation and classification of PCD. Typically, for PCD from active sensors such as LIDAR, manmade objects are denser than natural ones. With active sensors such as LIDAR, manmade objects tend to reflect better than most natural objects, such as tree leaves, grass, or wetlands. As a result, manmade objects, in general, produce more points than natural objects. Accordingly, the manmade objects are denser. Objects with a smaller dispersion-based density metric as will be described in Sec. 2.1.2 are more likely to be natural, whereas objects with a larger value are more likely to be man-made.

2.1.1 *Mean-shift initial segmentation*

A mean-shift clustering is proposed for performing the spatial segmentation of PCD. This method is a nonparametric clustering technique that requires no knowledge of the number of clusters or the shape of these clusters. The method was originally introduced by Fukunaga[27] and is discussed further in Refs. 28 and 29. The goal of using mean shift is to cluster densest object in 3-D within a spatial distance defined by parameter "bandwidth." An object that is clustered together is one that can fit within the boundary of 3-D bandwidth. For instance, with bandwidth of 3 m, any object with each dimension of <3 m is clustered to become an object.

Clusters from this method are formed at the densest areas in R^d space. Mean shifting from an initial point toward a denser area occurs and clusters are formed with a particular size. This causes either new clusters to be created or existing ones to expand along its path. When the mean-shift converges to a fixed point, it is reinitialized with a new unused point. The process is repeated until all points are used.

More specifically, mean-shift theory is based on density search in space R^d, for a given dimension d. It is a hill climbing algorithm applied to the multivariate density function $\hat{f}(\mathbf{r})$ at the point $\mathbf{r}$ by kernel density estimate function for N data points

$$\hat{f}(\mathbf{r}) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{\mathbf{r} - \mathbf{r}_i}{h}\right), \quad \forall\ \mathbf{r} \in R^d, \tag{1}$$

where $h > 0$ is a smoothing bandwidth parameter and $K(\cdot)$ is a symmetric kernel. Typical kernels are obtained from the uniform and Gaussian distributions. If the kernel is Gaussian (as in our case), $K(\cdot)$ can be expressed as

$$K(\mathbf{x}, h) = \frac{1}{(2\pi h)^{\frac{d}{2}}} \exp\left(-0.5 \frac{\|\mathbf{x}\|_2^2}{h}\right). \tag{2}$$

Taking the gradient of the multivariate kernel density estimate, we obtain the following expression:

$$\nabla\hat{f}(\mathbf{r}) = \left[\frac{2c_k}{Nh^{d+2}} \sum_{i=1}^{n} g\left(\| \frac{\mathbf{r} - \mathbf{r}_i}{h} \|^2\right)\right] \times \left[\frac{\sum_{i=1}^{n} \mathbf{r}_i g\left(\| \frac{\mathbf{r}-\mathbf{r}_i}{h} \|^2\right)}{\sum_{i=1}^{n} g\left(\| \frac{\mathbf{r}-\mathbf{r}_i}{h} \|^2\right)} - \mathbf{r}\right], \tag{3}$$

where $g(\mathbf{r}) = -k'(\mathbf{r})$ is defined as the derivative of $K(\cdot)$. The second term in the right-hand side of (3) is the mean shift at $\mathbf{r}$, denoted as $m_g(\mathbf{r})$. The vector always points toward the direction of maximum increase in the density. The dimension of the space is denoted by d. With this model, we implement the mean-shift algorithm as follows:

Step 1) Compute the mean-shift vector $m(r_i)$

$$= \left[\frac{\sum_{i=1}^{n} \mathbf{r}_i g\left(\| \frac{\mathbf{r}-\mathbf{r}_i}{h} \|^2\right)}{\sum_{i=1}^{n} g\left(\| \frac{\mathbf{r}-\mathbf{r}_i}{h} \|^2\right)} - \mathbf{r}\right]. \tag{4}$$

Step 2) Translate the density estimation $\mathbf{r}_i^{t+1} = \mathbf{r}_i^t + m(\mathbf{r}_i)$. (5)

Step 3) Iterate (1) and (2) until $\nabla\hat{f}(r) < \tau$, for a predefined tolerance τ. (6)

In summary, the mean-shift algorithm groups a point cloud into separate clusters by shifting the mean toward a denser direction where the majority of points are located. These clusters are fed into a dispersion-based segmentation method.

2.1.2 *Dispersion-based segmentation*

Once the point cloud has been segmented into a number of clusters, these clusters can be further segmented based on density. We use a dispersion-based density metric to measure the density of each cluster in a space of dimension d. This algorithm is described as follows:

$$D_j = \frac{1}{d}\left[d - \frac{\sigma_x}{|X|} - \frac{\sigma_y}{|Y|} - \frac{\sigma_z}{|Z|}\right] \to [0,1], \tag{7}$$

where $d = 3$ for 3-D data; j is the cluster number; σ_x, σ_y, and σ_z are the standard deviations in the x-, y-, and z-directions; and $|X| = |r_{x\ \max} - r_{x\ \min}|$, $|Y| = |r_{y\ \max} - r_{y\ \min}|$, and $|Z| = |r_{z\ \max} - r_{z\ \min}|$, where the maximum and minimum values are taken over cluster j. This function maps the sum of all standard deviations of a cluster to [0, 1]. Note that dense clusters have a density metric close to one and sparse clusters have a density metric close to zero. Figure 2 shows a scatter plot for a cluster of points in two directions that illustrate the information we are trying to capture with this function. The cluster in Fig. 2 is rotated so that its longest length is along x-axis and its highest point to center mass is perpendicular to this axis.

2.2 *Multiband Spectral Segmentation*

An unsupervised multiband segmentation approach is used on the spectral data that makes use of a stochastic mixture model (SMM),[2] also sometimes called a finite mixture model.[3] The stochastic expectation algorithm is proposed as a means to obtain and implement the solution.[2,3,30] Our implementation of the SMM and proposed algorithm is described in this section.

2.2.1 *Stochastic mixture model*

An SMM is used as the foundation for the proposed multiband spectral segmentation method. According to the SMM, a multimodal probability density function can be obtained by the mixture model

$$f(x;\Psi) = \sum_{i=1}^{M} \pi_i f_b(x;\Theta_i), \tag{8}$$

π_i is the mixture weight for class i, $x \in R^b$, and $f_b(x;\Theta_i)$ is a single unimodal distribution of dimension b, with parameters Θ_i. Since we are now in spectral space, we distinguish the dimensionality as different from spatial space by using the notation b as the number of dimensions in spectral space. The SMM describes the spectral data as a mixture of pixels from M different spectrally homogeneous (unimodal) distributions. In general, the unimodal distributions could be of any type; however, we will be presuming the distributions are multivariate normal. Consequently, each distribution $f_b(x;\Theta_i)$ can be fully characterized by the parameters $\Theta_i = \{m_i, K_i\}$, where m_i and K_i are the mean and covariance for class i. Figure 3 shows an example of three unimodal Gaussian distributions in a 2-D space that could be modeled using (8).

The parameters of the SMM model

$$\Psi = \{M, \pi_1, \pi_2, \ldots, \pi_M, \Theta_1, \ldots, \Theta_M\}, \tag{9}$$

are not known and are estimated through an iterative procedure that maximizes the likelihood function $L(X;\Psi)$ of the data set X

$$L(X;\Psi) = \prod_{n=1}^{N} \sum_{i=1}^{M} \pi_i f_b(x_n; m_i, K_i) \tag{10}$$

$$L(X;\Psi) = \prod_{n=1}^{N} L(x_n;\Psi), \tag{11}$$

where $L(x_n;\Psi)$ is the pixel likelihood function.

The maximum likelihood solution is $L(X;\hat{\Psi})$, where $L(X;\hat{\Psi}) > L(X;\Psi)$ for every $\Psi \neq \hat{\Psi}$.

2.2.2 *Expectation–maximization*

The expectation–maximization (EM) algorithm is an iterative method for finding a maximum likelihood solution of parameters in statistical models, where the model depends on unobserved latent variables.[30] The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter estimates are then used to determine the distribution of the latent variables in the next E step. With each iterative step (j), the model parameters Ψ^j are updated, such that there is an increase in the likelihood function

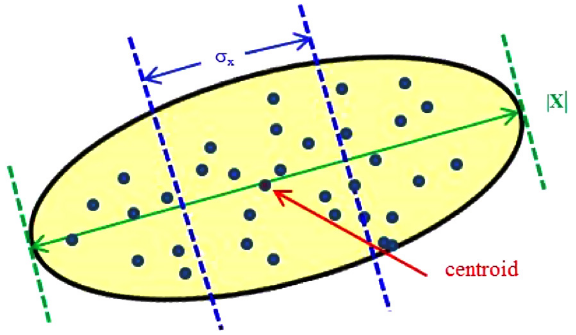

Fig. 2 Scatter-plot of a cluster in 2-D. The x-axis crosses the centroid point of the cluster at the direction, in which the cluster width is at its maximum. The y-axis is perpendicular to the x-axis. The vertical z-axis is orthogonal to the horizontal xy-plane.

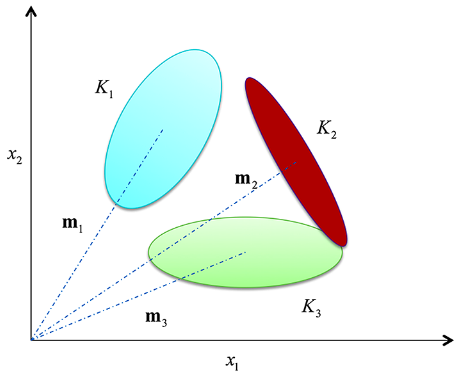

Fig. 3 Three unimodal Gaussian distributions are shown in 2-D space. Each distribution $f_b(x;\Theta_i)$ can be fully characterized by the parameters $\Theta_i = \{m_i, K_i\}$, where m_i and K_i are the mean and covariance for class i.

$$L(X;\Psi^{j+1}) \geq L(X;\Psi^{j}). \tag{12}$$

The algorithm terminates when a maxima has been found.

Unfortunately, there are some disadvantages to the EM algorithm. First, the maxima achieved through the process just described may be either a local maxima or the global maximum. The EM process of increasing the overall mixture likelihood in incremental steps can sometimes cause it to get stuck in a local maxima (rather than converge to the global maximum) if that path requires a step which does not increase the overall likelihood. Second, the EM process tends to converge slowly to a final solution. Finally, the number of model components (M) must be determined in advance.

2.2.3 *Stochastic expectation–maximization algorithm*

The SEM algorithm overcomes some of the disadvantages of the EM approach and it is proposed here to estimate the parameters of the SMM. SEM models and fits iteratively an M-component Gaussian mixture model to the given set (or bands) of data. The method is a variation of the EM algorithm, mentioned previously. The SEM algorithm is an iterative method that uses stochastic cluster assignment for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models. During each iteration, SEM segmentation assigns each pixel to one cluster based on the maximum of posterior probability, similar to the EM. However, it does so in a way that overcomes some of the disadvantages of SEM.

In an attempt to avoid getting stuck in local minima, SEM uses a hard segmentation technique to compute estimates of the weight, mean, and covariance for each unimodal distribution. Specifically, only the pixels within each cluster are used to compute these estimates. The process of assigning pixels into clusters causes SEM to take larger steps than the EM approach and offers an ability to jump away from a path toward local minima in favor of reaching the global maximum. However, unfortunately, although the chances are improved for reaching the global maximum, there is still no guarantee the final model solution corresponds to the global solution. Taking larger steps also has the benefit of causing SEM to converge faster than the EM.

SEM provides an unsupervised approach to segment multivariate data into clusters without having *a priori* knowledge of the number model components M. Unlike the EM algorithm, SEM does not require the actual number of model component to be known. SEM only requires the maximum number of components to be specified. SEM begins with a specified maximum number of mixture model components. Subsequently, during any of the iterations, if a model component weight falls below a specified threshold, $\gamma_{\min}$ then SEM is reinitialized with one less component: $M \rightarrow M-1$.

Initialization of SEM algorithm. An initial "guess" of the maximum number of mixture components M is chosen, along with the minimum allowable component weight $\gamma_{\min}$. The pixel component posterior probability is initialized with the uniform distribution

$$P_i(x_n;\hat{\Psi}_i^{(j)}) \sim U[0,1] \quad \forall\ i,n. \tag{13}$$

Stochastic assignment (S-step at the ***j****-th iteration).* A stochastic mixing model (SMM) provides an effective method for describing the spectral statistics of pixel ensembles of this type. The SMM is defined in terms of several "pure" or "hard" constituent classes, each of which is described by a unique multivariate Gaussian probability density. Image pixels belonging to a given class are assumed to be independently drawn from the density that defines that class. Figure 4 shows each pixel randomly assigned to a class i assigned using a multinomial distribution with probability p_i.

An indicator function is used to designate class membership

$$I_i(x_n) = 1, \text{if } x_n \text{ is assigned to class } i; \text{otherwise,}$$
$$I_i(x_n) = 0. \tag{14}$$

Expectation (E-step at ***j****-th iteration).* Compute the pixel-mixture model likelihood and pixel-component posterior probabilities

$$L(x_n;\hat{\Psi}^{(j)}) = \sum_{i=1}^{M} \pi_i^{(j)} f_b(x_n; m_i^{(j)}, K_i^{(j)}) \tag{15}$$

$$P_i(x_n;\hat{\Psi}^{(j)}) = \frac{\pi_i^{(j)} f_b(x_n; m_i^{(j)}, K_i^{(j)})}{L(x_n;\hat{\Psi}^{(j)})}. \tag{16}$$

Maximization (M-step at the ***j****-th iteration).* The SMM parameters for the next step $\pi_i^{(j+1)}$ and $\Theta_i^{(j+1)} = \{m_i^{(j+1)}, K_i^{j+1}\}$ are estimated

$$\hat{\pi}_i^{(j+1)} = \frac{N_i^{(j)}}{N}, \tag{17}$$

$$m_i^{(j+1)} = \frac{1}{N_i^{(j)}} \sum_{n=1}^{N} I_i^{(j)}(x_n) x_n, \tag{18}$$

$$K_i^{(j+1)} = \frac{1}{N_i^{(j)}} \sum_{j=1}^{N} I_i^{(j)}(x_n)(x_n - m_i)(x_n - m_i)^T, \tag{19}$$

where N is the number of pixels and $N_i^{(j)}$ is the number of pixels in class i at the j-th iteration.

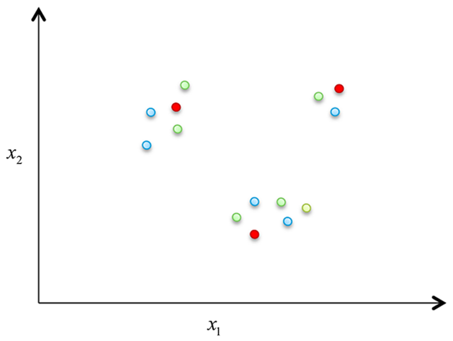

Fig. 4 Stochastic assignment step. This diagram shows an example of each pixel randomly assigned to a class i assigned using a multinomial distribution with probability p_i.

Figure 5 shows an illustration of the multivariate normal contours resulting from such estimations.

2.2.4 *SEM processing flow*

Figures 6 and 7 are diagrams that show the processing flow of the multiband segmentation. The SEM portion described in Sec. 2.2.3 is shown by the block diagram in Fig. 6. The overall multiband processing (including band reduction) is shown in Fig. 7. If the number of spectral bands is high (larger than, say, 10), band reduction should be performed prior to running SEM. If so, among all the input spectral bands b, the user defines the number of principle components bands q, which is the reduced number of bands for SEM to apply in the SMM (finite mixture model). The original input of b bands of a spectral data cube is transformed to a smaller number q, the number of principle component bands, resulting in the dimensional reduction to the q band data cube. SMM is then applied with mixture model components and parameters (mean, covariance, weight) from the process computed as shown in Fig. 6 to perform clustering to obtain a cluster pixel index image map. The result is a classification map in which a pixel contains its clustered class. In SEM segmentation, a pixel is assigned to only one cluster, at each iteration, based on the maximum of posterior probability (similar to the mean-shift algorithm).

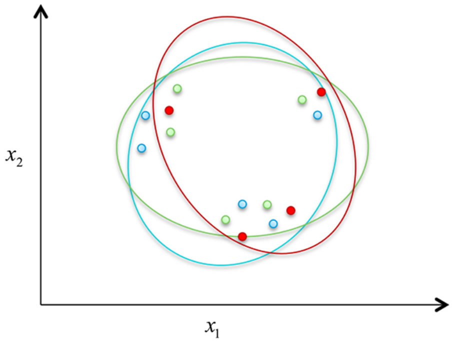

Fig. 5 This diagram shows an example of the contours for three multivariate Gaussian distributions (in two dimensions) defined by the mean and covariance estimates computed at one of the iterations.

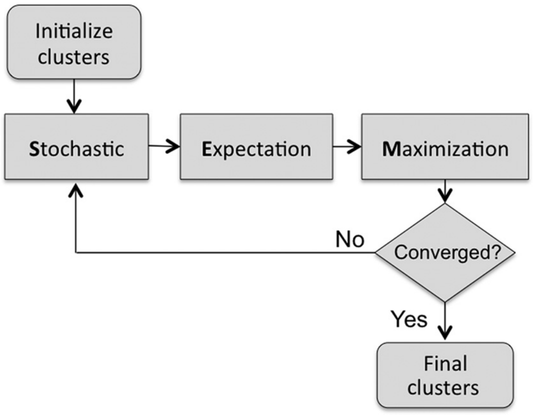

Fig. 6 This block diagram shows the SEM portion of the processing flow for MSI/HSI processing, implemented as a loop of the stochastic, expectation, and maximization steps.

2.3 *Spectral–Spatial Neural Net Sensor Fusion*

We propose a cascaded architecture for spectral–spatial neural net sensor fusion (spectral–spatial NNSF) using cascaded back propagation neural networks to accomplish sensor fusion classification for our two independent data sources (spectral and LIDAR/3-D point cloud). Our motivation comes from observing the successful performance of NNSF methods in scenarios relevant to our application; specifically, the successful data fusion by Chair and Varshney[31] was shown to be achieved similarly (in terms of probability of detection and probability of false alarms) by a cascaded neural net by Levine and Khuon.[32–36] Successful performance of neural networks was achieved for nonlinear detection and classification problems for imagery data fusion in this research. A detailed discussion of the FNN for data/sensor fusion can be found in Refs. 31–37.

The spectral–spatial NNSF builds upon the work of Levin and Khuon[32] to accomplish multisensor data fusion. The architecture consists of a set of independent-sensor neural nets, one for each sensor, coupled to a fusion net. Each sensor is trained (from a representative data set of the particular sensor) to map to a hypothesis space output. The decision outputs from the sensor nets are used to train the fusion net to an overall decision. Unlike the NNSF used for the firefly experiments in Ref. 32, where the decision set is the same for all sensor neural nets and the FNN, in the proposed spectral–spatial NNSF, the decision set of the spectral neural net can be different from the decision set of the PCD neural net and both decision sets are different from the fusion decision set. This is because of the different types of PCD and spectral multiband sensors (e.g., LIDAR, PCD, and HSI) and classifications.

The cascaded networks are implemented within the distributed adaptive framework for spectral and PCD fusion shown in Fig. 8. Two registered classified data sets are first generated. The spectral multiband data set that is classified by SEM is fed into the first sensor neural net (SN1) along with the Z data. As shown, a "Get Z from PCD" process is performed prior to feeding the spectral segmentation into the SN1 network to provide additional information about the spectrally segmented clusters extracted by SEM. In this step, after the HSI is registered to the LIDAR data, the Z information (elevation) on each HSI pixel is obtained and used with the segmentation to help define the SN1 network. For instance, a cluster with sand material can be defined as part of the beach if the height is relatively low but another cluster with sand material can be part of tall building or a car garage if the height is relatively high.

Similarly, the PCD data set that is segmented and classified by mean shift is fed into the second sensor neural (SN2). During the training phase, a number of selected objects in the PCD classification map are used to train its sensor net. All pixels of a particular object are mapped to the same output neuron in SN1 and for the same object in spectral classification map all pixels are mapped to the same output neuron in SN2, as described in the material that follows.

The spectral–spatial NNSF is used to combine spectral classification of multiband (pixel) with spatial classification of PCD containing point cloud clusters. NNSF is a supervised classifier and is implemented in two phases as follows:

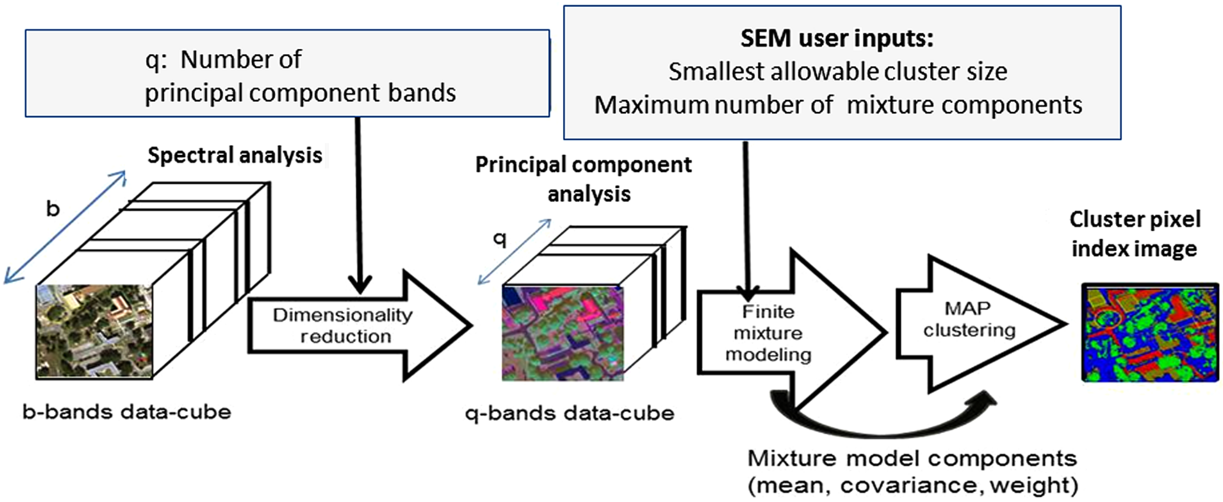
q: Number of
principal component bands
SEM user inputs:
Smallest allowable cluster size
Maximum number of mixture components
Spectral analysis
b
Dimensionality reduction
b-bands data-cube
Principal component analysis
q
q-bands data-cube
Finite mixture modeling
MAP clustering
Cluster pixel index image
Mixture model components
(mean, covariance, weight)

Fig. 7 The overall processing steps for the spectral segmentation (including band reduction if it is needed) are shown.

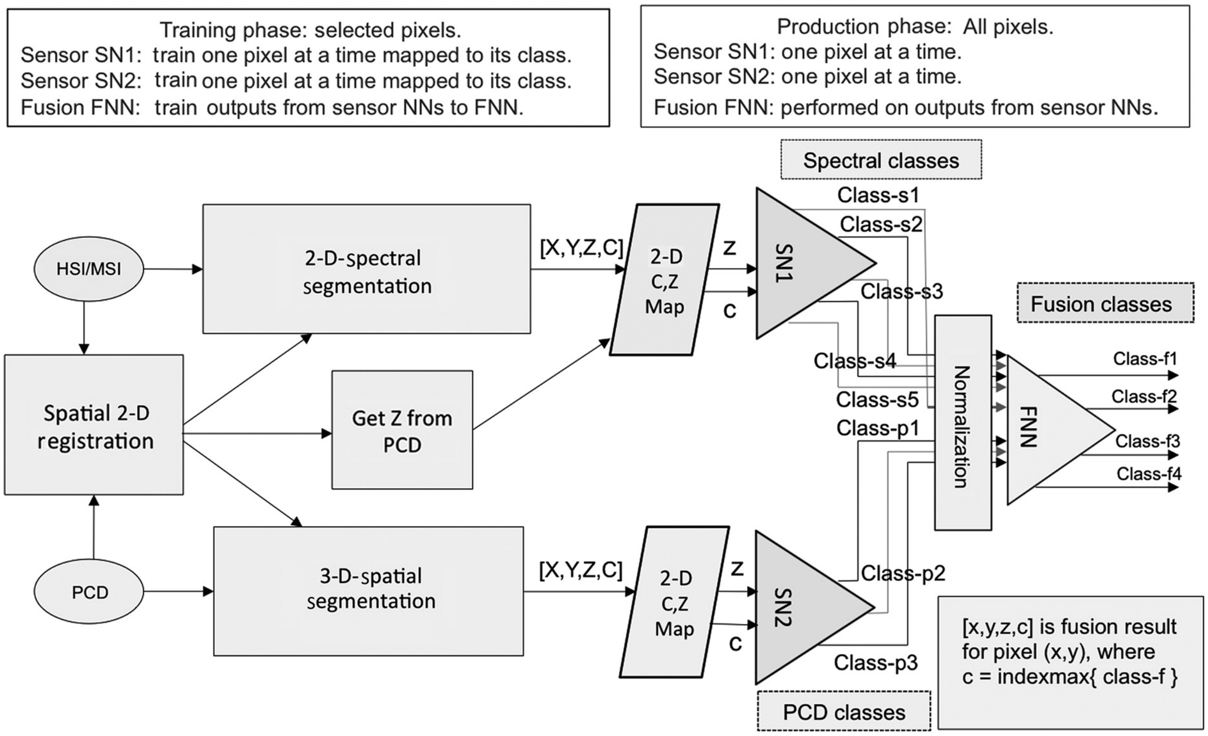
Training phase: selected pixels.
Sensor SN1: train one pixel at a time mapped to its class.
Sensor SN2: train one pixel at a time mapped to its class.
Fusion FNN: train outputs from sensor NNs to FNN.
Production phase: All pixels.
Sensor SN1: one pixel at a time.
Sensor SN2: one pixel at a time.
Fusion FNN: performed on outputs from sensor NNs.
Spectral classes
HSI/MSI
2-D-spectral segmentation
[X,Y,Z,C]
2-D C,Z Map
Z
C
SN1
Class-s1
Class-s2
Class-s3
Class-s4
Class-s5
Class-p1
Fusion classes
Normalization
FNN
Class-f1
Class-f2
Class-f3
Class-f4
Spatial 2-D registration
Get Z from PCD
PCD
3-D-spatial segmentation
[X,Y,Z,C]
2-D C,Z Map
Z
C
SN2
Class-p2
Class-p3
PCD classes
[x,y,z,c] is fusion result for pixel (x,y), where c = indexmax{ class-f }

Fig. 8 The proposed spectral sensor and PCD fusion architecture are shown. This approach is initiated with concurrent spectral and spatial segmentation processes that operate on geometrically registered spectral and spatial data, respectively. Elevation data (*Z* data) available from the PCD is subsequently used to help label classes from the spectral segmentation just prior to input into the spectral sensor neural network (SN1). Spectral and spatial neural networks (SN1, SN2) are initialized from the *Z*-data modified spectral segmentation and spatial segmentation processes. Outputs from the SN1 and SN2 processes are fed into the FNN.

Training phase: Repeat a training set that covers all classes until all neural nets are fully trained when the error at each output neuron is smaller than a predefined tolerance:

- Spectral-NN (SN1): Choose a pixel in color/MSI/HSI with its pixel class mapped to its class neuron. Then train the net.
- PCD-NN (SN2): Choose nearest pixel in the PCD with its cluster-class mapped to its class neuron. Then train the net.
- FSN-NN: Outputs from spectral-NN and PCD-NN are mapped to its class neuron. Then train the net.

Production phase: Repeat for each pixel p to generate output image.

- Spectral-NN: Input p class in color/MSI/HSI and output to the FNN.
- PCD-NN: Input p class in PCD with its cluster-class and output to the FNN.
- FSN-NN: Outputs from spectral-NN and PCD-NN are input into FSN-NN and then output the result.

After both sensor neural nets are completely trained, the final outputs are fed into the fusion net (FNN) for fusion training. When the fusion net is completely trained, the cascaded neural nets are ready for the production phase.

In the production phase, each pixel in the PCD data set and the corresponding pixel in spectral multiband data set are both fed simultaneously into their own sensor net. The outputs from all sensor nets are fed simultaneously into the fusion net. By repeating this routine, the fusion classification map is generated. Using the z-coordinate obtained from the PCD data set, the fusion classification map is generated in 3-D.

3 Description of the Experiment

3.1 *Data Collection and Image Registration*

The experiment considered the specific case of HSI for spectral multiband data and PCD extracted from LIDAR. Data were collected during an unclassified airborne collection of LIDAR and HSI data that was performed under contract with NGA by the University of Florida and Optech International in November 2010. A CASI 1500 sensor was used to collect 72 spectral channels of HSI data in the 0.367 to 1.04 μm spectral region at a 1-m spatial ground sampling distance (GSD). An Optech ALTM sensor-3001 (a linear-mode sensor) was used to collect LIDAR data at 1 point/m^2. The two sensors were mounted on the same aircraft enabling the data to be collected simultaneously. This also enabled the geometric registration of the LIDAR and HSI data to be relatively simple. Multiple flight lines were collected. The specific scenes used for this experiment were collected over the University of Southern Mississippi Gulf Park Campus near Long Beach, Mississippi.

The LIDAR PCD was registered to the HSI using an eight-parameter transformation. The corresponding points were manually picked and then used in a least squares solution to determine the parameters of the transformation. Application of this new transformation was quite simple because the LIDAR data are composed of simple points, which can be modified via an equation without any dependence on adjacent pixels and no worries of interpolation/resampling. With a 1-m resolution for both data sets, the error at the center of the data was determined to be $\sim 1/3$ m. We are studying a relatively small area and so this registration error was acceptable to us. Improvements to the process are certainly possible in terms of accuracy and automation, but this is a subject of its own, and further investigation is considered beyond the scope of our current study.

Figures 9 and 10 show scenes of the study site from the registered HSI data and LIDAR data, respectively. Figure 9 is an RGB color composite scene of HSI visible bands of this area. Figure 10 shows the corresponding registered PCD-derived elevation data (Z data). In Fig. 10, the brightness level corresponds to elevation, where the height is indicated by brightness. Bright areas correspond to high areas; dark values correspond to low areas. There is some stripping of periodic lines running diagonal across the scene, which creates a challenge to our approach. This artifact was introduced into the data at some point during the sensor collection.

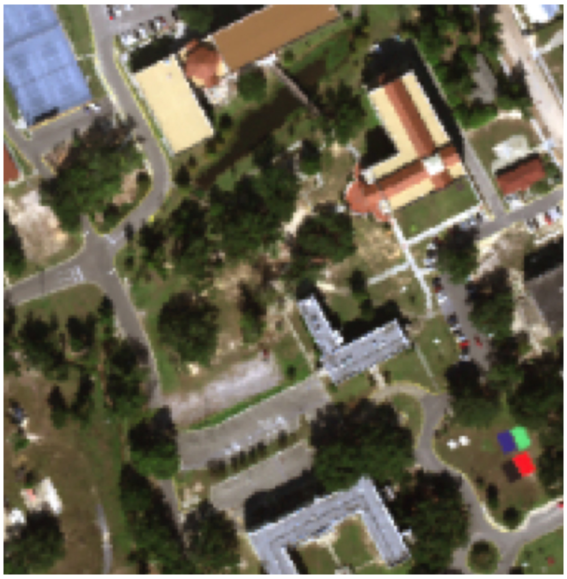

Fig. 9 RGB color composite of visible bands from the 72 bands of HSI data shows the study area of the experiment. Privacy statement: Domestic imagery was collected in accordance with a valid Proper Use Memorandum approved by NGA, and is not directed at U.S. persons.

3.2 *Implementation of Algorithms to Registered Hyperspectral and LIDAR Data*

Once the registration was completed, the algorithms described in Sec. 2 were applied to the registered data. To

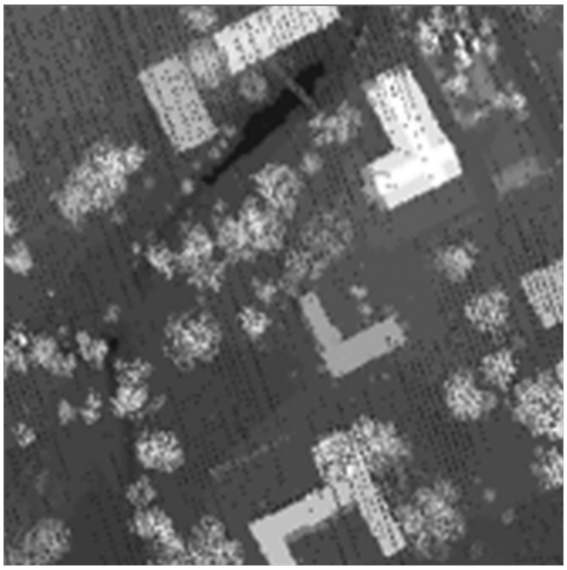

Fig. 10 The PCD-derived elevation data (Z data) from LIDAR, registered to the HSI data, is shown. The brightness level corresponds to elevation, where the height is indicated by brightness. Bright areas correspond to high areas; dark values correspond to low areas. Privacy Statement: Domestic imagery was collected in accordance with a valid Proper Use Memorandum approved by NGA, and is not directed at U.S. persons.

begin the processing, the 3-D point cloud LIDAR data were spatially segmented into a map of clustered objects based by a multidimensional mean-shift segmentation, which contains numeric labels for each object. Independently, the multiband HSI data were spectrally segmented using SEM into a map containing numeric labels for the segmented regions based on spectral similarity. Up to this stage, the process has been unsupervised, and neither the labels for the object map or spectral map are associated with known objects or classes. To begin the next stage, the labels are manually given an association with specific types of objects (for the object map) and specific materials (for the spectral map). In this manner, the segmentation maps become class maps, providing classifications of spatial objects and classifications of material classes. These class maps provide the inputs for the process shown in Fig. 8.

For sensor fusion, spatial detections and spectral detections are fused into final detections by a cascaded neural network, which consists of two levels of neural nets. The first layer is the sensor level consisting of two neural nets: spatial neural net and spectral neural net. The second level consists of a single neural net, i.e., the FNN.

Two experiment trials were made. In the first trial, the SEM parameter controlling the number of class labels was set to five. In the second trial, this parameter was set to six. This parameter determines the maximum number of classes so there is the possibility the SEM algorithm might generate less than this maximum. Otherwise, the processing for the mean-shift and the cascaded fusion processes were similar in these trials, except that for the fusion process, the number of input neurons in the neural net would change based on the class labels resulting from the SEM process.

4 Results

In the first trial, the spectral and spatial segmentation algorithms resulted in five spectral categories and four spatial categories, respectively. For the spectral segmentation, SEM's parameter to control the maximum number of cluster labels was set to five. SEM generated exactly five labels, even though less than five labels were possible. We assigned class labels posteriori based on our observation of the correspondence between the segmented clusters and obvious ground features and clusters in each of the five classes. In this manner, the labels were assigned the names Non-Veg01, Non-Veg02, Veg01, Veg02, and Others. Accordingly, each pixel in the scene was classified as some type of material class. For the spatial segmentation, the mean-shift and dispersion classifier algorithms segmented the original point cloud into four categories associated with objects: Man-made, Natural, Higher ground, and Grass/Ground (Bare-earth).

Figure 11 shows the segmentation results for the HSI and LIDAR of the study area, as well as the results of fusing the information through a cascaded neural net. This figure shows a sequence of results that demonstrates (qualitatively/visually) that the fusion step improves classification beyond the single sensor level. Figure 11 contains four 2-D images: the classification map of HSI multibands (top left), the classification map of LIDAR point cloud (top right), the fusion classification map (bottom left), and the RGB composite of the HSI scene (bottom right).

In Fig. 11(a), the spectral classes resulting from clustering the HSI data are broad categories that correspond to spectrally similar materials but they do not always correspond to a similar type of physical object: a red class object can be a rooftop, road, parking lot; and a brown object can be a different type rooftop or road, as well as a tennis court.

Similarly, the spatial classes of LIDAR data in Fig. 11(b) are vague: Higher ground can be a grassy area; and a Grass area can contain no actual grass at all. This figure shows that in some cases, tree branches appear to obscure a portion of a building (highlighted by two yellow circles). The mean-shift method provides foliage penetration and feature extraction from the scene in 3-D. Note also that despite the periodic (diagonal) lines seen in Fig. 9(b), the segmentation as shown here is performed with only some minor problems with these lines. This is due to the robustness in mean-shift segmentation and classification algorithm. Although some of the periodic lines remain in the LIDAR class map, later we will see these lines are removed during fusion process.

The success of the fusion step is shown in Fig. 11(c), where the fusion classes are clearer and more specific than any single sensor class. For instance in the fusion class map, the red class specifically corresponds to building rooftops and are clearly separated from the parking lots, roads, and tennis court. Grassy areas are clearly separated from trees.

The regions within the yellow circle and rectangles provide specific examples of the success achieved by the fusion approach. As a first example, consider the region within the circle of Fig. 11(a). The SEM algorithm had difficulty in this region, where too much of the region was segmented into clusters that are obviously associated (visually) with grass that was found throughout the rest of the scene. This region should have been segmented predominantly into trees. Use of the PCD information provided the fusion process sufficient information so that the area was correctly segmented into a cluster associated (visually) with trees that were found throughout the rest of the scene. As a second example, consider the two rectangular regions at the top left portion of the scene. The SEM algorithm incorrectly associates these two regions with the same class; however, the rectangular region on the left is a tennis court and the region to its right is a rooftop containing a multiple materials. Use of the elevation information in the PCD provides the fusion algorithm with sufficient data to segment the region on the right into the same cluster as the rooftop that is adjacent to it (on the right side). As a third example, the SEM algorithm incorrectly segmented the rectangular region at the far right of the scene into many areas of clutter. The fusion process consolidated these areas into far less clutter.

Figure 12 shows a class reference map, which is an RGB image with an overlay delineating some of the test areas used as ground truth to compute quantitative segmentation results given in the tables. Test areas were defined for five regions containing a building (or part of a building), five regions containing a road or parking lot, five regions containing at least one tree, and five grassy regions for a total of 20 test areas.

Tables 1 and 2 show some quantitative segmentation results for the HSI and fused HSI/LIDAR runs. The number of pixels in each test area was different, but it seemed most appropriate to compute an unweighted average of class

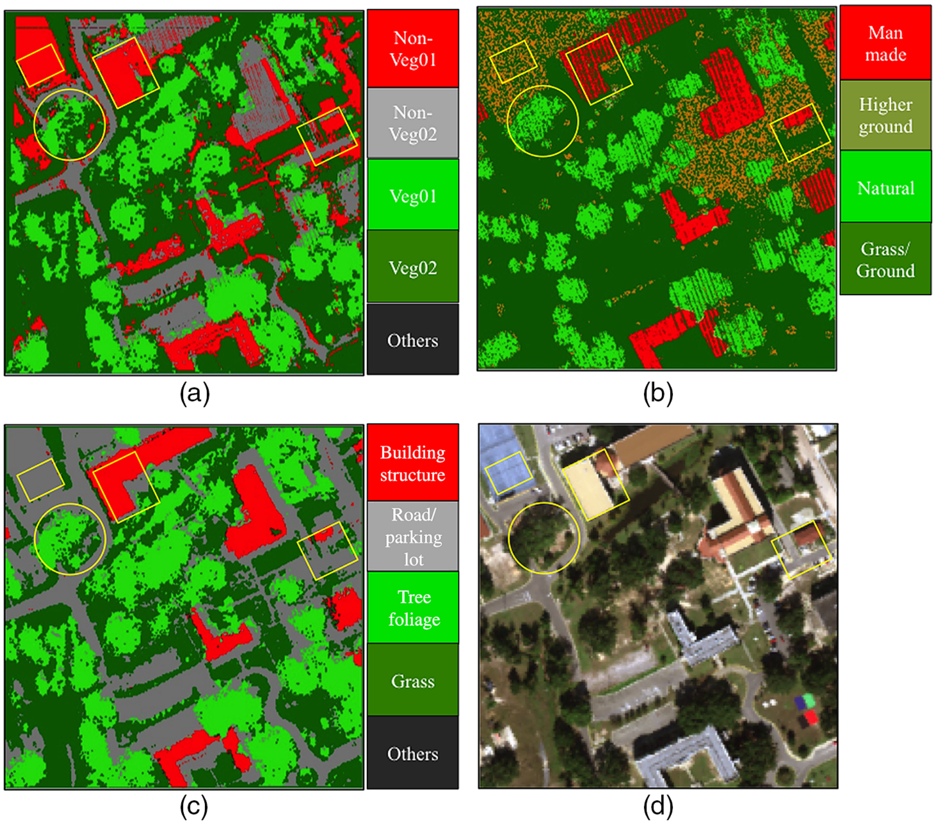

Fig. 11 The class map results of the first trial are shown. The HSI SEM segmentation map is displayed in (a). The red class consists of one type of spectrally similar nonvegetation material: Non-Veg01. The gray class consists of another type of nonvegetation material: Non-Veg02. The green class consists of one type of spectrally similar vegetation: Veg01. The dark green class consists of another type of vegetation: Veg02. The black class is of Others. The LIDAR mean-shift segmentation map is displayed in (b). There are four classes in the map: Man-made (red), Higher ground (orange), tall Natural (green), and Grass/ground (dark green). The class map for the FNN fusion processing is shown in (c). The corresponding RGB scene of HSI for the study site is shown in (d). Note that errors with the SEM segmentation found within the yellow circle and rectangles in (a) are resolved by using the information from the LIDAR segmentation (b) in a cascaded neural net to produce the fusion class map in (c).

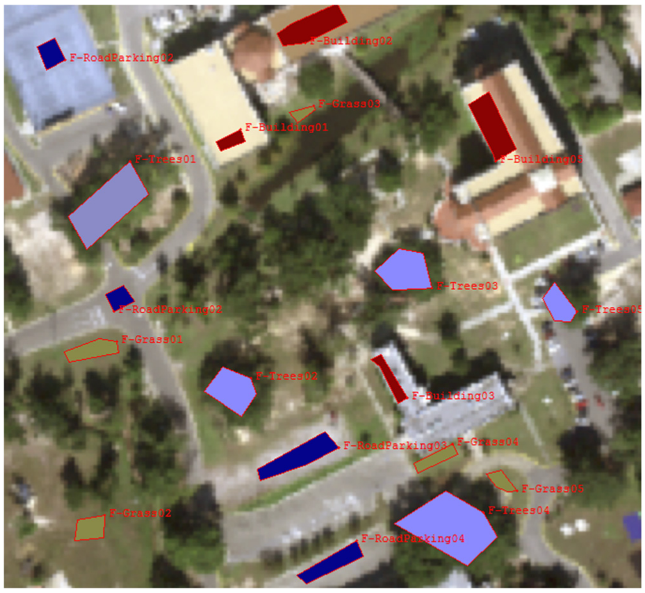

Fig. 12 An RGB reference map with an overlay delineating some of the test areas used as ground truth to extract assigned class labels and compute the confusion matrix results in Table 2. A similar map overlay (not shown) was used to compute the results shown in Table 1.

assignments over the regions. This is because we are interested in the results for class objects not individual pixels. Consequently, e.g., two buildings (or two grassy regions) should be given equal weight even though one building (or grassy region) may be much larger than the other. Segmentation results were extracted over these 20 test areas and the individual percentages of class assignments from each area were averaged giving equal weight to each area. Summary confusion matrices are shown that were computed for four broadly defined classes. Both the number assigned to each category and the percentages assigned to each category are shown.

These tables show the some of the success achieved for the proposed HSI-LIDAR processing approach. The overall (average) classification accuracy improved from 74.5% (HSI only) to 98.5% (HSI-LIDAR). The overall false alarm rate decreased from 25.5% to 1.5%.

We should mention that the confusion matrices in these tables do not exhaustively represent the performance of the algorithm. In particular, the classification accuracies of the regions containing "L shaped" buildings in the center and upper center of the image do not exhibit the positive improvements expected of the algorithm. More investigation is needed to determine the full reason why the algorithm fails in some regions. However, a partial explanation for some of

Table 1 HSI segmentation results. A summary confusion matrix of the HSI trial is shown that was computed for four broadly defined classes. Both the number assigned to each category and the percentages assigned to each category are shown. Test areas were defined for five regions containing a building (or part of a building), five regions containing a road or parking lot, five regions containing at least one tree, and five grassy regions (total of 20 test areas). Segmentation results were extracted over these 20 test areas and the individual percentages of class assignments from each region were averaged giving equal weight to each area. The total number of points is 26,821.

	Building	Road/parking	Trees	Grass	Total
Building (5 areas)	2931	1961	0	45	4937
Road/parking (5 areas)	2958	3033	0	213	6204
Trees (5 areas)	14	186	10,268	2158	12,626
Grass (5 areas)	0	0	2	3052	3054
Building (5 areas)	63.8%	35.5%	0.0%	0.7%	
Road/parking (5 areas)	44.5%	53.4%	0.0%	2.1%	
Trees (5 areas)	0.1%	0.8%	80.8%	18.4%	
Grass (5 areas)	0.0%	0.0%	0.1%	99.9%	

Average correct ID = 74.5%

Average false alarm = 25.5%

Total number of points = 26,821

Table 2 HSI-LIDAR segmentation results. A summary confusion matrix of the HSI-LIDAR trial is shown that was computed for four broadly defined classes. Both the number assigned to each category and the percentages assigned to each category are shown. Twenty test areas were defined in a similar manner as for the HSI trial results shown in Table 1. Segmentation results were extracted over these test areas and the individual percentages of class assignments from each area were averaged giving equal weight to each area region. The total number of points is 25,766.

	Building	Road/parking	Trees	Grass	Total
Building (5 areas)	4989	0	0	0	4989
Road/parking (5 areas)	0	5589	0	31	5620
Trees (5 areas)	0	358	11,658	465	12,481
Grass (5 areas)	0	0	0	2676	2676
Building (5 areas)	100.0%	0.0%	0.0%	0.0%	
Road/parking (5 areas)	0.0%	99.7%	0.0%	0.3%	
Trees (5 areas)	0.0%	1.9%	94.4%	3.7%	
Grass (5 areas)	0.0%	0.0%	0.0%	100.0%	

Average correct ID = 98.5%

Average false alarm = 1.5%

Total number of points = 25,766

the regions is that problems can occur with zero (or near zero) data returning from either sensor data source, such as with shadows or obscurations. Such missing data can cause confusion to the FNN, resulting in an erratic mapping between sensor neural net outputs.

Figure 13 shows the results of generating fusion 3-D classification maps in two perspective views using the cascaded neural net result shown in Fig. 11(c) and the 3-D information in LIDAR. Here, a 3-D image is shown from two different views. The colors for some of the label assignments are different due to the rendering process, but it is basically the same image as Fig. 11(c) portrayed in 3-D. With the additional z-dimension and two different views, this image provides a clearer separation of two objects: the tree branch leans over the building. In foliage penetration, with this mean-shift capability, the existence of a particular target can be verified simply by removing the overhanging foliage to expose its physical shape in red color.

In the second trial, SEM's parameter to control the maximum number of classes was set to six, and the result of the SEM run was to generate exactly six classes. Figure 14 shows the final product of this trial: fusion images in 3-D from PCD and HSI containing six classes. In addition to the five classes in Fig. 13, an additional class is added into the fusion classification. Instead of a single class of Trail or Road for a brown object, it is replaced by two brown classes. The light brown class is for Court (tennis court), Parking (parking lot), or Sand (sandy object/area) and the dark brown class is for Road (dark road) or Asphalt.

5 Discussion

Two-stage neural net classification provides a capability in multi-intelligence sensor fusion of two independent sensors, PCD and spectral multiband. The results from HSI and LIDAR multisensor fusion suggest that classification based on these two data types can be less ambiguous than single sensor classification.

The spectral and spatial sensor fusion of HSI and PCD provides a number of capabilities. The results of our investigation can be summarized as follows:

- The mean-shift algorithm provides an unsupervised method for processing PCD data. The algorithm provides a data reduction technique to remove redundant data, and an unsupervised segmentation method to organize data in nonoverlapped clusters for classification.
- The SEM algorithm is an unsupervised method for processing MSI/HSI data that provides finite mixture models and clustering.
- Spectral and spatial sensor-level classes are complementary.
- Sensor fusion can be achieved by cascaded neural nets.
- Fusion classification provides more detailed and accurate classification than a single-sensor classifier.

Results of the experiment support our initial hypothesis. Specifically, the overall (average) classification accuracy for the areas examined quantitatively in the scene improved from 74.5% (HSI only) to 98.5% (HSI-LIDAR). The overall false alarm rate decreased from 25.5% to 1.5% in these regions. However, as mentioned in Sec. 5, these quantitative results do not exhaustively represent the performance of the

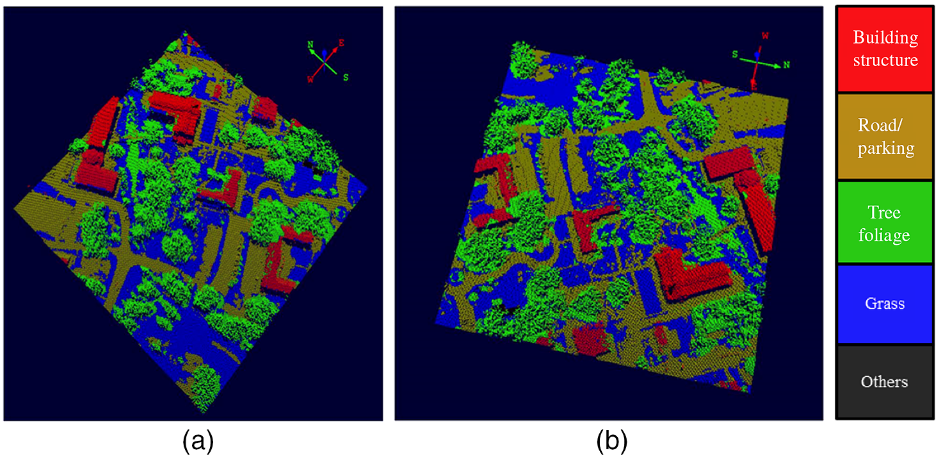

Fig. 13 Fusion 3-D classification maps for five classes. Shown are fusion 3-D classification maps for five classes in two perspective views. Building structure (red) objects are seen in open area or partially under foliage. Tree/Foliage (green) with branches and leaves cover a large area in the scene and some of them block objects underneath from being seen. The Road/Trail (tan) objects cover a higher ground. The Grass (blue) consists of lower vegetation at the ground level. The undeterministic class Others (black) is for shadow and less reflectance areas.

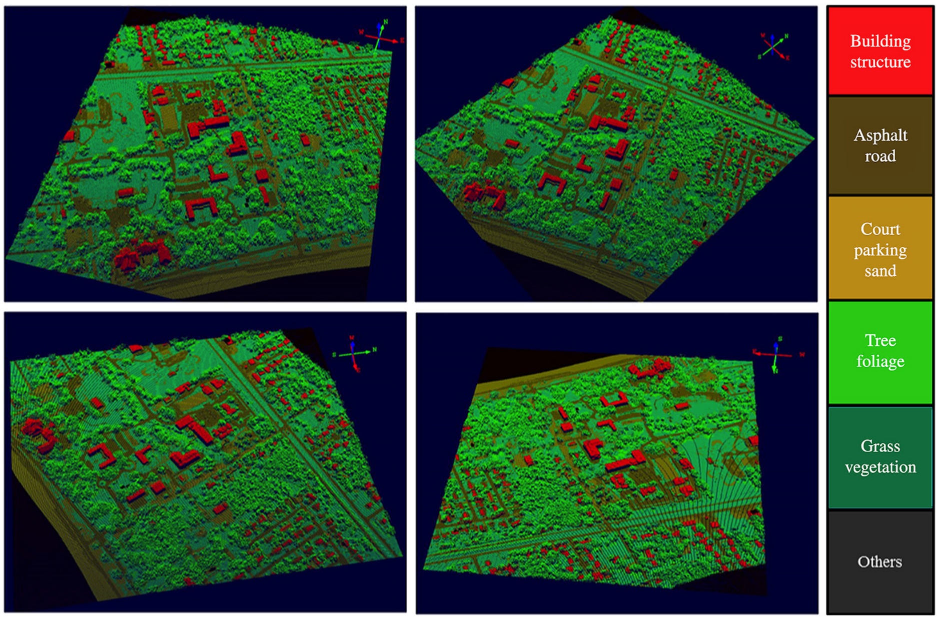

Fig. 14 Fusion classification maps for six classes. Fusion 3-D classification maps are shown for six classes in four perspective views. Building structure (red) objects are seen in open area or partially under foliage. Tree/Foliage (light green) with branches and leaves cover a large area in the scene and some of them block objects underneath from being seen. Asphalt/Road (brown) creates a drivable and connected road system. The Court/Parking/Sand (tan) covers a higher ground. The Grass or low vegetation (dark green) consists of lower vegetation at the ground level. The undeterministic class Others (black) is for shadow and less reflectance areas. These six fusion-level classes are less ambiguous compared to those ambiguous sensor-level classes.

algorithm. There are some exceptions we noted, where there are regions of the image that do not exhibit the positive improvements expected. We gave a partial explanation, above, but more investigation is needed to determine the full reason why the algorithm does not always act as expected.

The proposed method provides a classification map that delineates objects based on similarities in 3-D geometry and material composition. A number of spectral ambiguities were observed in the HSI scene, such as the ambiguity between roofs and parking lots, because they are often both composed of asphalt, and thus have similar signatures. The spectral classifier operating on HSI data is not able to distinguish this difference. Combining HSI and PCD by cascaded neural nets provides a better 3-D classification map than either sensor-level classification. False alarms are reduced by resolving the geometric and spectral ambiguities that often occur when using a single source of imagery. In addition to better classification, the periodic lines observed in the original LIDAR map and classification map are removed completely from the fusion classification maps by the neural net sensor fusion. Given that we have only validated this approach on one data set, the proposed framework should be tested on additional combined MSI/HSI and PCD data sets.

Fortunately, the proposed multisource fusion approach of combining spectral and spatial information is observed to be quite forgiving of mistakes made by either individual method, which is the strong advantage of multisource fusion. However, there is much potential for improvement. The proposed framework is generic in the sense that alternative approaches to mean shift and SEM that feed the FNN could be used and possibly improve performance. Certainly, better spectral-only classification accuracy can be achieved than that shown in Fig. 14(a). For example, the SEM method for HSI segmentation could be replaced by alternative unsupervised approaches, such as one of the newer spectral segmentation methods mentioned earlier,[6–12] older methods in Ref. 13, or the spectral–spatial methods mentioned earlier[16–24] Furthermore, PCD data also provide the opportunity to use scene geometry and solar illumination conditions to make radiometric corrections to improve classification results by mitigating shadows and illumination variations.[38]

Additional modifications could be made to the sensor fusion models. For example, neural net sensor fusion could be extended to more than two sensors. If there were two simultaneous data collections of LIDAR and HSI sensors several days before a storm and there was another data collection of radar sensor several days after the storm, change detection could be performed by fusing HSI and LIDAR data for classification, which can then be compared against the radar data classification. Alternatively, other fusion techniques, beyond neural net sensor fusion, such as the belief/mass Dempster Shafer and the linear-quadratic estimation Kalman filtering could be explored for sensor fusion. Further, the fully unsupervised sparse modeling approach for direct data fusion of HSI and LIDAR[8] is an alternative classification algorithm that could be combined with the neural net fusion algorithm.

References

1. P. Masson and W. Pieczynski, “SEM algorithm and unsupervised statistical segmentation of satellite images,” *IEEE Trans. Geosci. Remote Sens.* **31**(3), 618–633 (1993).
2. A. Stocker and A. Schaum, “Application of stochastic mixing models to hyperspectral detection problems,” *Proc. SPIE* **3071**, 47 (1997).
3. D. Manolakis and D. Marden, “Statistical modeling of hyperspectral imaging data and their applications,” Technical Report, MIT Lincoln Laboratory, Lexington, Massachusetts (2004).
4. D. Manolakis, D. Marden, and G. Shaw, “Hyperspectral image processing for automatic target detection applications,” *MIT Lincoln Lab. J.* **14**, 79–116 (2003).
5. D. Manolakis and G. Shaw, “Detection algorithms for hyperspectral imaging applications,” *IEEE Signal Process Mag.* **19**(1), 29–43 (2002).
6. R. Rand, “Exploitation of hyperspectral imagery using adaptive resonance networks,” *Proc. SPIE* **6696**, 66960M (2007).
7. R. Rand and D. Wang, “Oscillatory correlation for scene segmentation of hyperspectral imagery for geospatial analysis,” *Proc. SPIE* **7812**, 78120F (2010).
8. A. Castrodad et al., “Sparse modeling for hyperspectral imagery with LIDAR data fusion for subpixel mapping,” in *2012 IEEE Int. Geoscience and Remote Sensing Symp.*, Munich, Germany (2012).
9. Y. Chen, N. Nasrabadi, and T. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE Trans. Geosci. Remote Sens.* **49**(10), 3973–3985 (2011).
10. J. Li, J. M. Bioscas-Dias, and A. Plaza, “Hyperspectral image segmentation using a new Bayesian approach with active learning,” *IEEE Trans. Geosci. Remote Sens.* **49**(10), 3947–3960 (2011).
11. O. Eches et al., “Adaptive Markov random fields for joint unmixing and segmentation of hyperspectral images,” *IEEE Trans. Image Process.* **22**(1), 5–16 (2013).
12. J. Ham et al., “Investigation of the random forest framework for classification of hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.* **43**(3), 492–501 (2005).
13. C. Therrien, *Decision Estimation and Classification–An Introduction to Pattern Recognition and Related Topics*, John Wiley and Sons, New York (1989).
14. N. Renard and D. Bourennane, “Dimensionality reduction based on tensor modeling for classification methods,” *IEEE Trans. Geosci. Remote Sens.* **47**(4), 1123–1131 (2009).
15. W. Li et al., “Locality-preserving dimensionality reduction and classification for hyperspectral image analysis,” *IEEE Trans. Geosci. Remote Sens.* **50**(4), 1185–1198 (2012).
16. X. Guo et al., “Support tensor machines for classification of hyperspectral remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.* **54**(6), 3248–3264 (2016).
17. R. Rand and D. Keenan, “A spectral mixture process conditioned by Gibbs-based partitioning,” *IEEE Trans. Geosci. Remote Sens.* **39**(7), 1421–1434 (2001).
18. R. Rand and D. Keenan, “Spatially smooth partitioning of hyperspectral imagery using spectral/spatial measures of disparity,” *IEEE Trans. Geosci. Remote Sens.* **41**(6), 1479–1490 (2003).
19. J. Li, J. Bioucas, and A. Plaza, “Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields,” *IEEE Trans. Geosci. Remote Sens.* **50**(3), 809–823 (2012).
20. M. Khodadadzadeh et al., “Spectral-spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization,” *IEEE Trans. Geosci. Remote Sens.* **52**(10), 6298–6314 (2014).
21. M. Fauvel et al., “Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles,” *IEEE Trans. Geosci. Remote Sens.* **46**(11), 3804–3814 (2008).
22. Y. Tarabalka, J. Benediktsson, and J. Chanussot, “Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques,” *IEEE Trans. Geosci. Remote Sens.* **47**(8), 2973–2987 (2009).
23. Y. Zhao, L. Zhang, and S. Kong, “Band-subset-based clustering and fusion for hyperspectral imagery classification,” *IEEE Trans. Geosci. Remote Sens.* **49**(2), 747–756 (2011).
24. A. Romero, C. Gatta, and G. Camps-Valls, “Unsupervised deep feature extraction for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.* **54**(3), 1349–1362 (2016).
25. A. K. Aijazi and P. Checchin, “Segmentation based classification of 3D urban point clouds: a supervoxel approach with evaluation,” *Remote Sens.* **5**(4), 1624–1650 (2013).
26. H. Woo and E. Kang, “A new segmentation method for point cloud data,” *Int. J. Mach. Tools Manuf* **42**(2), 167–178 (2002).
27. K. Fukunaga, “The estimation of the gradient of a density function and its applications in pattern recognition,” *IEEE Trans. Inf. Theory* **21**(1), 32–40 (1975).
28. M. Fashing and C. Tomasi, “Mean shift is a bound optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(3), 471–474 (2005).
29. D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 603–619 (2002).
30. A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from complete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B* **39**(1), 1–38 (1977).
31. Z. Chair and P. Varshney, “Optimal data fusion in multiple sensor detection systems,” *IEEE Trans. Aerosp. Electron. Syst.* **AES-23**, 644–653 (1987).
32. R. Levine and T. Khuon, “Neural net sensor fusion,” MIT Technical Report TR 926, Defense Technical Information Center, 1991, http://www.dtic.mil/docs/citations/ADA241765.
33. R. Levine and T. Khuon, “Neural network for distributed sensor data fusion: the firefly experiment,” *Proc. SPIE* **1611**, 52 (1992).
34. R. Levine and T. Khuon, “Decision-level neural net sensor fusion,” in *Handbook of Statistics*, Vol. **10**, Signal Processing and Its Applications, N. Bose and C. Rao, Eds., Elsevier Science Pub Co. (1993).
35. R. Levine and T. Khuon, “Training set-based performance measures for neural net hypothesis testing,” in *IEEE Joint-Conf. on Neural Networks Symp.*, Baltimore (1992).
36. Y. Levine and T. Khuon, “Training set based performance measures for data-adaptive decisioning systems,” *Proc. SPIE* **1766**, 518 (1992).
37. R. Levine and T. Khuon, “Performance measures for adaptive decisioning systems,” Technical Report TR 927, MIT Lincoln Laboratory, 1991, http://www.dtic.mil/docs/citations/ADA243183.
38. E. Ientilucci, “Leveraging LiDAR data to Aid in hyperspectral image target detection in the radiance domain,” *Proc. SPIE* **8390**, 839007 (2012).

Robert S. Rand received his BS degree in physics from the University of Massachusetts at Lowell. He received his ME degree in engineering physics from the University of Virginia and his PhD in engineering physics from the University of Virginia. His recent efforts have focused on hyperspectral exploitation using feature-based methods and spectral mixture models, as well as the use of 3-D information from PCD to improve the exploitation of hyperspectral imagery. His interests include statistical, physics-based, and bioinspired methods of image segmentation and pattern analysis.

Timothy Khuon received his BS and MS degrees in computer science/electrical engineering from Illinois Institute of Technology and additional graduate education at Massachusetts Institute of Technology (MIT). He has performed research at NGA, JHU, Applied Physics Laboratory, MIT Lincoln Laboratory, and Bell Laboratories. His research interests are in space-time-adaptive-processing, coherent change detection, and neural net sensor fusion. His technical publications include multiple papers with National Academy of Sciences, IEEE, and SPIE, MIT technical reports, and a *Bose Statistics Handbook* chapter. He was elected to Sigma-Xi MIT Chapter.

Eric Truslow received his BS degree in electrical engineering from Union College in 2010. He received his MS degree in electrical engineering from Northeastern University in 2012, and his PhD in electrical engineering from Northeastern University. Currently, he is a staff member at MIT-Lincoln Laboratory. His PhD thesis investigated the performance of hyperspectral chemical detection systems. His research interests include hyperspectral imagery, detection theory, machine learning, and signal processing. His work continues to involve performance analysis of signal detection algorithms using hyperspectral imagery.