Hyperspectral image compressive projection algorithm

Joseph P. Rice and David W. Allen

Optical Technology Division
National Institute of Standards and Technology, Gaithersburg, MD 20899

ABSTRACT

We describe a compressive projection algorithm and experimentally assess its performance when used with a Hyperspectral Image Projector (HIP). The HIP is being developed by NIST for system-level performance testing of hyperspectral and multispectral imagers. It projects a two-dimensional image into the unit under test (UUT), whereby each pixel can have an independently programmable arbitrary spectrum. To efficiently project a single frame of dynamic realistic hyperspectral imagery through the collimator into the UUT, a compression algorithm has been developed whereby the series of abundance images and corresponding endmember spectra that comprise the image cube of that frame are first computed using an automated endmember-finding algorithm such as the Sequential Maximum Angle Convex Cone (SMACC) endmember model. Then these endmember spectra are projected sequentially on the HIP spectral engine in sync with the projection of the abundance images on the HIP spatial engine, during the single-frame exposure time of the UUT. The integrated spatial image captured by the UUT is the endmember-weighted sum of the abundance images, which results in the formation of a datacube for that frame. Compressive projection enables a much smaller set of broadband spectra to be projected than monochromatic projection, and thus utilizes the inherent multiplex advantage of the HIP spectral engine. As a result, radiometric brightness and projection frame rate are enhanced. In this paper, we use a visible breadboard HIP to experimentally assess the compressive projection algorithm performance.

Keywords: calibration, hyperspectral, imaging, micromirror, multispectral, projector, remote sensing, spectroscopy, test and evaluation, validation

1. INTRODUCTION

As described in our previous papers, we are developing a Hyperspectral Image Projector (HIP) to enable performance testing and data validation of multispectral and hyperspectral imaging systems, including not only the hardware but calibration and data product retrieval algorithms.[1-7] A prototype of the HIP operating in the visible-near-infrared (VNIR), short-wavelength infrared (SWIR), and mid-wavelength infrared (MWIR) spectral ranges is currently being designed.[8] In parallel with this effort, we are developing control algorithms and software, and testing them using a visible-band breadboard version of the HIP. In this paper, we describe an algorithm for efficient image projection with the HIP, and provide results for a test datacube that was projected and measured by an imaging spectrometer. First we review the HIP breadboard hardware and point out a few changes from previous versions, and describe the compressive projection algorithm. We then discuss the imaging spectrometer used for the measurements, and its calibration. A test datacube that we computer-generated is described. Finally, we present the results of projecting the test datacube by the HIP and measuring it with the imaging spectrometer.

Correspondence: joe.rice@nist.gov

Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV,
edited by Sylvia S. Shen, Paul E. Lewis, Proc. of SPIE Vol. 7334, 733414 · © 2009 SPIE
CCC code: 0277-786X/09/$18 · doi: 10.1117/12.818844

2. HYPERSPECTRAL IMAGE PROJECTOR

The current version of the breadboard HIP hardware is shown in Fig. 1. It consists of a spectral engine coupled to a spatial engine. The spectral engine produces a series of programmable spectra. The spatial engine maps them in the correct proportions, through a collimator, into a unit under test (UUT), such as the charge-coupled device (CCD) camera-based imaging spectrometer shown on the translation stage in Fig. 1. Two digital micromirror devices (DMD) are used: one in the spectral engine (DMD1), and another in the spatial engine (DMD2), as we have described previously.[1] One improvement over the previous versions of this HIP breadboard is the addition of a custom-designed, high-performance, multi-element collimator lens. The collimator projects the image displayed on DMD2 at infinity, such that when the UUT is focused at infinity, the image displayed on DMD2 is focused onto the UUT sensor focal plane (here the CCD). The collimator is specifically designed with an accessible exit pupil having appropriate relief to enable the HIP exit pupil to be coincident with the UUT entrance pupil. This arrangement is intended to avoid vignetting.

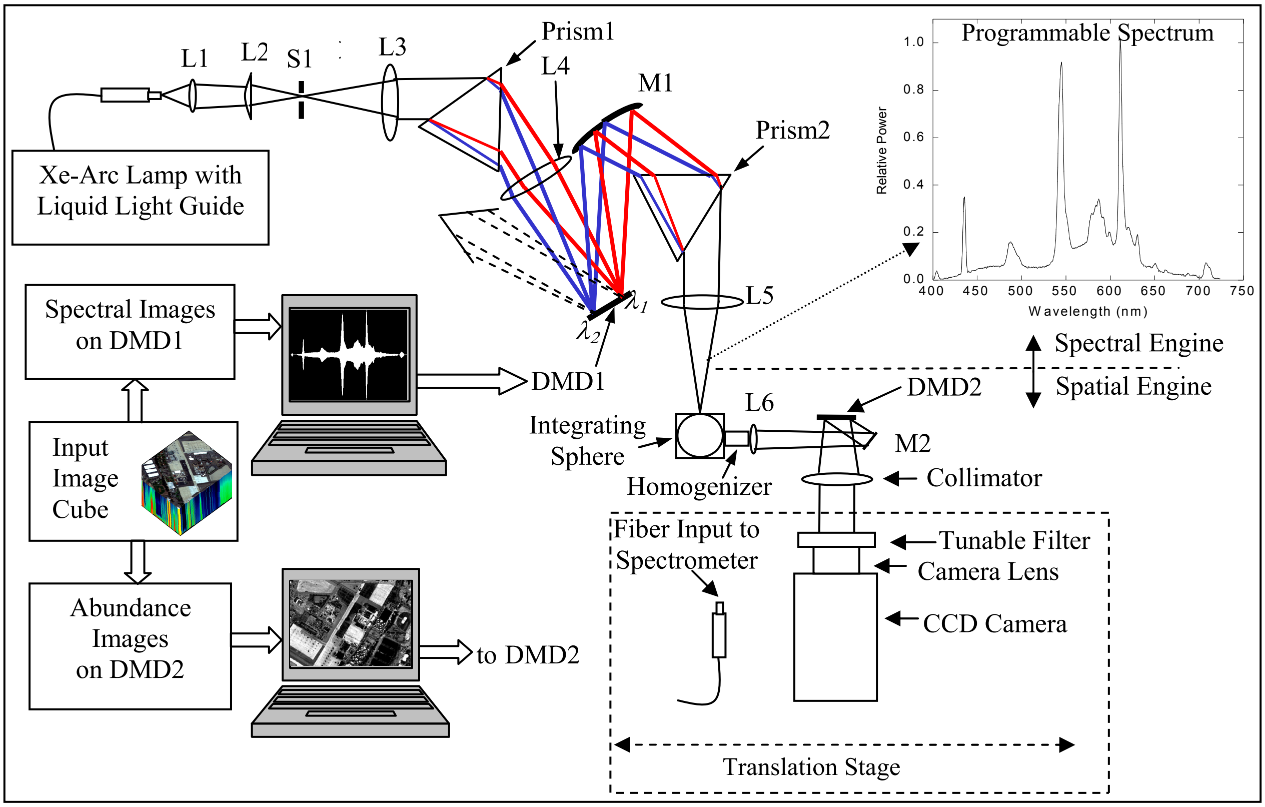

Figure 1. Schematic of the visible-breadboard version of the Hyperspectral Image Projector being used for software development. The spectral images that shape the endmember spectra are displayed on DMD1 in sync with the display of the corresponding abundance images on DMD2. The DMDs are shown edge-on in this schematic. The full set of endmember spectra and abundance images are displayed within the integration time of the imaging spectrometer under test, which then measures the projected hyperspectral image. (Though two computers are shown here for illustration purposes, in practice only one computer is used). L=lens. M=mirror. S=slit.

To implement the compressive projection algorithm, an input image cube (one frame of an image cube time sequence, for example) is first broken down into its endmembers, consisting of eigenspectra and corresponding abundance images. The HIP spectral engine is basically a programmable spectrum generator. It converts the xenon-arc lamp spectral profile into the desired eigenspectrum by displaying a binary spectral image on DMD1 that shapes the spectral profile. It is analogous to the color filter in a conventional single-chip Digital Light Processing (DLP$^{®}$)* projector, except that it can provide programmable spectra on demand. While the HIP spectral engine provides an eigenspectrum, the HIP spatial engine displays, on DMD2, the abundance image corresponding to that spectrum. The gray scale value of each pixel in the two-dimensional abundance image indicates the relative proportion of the corresponding eigenspectrum required by the image cube for that pixel. Upon projection of the entire set of eigenspectra and corresponding abundance images in sync with each other during the integration time of the sensor, the sensor will have measured the entire datacube for that frame. This mode of operation is exactly analogous to the rapid serial projection of red-green-blue spectra to simulate color images common in DLP display technologies. However, the HIP is not limited to color – it can provide programmable spectra. Also, it is not limited to the visible spectral range.

Both of the HIP DMDs have drive electronics that are configured to display their respective images in sync with a 5 V trigger pulse supplied by a pulse sequence generator (PSG). The PSG, in turn, is configured to supply a series of *N* such trigger pulses when triggered by the UUT sync pulse. The UUT provides a sync pulse each time it begins a new exposure, and the length of the pulse indicates the exposure duration. For the work described here, *N* successive PSG trigger pulses were generated for each UUT sync pulse, and the PSG was used to trigger both DMD1 and DMD2 simultaneously. Successive PSG pulses triggered the display of the successive eigenspectra on DMD1 and corresponding abundance images on DMD2. The frequency of the trigger pulses was adjusted such that the complete set of the *N* endmembers that comprise the image cube were displayed within the integration time of the UUT.

3. TEST INSTRUMENTS AND CALIBRATION

For the measurements described below, the translation stage shown in Fig. 1 was configured with two test instruments that could alternately measure the HIP output beam. One was an un-calibrated imaging spectrometer, and the other a well-calibrated non-imaging spectrometer. A precision servo motor on the stage enabled repeatable horizontal positioning of either instrument without the need for re-alignment.

The imaging spectrometer is based on a 2-dimensional staring array and tunable filter. It is a Princeton Instruments PIXIS 1024BR 16-bit CCD camera with a Nikon 85 mm camera lens and a Cambridge Research Instruments liquid crystal tunable filter (LCTF). The CCD has a 1024 × 1024 array of 13 μm × 13 μm pixels, was operated at a temperature of -70 °C, and provides images at an intensity resolution of 16 bits. The CCD camera has a mechanical shutter for exposure control. The camera lens stop was set to f/8 and it was focused at infinity and aligned such that the position of its entrance pupil coincided with the exit pupil of the HIP 100 mm, f/2.8 collimator. Thus the 14 mm × 10.5 mm HIP spatial engine DMD was de-magnified by a factor of 0.85, thereby fitting on the CCD with margin. The LCTF operates nominally over the spectral range of 400 nm to 720 nm. Only the subset 460 nm to 680 nm was used here, and here the bandwidth ranged from about 6 nm to 12 nm. We developed software to enable automated measurements of the HIP-projected datacube using this LCTF/CCD imaging spectrometer by collecting a series of 2-dimensional images, each at a different wavelength setting of the LCTF.

The non-imaging spectrometer is an Analytical Spectral Devices (ASD) FieldSpec3, which is fiber-coupled and capable of spectral radiance measurements from 350 nm to 2500 nm. Only its bare input fiber end was mounted on the translation stage, as shown in Fig. 1; the fiber transmitted the input light to a fixed platform about a meter away where the spectrometer itself was located. Also, whenever the ASD spectrometer was used in the measurements described below, the 85 mm camera lens and, for certain tests, the LCTF, were removed from the LCTF/CCD and kinematically mounted in front of the ASD spectrometer bare input fiber in the same position (relative to the HIP) that they occupied when used with the CCD camera. The de-magnified image of the DMD was then accessible, and the ASD spectrometer bare fiber input was positioned in the center of this image, at the same location where the CCD was when it was used. Prior to using it, the ASD spectrometer was calibrated at NIST for radiance responsivity with its bare fiber measuring the output of a lamp-illuminated integrating sphere using methods consistent with a < 5 % (k=1) uncertainty.

A careful absolute radiometric calibration of the LCTF/CCD and the HIP has not been performed yet. However, a quick set of relative spectral and spatial measurements were performed to roughly (within 10 % k=1 uncertainty) flat-field the spectra and images below.

First, all DMD mirrors of the HIP spatial engine were turned on, and the HIP spectral engine was calibrated for wavelength against the ASD spectrometer using methods described previously.[1] This provided a look-up table (LUT) giving the wavelength corresponding to each DMD1 column number, which was used for all subsequent measurements.

Then the full-intensity (here meaning both DMD1 and DMD2 have all-mirrors on) spectrum was measured, again as described previously.[1] From this, the spectral range of useful (> 50 % of the peak) radiance from the HIP breadboard was determined to be between 460 nm and 680 nm. Next, a relative intensity calibration was performed on the HIP spectral engine, turning on successively more rows of DMD1 and measuring the spectrum with the ASD spectrometer (again with all DMD2 mirrors on). The relative intensity calibration data and the wavelength calibration LUT were inputs into the spectral matching algorithm, which is an improved version of the one described previously.[1] The spectral matching algorithm automatically determines the spectral image on the DMD that will provide a measured spectrum that most closely resembles the target spectrum. It was run on all eight eigenspectra from the test datacube (to be described below in Section 4) and the spectral images were saved to the HIP computer disk for later recall.

The LCTF spectral transmittance was measured by mounting the LCTF on the front of the 85 mm camera lens, turning all HIP mirrors on (both DMD1 and DMD2), and measuring the spectrum using the ASD spectrometer at each band of the LCTF. The measurements were made at each of the 111 LCTF bands used, from 460 nm to 680 nm, every 2 nm, as these were the bands chosen for the datacube. The LCTF transmittance for each band was then computed by normalizing each of these spectra with the full-intensity spectra measured above, since the full intensity spectra effectively constitute the "filter out" phase of the transmittance measurement. Because the bandwidths vary spectrally, the band-integrated LCTF transmittance was computed for each band.

The unfiltered CCD radiance response was measured with the CCD and lens at the HIP output, but omitting the LCTF. For this measurement, again all DMD2 mirrors were on, but now the HIP spectral engine was operated in monochromatic mode, stepping through all 111 bands from 460 nm to 680 nm at 2 nm spacing, and providing monochromatic spectral radiance to the CCD at each band. At each band, the CCD response was measured by capturing the full image. Then the spectral radiance of each of these monochromatic bands was measured using the ASD spectrometer (including the 85 mm lens but not the LCTF). The bandwidths of these monochromatic spectra varied from 10 nm to 30 nm, mainly because of the dispersion of Prism1 in the HIP spectral engine. The CCD response images were first dark count corrected, then divided, for each band, by the corresponding band-integrated radiance from ASD spectral radiance measurements, to provide a spectrally-corrected CCD (band-integrated) radiance responsivity calibration. Since the ASD only measured the center region of HIP output image, this calibration is only accurate in the absolute sense for the center pixels of the CCD. However, a characterization of the spatial non-uniformity of the HIP output radiance would enable it to be applied to the other CCD pixels. Finally the filtered CCD radiance responsivity was computed by multiplying the LCTF transmittance results (from the previous paragraph) by the unfiltered CCD radiance responsivity. This resulted in a set of 111 images, one for each band, that were used to calibrate (in a relative spectral sense) the datacube images measured below. Because these calibration images include the spatial non-uniformity of the HIP output and because the datacube images include the same spatial non-uniformity, and because the calibration consists of dividing (at each band) one of these images by the other, the non-uniformity of the HIP spatial output tends to cancel out, providing some degree of flat-fielding.

4. TEST DATACUBE

A test datacube was generated for use in this study. It consisted of N = 8 sinusoidal eigenspectra as shown Fig. 2, distributed over the 2-D spatial scene as shown in Table 1. The sinusoidal spectral functions were computed at M = 111 discrete wavelengths λ_m at equal spacing of $\Delta\lambda$ = 2 nm, spanning the range λ_1 = 460 nm to λ_M = 680 nm, as

$$G_i(\lambda_m) = \frac{1}{2}\left[1 - \cos\left(\frac{m-1}{M-1} i\pi\right)\right], \; m = 1 \text{ to } M, \; i = 1 \text{ to } N. \tag{1}$$

This particular set of sinusoidal spectra was chosen since they are mathematically orthogonal to each other, and because they form the beginning of a Fourier series. These properties provide a degree of mathematical simplicity for this test datacube compared to other possible test scenes or to natural scenes. The 1024 × 768 spatial image consists of an 8 × 8 array of cells, each having dimensions of 128 projector pixels horizontally and 96 projector pixels vertically. Different spectra were mapped to the cells according to Table 1. All pixels of a given cell have the same spectra. Cells along the northwest (NW) to southeast (SE) diagonal contain the eigenspectra. Off-diagonal cells mix eigenspectra, the more off-diagonal cells containing more spectra. The image is symmetrical about the NW-SE diagonal.

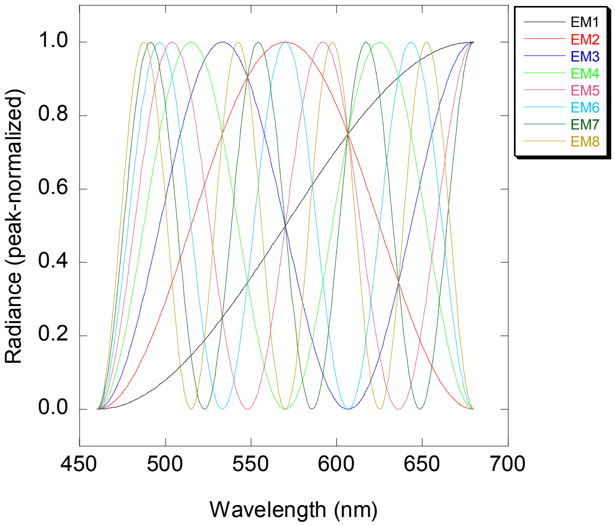

Figure 2. The eight sinusoidal eigenspectra used for the HIP test datacube, generated per Eq. (1). The plots are labeled according to endmember (EM) number, *i*. (for color image see electronic version)

$G_1(\lambda)$	$\frac{1}{2}\sum_{i=1}^{2}G_i(\lambda)$	$\frac{1}{3}\sum_{i=1}^{3}G_i(\lambda)$	$\frac{1}{4}\sum_{i=1}^{4}G_i(\lambda)$	$\frac{1}{5}\sum_{i=1}^{5}G_i(\lambda)$	$\frac{1}{6}\sum_{i=1}^{6}G_i(\lambda)$	$\frac{1}{7}\sum_{i=1}^{7}G_i(\lambda)$	$\frac{1}{8}\sum_{i=1}^{8}G_i(\lambda)$
$\frac{1}{2}\sum_{i=1}^{2}G_i(\lambda)$	$G_2(\lambda)$	$\frac{1}{2}\sum_{i=2}^{3}G_i(\lambda)$	$\frac{1}{3}\sum_{i=2}^{4}G_i(\lambda)$	$\frac{1}{4}\sum_{i=2}^{5}G_i(\lambda)$	$\frac{1}{5}\sum_{i=2}^{6}G_i(\lambda)$	$\frac{1}{6}\sum_{i=2}^{7}G_i(\lambda)$	$\frac{1}{7}\sum_{i=2}^{8}G_i(\lambda)$
$\frac{1}{3}\sum_{i=1}^{3}G_i(\lambda)$	$\frac{1}{2}\sum_{i=2}^{3}G_i(\lambda)$	$G_3(\lambda)$	$\frac{1}{2}\sum_{i=3}^{4}G_i(\lambda)$	$\frac{1}{3}\sum_{i=3}^{5}G_i(\lambda)$	$\frac{1}{4}\sum_{i=3}^{6}G_i(\lambda)$	$\frac{1}{5}\sum_{i=3}^{7}G_i(\lambda)$	$\frac{1}{6}\sum_{i=3}^{8}G_i(\lambda)$
$\frac{1}{4}\sum_{i=1}^{4}G_i(\lambda)$	$\frac{1}{3}\sum_{i=2}^{4}G_i(\lambda)$	$\frac{1}{2}\sum_{i=3}^{4}G_i(\lambda)$	$G_4(\lambda)$	$\frac{1}{2}\sum_{i=4}^{5}G_i(\lambda)$	$\frac{1}{3}\sum_{i=4}^{6}G_i(\lambda)$	$\frac{1}{4}\sum_{i=4}^{7}G_i(\lambda)$	$\frac{1}{5}\sum_{i=4}^{8}G_i(\lambda)$
$\frac{1}{5}\sum_{i=1}^{5}G_i(\lambda)$	$\frac{1}{4}\sum_{i=2}^{5}G_i(\lambda)$	$\frac{1}{3}\sum_{i=3}^{5}G_i(\lambda)$	$\frac{1}{2}\sum_{i=4}^{5}G_i(\lambda)$	$G_5(\lambda)$	$\frac{1}{2}\sum_{i=5}^{6}G_i(\lambda)$	$\frac{1}{3}\sum_{i=5}^{7}G_i(\lambda)$	$\frac{1}{4}\sum_{i=5}^{8}G_i(\lambda)$
$\frac{1}{6}\sum_{i=1}^{6}G_i(\lambda)$	$\frac{1}{5}\sum_{i=2}^{6}G_i(\lambda)$	$\frac{1}{4}\sum_{i=3}^{6}G_i(\lambda)$	$\frac{1}{3}\sum_{i=4}^{6}G_i(\lambda)$	$\frac{1}{2}\sum_{i=5}^{6}G_i(\lambda)$	$G_6(\lambda)$	$\frac{1}{2}\sum_{i=6}^{7}G_i(\lambda)$	$\frac{1}{3}\sum_{i=6}^{8}G_i(\lambda)$
$\frac{1}{7}\sum_{i=1}^{7}G_i(\lambda)$	$\frac{1}{6}\sum_{i=2}^{7}G_i(\lambda)$	$\frac{1}{5}\sum_{i=3}^{7}G_i(\lambda)$	$\frac{1}{4}\sum_{i=4}^{7}G_i(\lambda)$	$\frac{1}{3}\sum_{i=5}^{7}G_i(\lambda)$	$\frac{1}{2}\sum_{i=6}^{7}G_i(\lambda)$	$G_7(\lambda)$	$\frac{1}{2}\sum_{i=7}^{8}G_i(\lambda)$
$\frac{1}{8}\sum_{i=1}^{8}G_i(\lambda)$	$\frac{1}{7}\sum_{i=2}^{8}G_i(\lambda)$	$\frac{1}{6}\sum_{i=3}^{8}G_i(\lambda)$	$\frac{1}{5}\sum_{i=4}^{8}G_i(\lambda)$	$\frac{1}{4}\sum_{i=5}^{8}G_i(\lambda)$	$\frac{1}{3}\sum_{i=6}^{8}G_i(\lambda)$	$\frac{1}{2}\sum_{i=7}^{8}G_i(\lambda)$	$G_8(\lambda)$

Table 1. Spatial map of the spectra used in the generated HIP test datacube. The NW-SE diagonal cells contain the eigenspectra. Off-diagonal cells, such as the one shaded, contain a normalized sum of a subset of the eigenspectra, and the number of eigenspectra included in the sum increases as the distance from the NW-SE diagonal increases. This test datacube is symmetric about the NW-SE diagonal. (The subscript *m* on λ_m has been suppressed to save space in this table.)

This test datacube was generated numerically using customized LabVIEW software being developed by us as part of the HIP. Two equivalent forms of the datacube were computed. The first form was as a set of gray-scale Tagged Image File Format (TIFF) images, each image representing one of the monochromatic bands. There were M = 111 such images. They were loaded into Environment for Visualization of Images (ENVI) software as metafiles, then each was labeled by its corresponding monochromatic wavelength in a header file, and the data cube was saved as a standard ENVI image file. The datacube in this form represents what an ideal imaging spectrometer would measure if it were to view the test datacube projected by an ideal HIP. Three of the 111 bands were used in ENVI to render the red-green-blue (RGB) image shown below in Fig. 4.

The second form was also computed by our LabVIEW software as the set of gray-scale abundance images. These were computed directly using the definition in Table 1, and are shown in Fig. 3. There is a one-to-one correspondence between these N = 8 abundance images and the 8 eigenspectra. The gray level of any pixel in each of the abundance images represents how much of the corresponding eigenspectra is contained in the datacube at that pixel. A sum-to-unity condition was applied during the abundance image generation, such that, for any given pixel, the sum of the gray levels from each of the 8 abundance images will equal 255. (For an 8-bit image, the values range from 0 (black) to 255 (white)). A linear contrast stretch was maintained during the generation of these images, since were used directly by the HIP spatial engine to project the datacube as described in the next section.

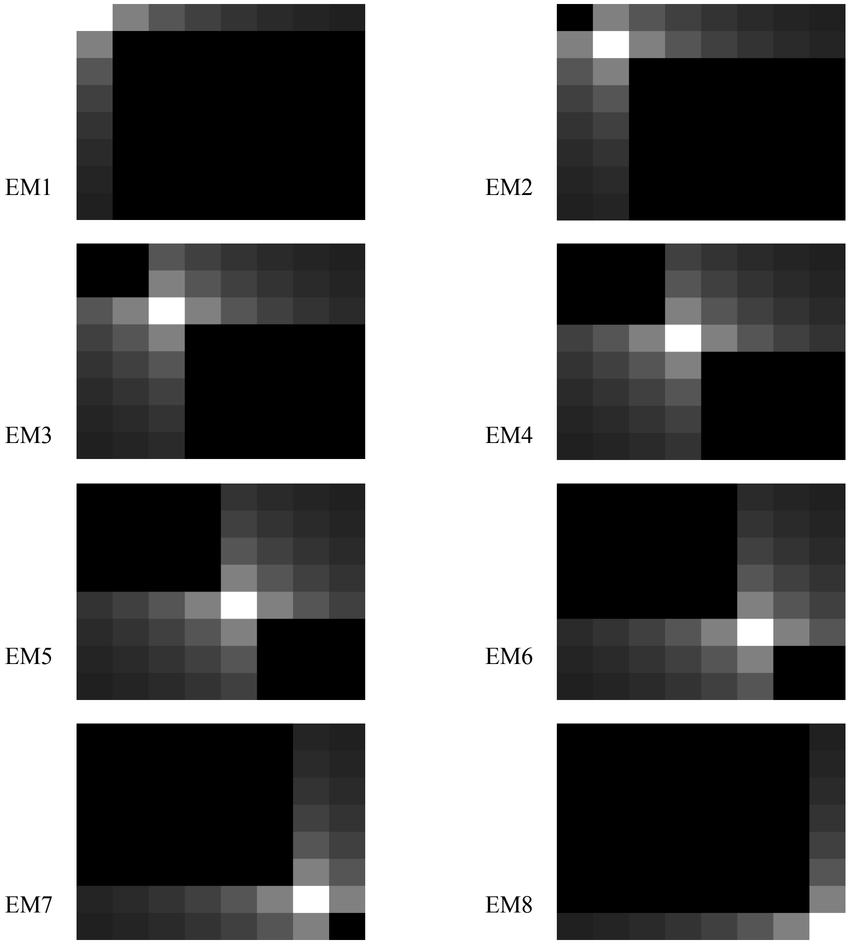

Figure 3. Abundance images corresponding to the test datacube endmember eigenspectra of Fig. 2, computed from the definition in Table 1, and used by the HIP for testing the compressive projection algorithm.

5. COMPRESSIVE PROJECTION RESULTS

At this point we could use the HIP to project the set of 111 monochromatic bands one after the other. The intensity of each of these projected monochromatic bands is already directly encoded in the 111 monochromatic band images that constitute our example datacube. All 111 monochromatic spectra would need to be projected by the HIP spectral engine within the integration time of the UUT, in sync with the projection of the corresponding monochromatic band images on the HIP spatial engine. This would be using the HIP in what we call monochromatic projection mode. However, this would not be the most efficient projection algorithm in terms of projector brightness level, since most of the time an unnecessarily large proportion of the available light would be rejected rather than projected.

It is radiometrically more efficient to use the HIP with a broadband projection algorithm, where the multiplex advantage inherent within HIP can be utilized. This is what we call compressive projection mode. In this mode, we project a smaller set of broadband eigenspectra, sampled appropriately for each spatial pixel according to a corresponding set of abundance images. Since we will (probably – just how much is scene dependent) have fewer spectra to project, the UUT can spend a larger fraction of its integration time on each one. Also, since the spectra are generally broadband, the UUT spectral bands are stimulated in a more parallel manner, as opposed to being stimulated in series as in the monochromatic projection mode. This is the multiplex advantage of compressive projection.

For real measured scenes, where one might be given only monochromatic band images as an input datacube for projection on the HIP, we first we have to determine the set of eigenspectra and corresponding abundance images from the monochromatic band image cube. In other work,[1,9] we have used the Sequential Maximum Angle Convex Cone (SMACC) algorithm.[10] ENVI now includes SMACC as a standard routine. This algorithm automatically finds endmember spectra from the data. The spectra from a few pure pixels match the endmember spectra. The spectra of the vast majority of pixels in the scene will not match any one endmember spectrum alone, but will match a linear combination of the endmember spectra. For a given endmember, the corresponding image that gives the proportion of that endmember spectrum included in each pixel is called the abundance image. The ENVI SMACC routine not only finds the endmember eigenspectra, but it also determines the abundance image corresponding to each endmember spectrum. We note that the SMACC endmember spectra may not necessarily correspond with material spectral library data, but it does not seem that they need to for implementing the compressive projection algorithm: we seek only a dimensional reduction of the data for more efficient projection.

For computer-generated scenes we may be able to generate the required eigenspectra and abundance images when rendering the monochromatic band images. This was indeed the case for our test datacube described in the last section, so it was rather straightforward to implement compressive projection in this case.

We projected and measured the test datacube as follows. Using our LabVIEW HIP control software, we loaded the 8 spectral images that resulted from matching to the eigenspectra of Fig. 2 into the HIP spectral engine DMD on-board memory, and we loaded the 8 gray scale abundance images of Fig. 3 into HIP spatial engine DMD on-board memory. Then we set the LCTF/CCD to a 1 s integration time and used its output sync signal to trigger the PSG as described in Section 2 above. The PSG provided 8 equally-spaced trigger pulses to the HIP DMDs during each 1 s CCD integration. Pulse-width modulation at a bit depth of 8 bits was used on the spatial engine to display the gray scale of the abundance images, and the spectral engine images were binary (1 bit). The 8 PSG pulses synchronized the display of all endmembers during the integration time of the CCD, which performed the required photon summation at each pixel.

After each 1 s integration, the CCD took about 0.6 s to read out the measured monochromatic band image corresponding to the wavelength setting of the LCTF. During the CCD readout time the LCTF was tuned to the next band, which took only a few ms, and the measured monochromatic band image was saved to the HIP control computer disk. The entire datacube was measured by having the LCTF tune through one band at a time, from 460 nm to 680 nm, in 2 nm steps. Each monochromatic band image was saved as a 16 bit TIFF image.

To perform the analysis, each of the 111 measured datacube monochromatic band images was first dark-count corrected, cropped to the DMD size, and calibrated by dividing (using the automated HIP LabVIEW software) by the corresponding band of the 111 calibration images that were described at the end of Section 3. This provided first-order relative spectral flattening and relative spatial flat-fielding. In principle it could also provide absolute radiometric calibration, but there are several characterization measurements that will need to be performed in order to provide accuracy. Then the calibrated 111 TIFF images were loaded into ENVI as a metafile and saved as an ENVI image file. The measured data cube was rendered as a red-green-blue (RGB) image by ENVI, as shown in Fig. 5 below. This is to be compared with Fig. 4, which shows the original generated test datacube, rendered the same way by ENVI. Spectral profiles of a sampling of the test scene cells were also pulled out using ENVI and compared with the target spectra, and these are shown in Fig. 6 and Fig. 7 below. All spectral profiles shown have been smoothed by first re-sampling to a 10 nm grid, then by re-sampling again back to the original 2 nm grid, and they have been normalized to the value at 556 nm for ease of comparison.

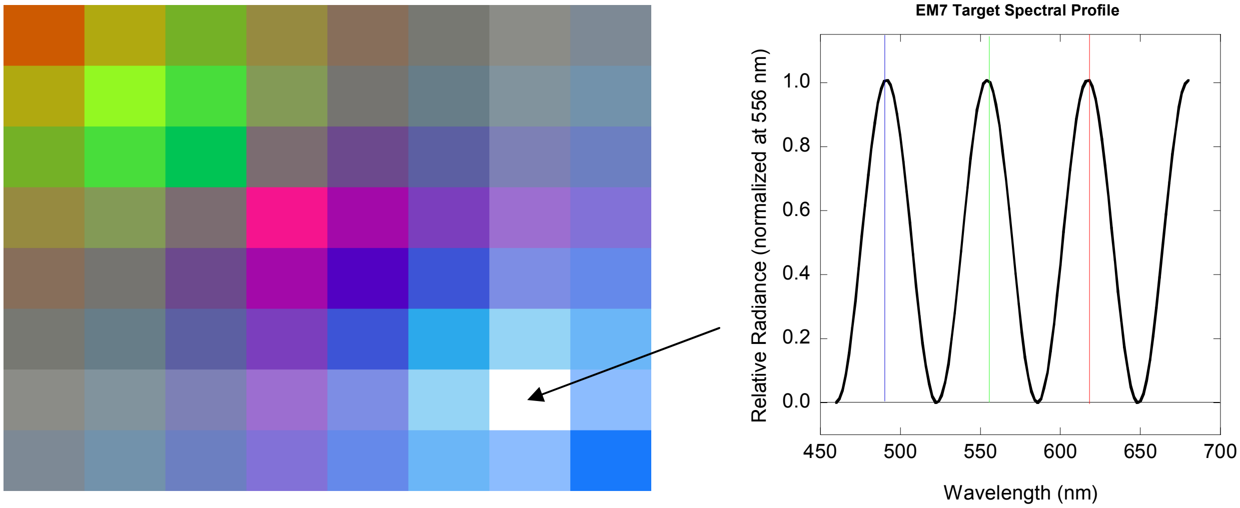

Figure 4. RGB image of the generated test datacube, as rendered by ENVI with the assignments marked in the spectral profile: R=618 nm, G=556 nm, B=490 nm. The spectral profile is a representative spectrum from one pixel in the white cell of the image.

(for color image see electronic version)

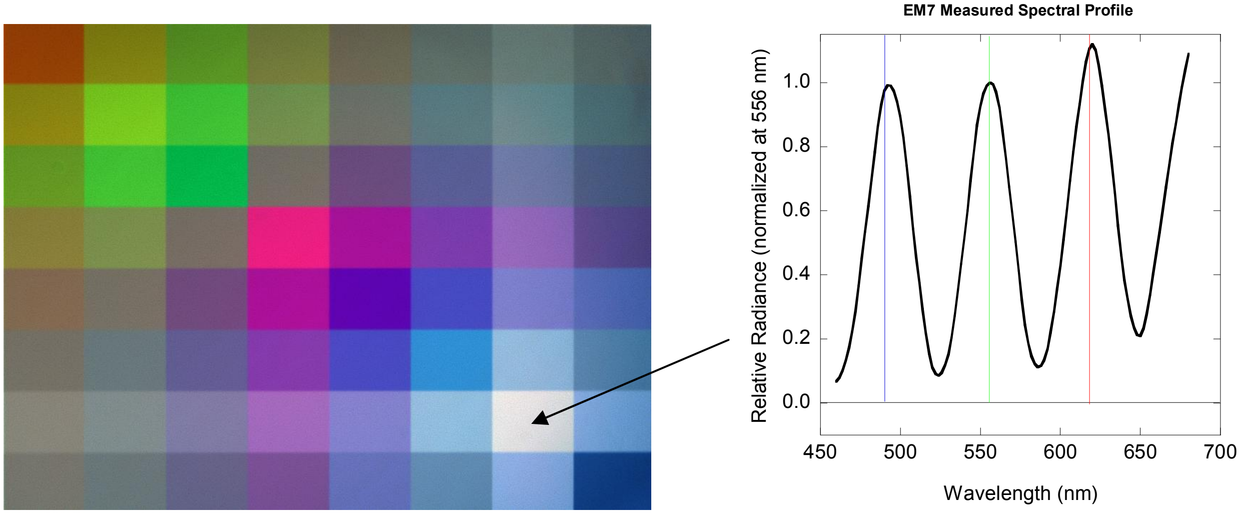

Figure 5. RGB image of the test datacube as projected by the HIP visible-band breadboard and measured by the LCTF/CCD imaging spectrometer. This image was rendered by ENVI with the same RGB assignments as used in Fig. 4, and the spectral profile is again from the white cell of the image. (for color image see electronic version)

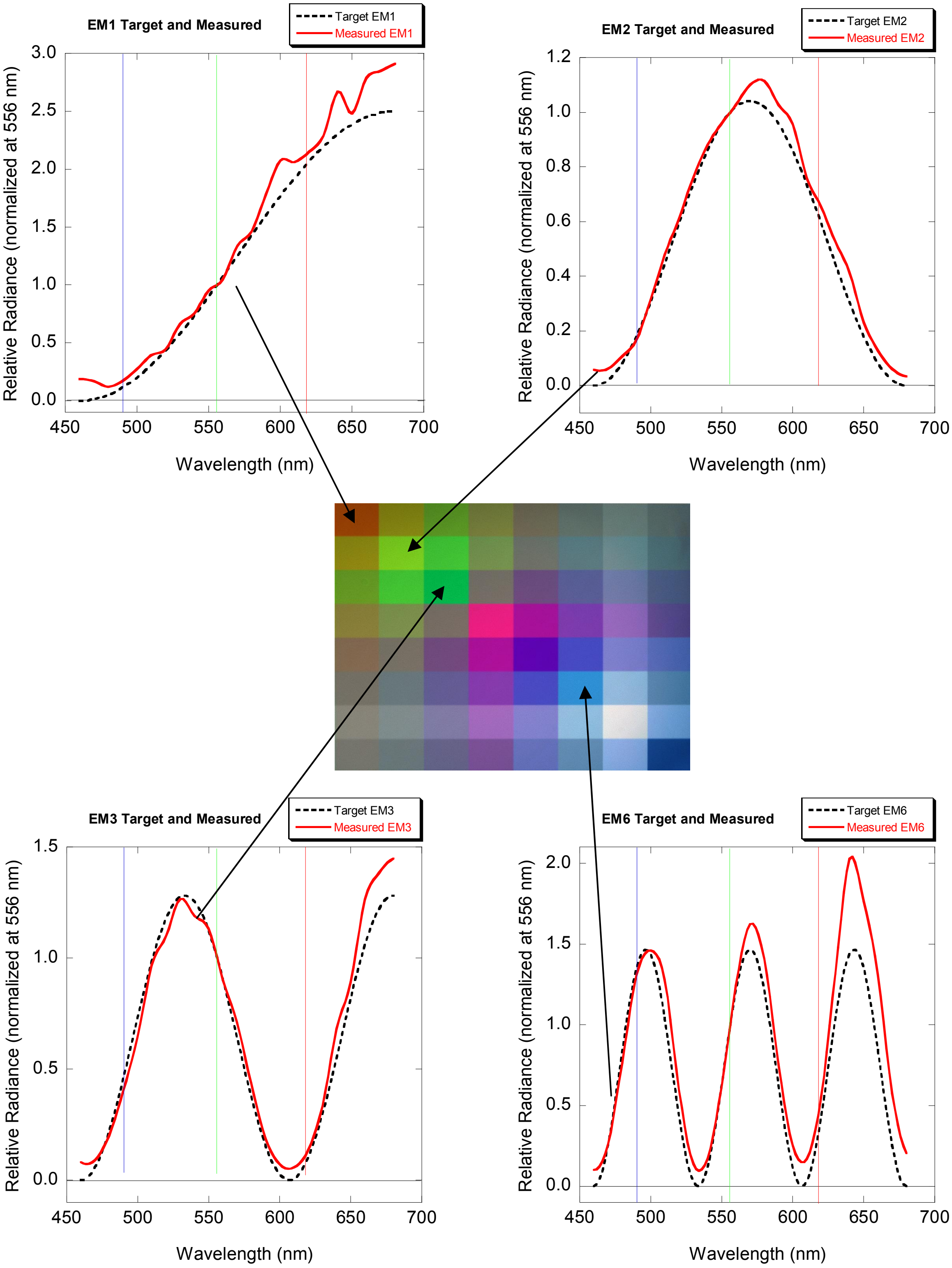

Figure 6. Spectral profiles for selected diagonal test datacube cells. Target spectra are from the computer-generated test datacube, and measured spectra are from the HIP-projected test datacube as measured by the LCTF/CCD imaging spectrometer. For ease of comparison, these spectra have been normalized to the value at 556 nm. (for color image see electronic version)

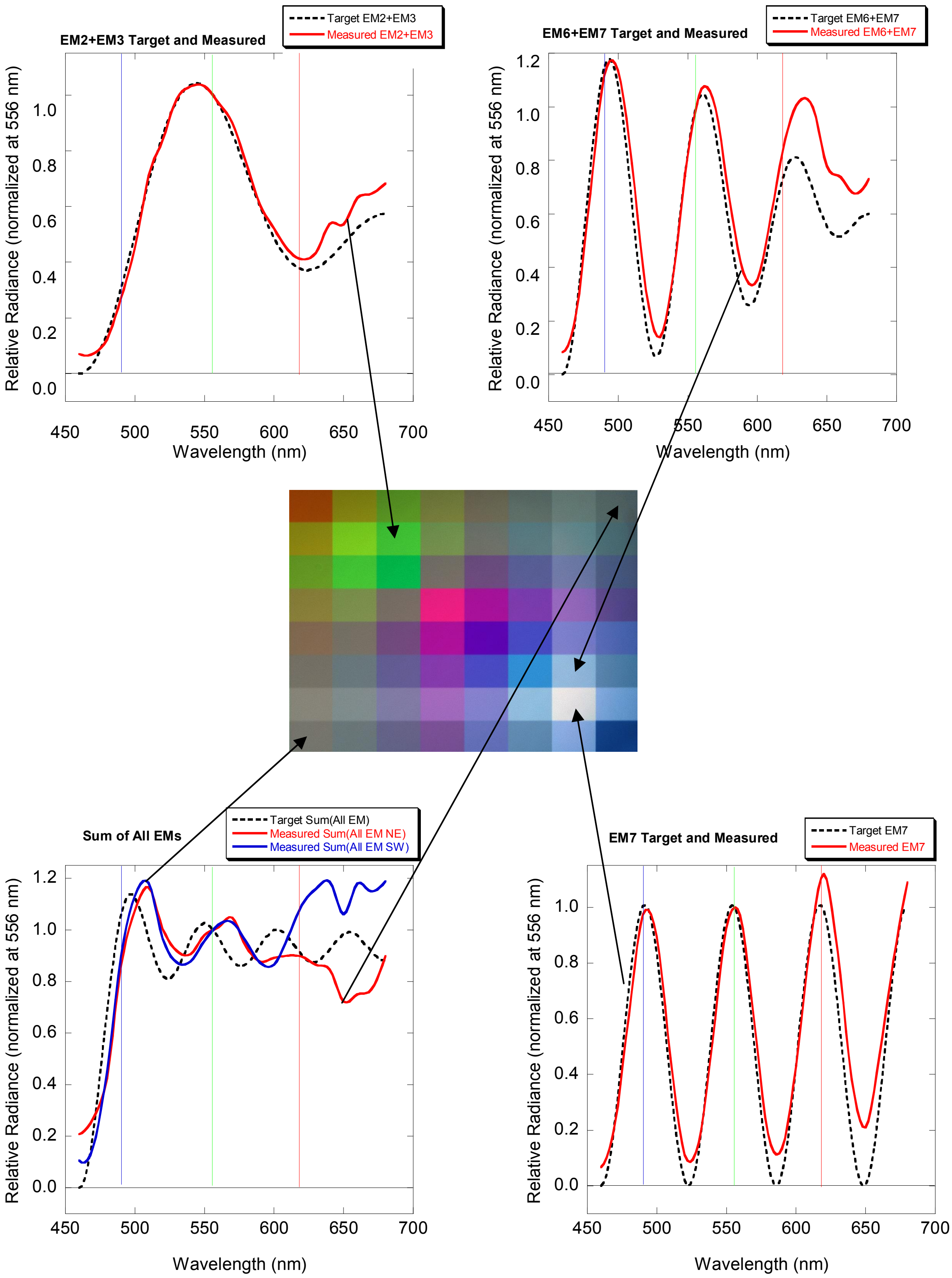

Figure 7. Spectral profiles for selected off-diagonal test datacube cells as indicated by the arrows and for EM7. Target spectra are from the computer-generated test datacube, and measured spectra are from the HIP-projected test datacube as measured by the LCTF/CCD imaging spectrometer. For ease of comparison, these spectra have been normalized to the value at 556 nm.

6. DISCUSSION

The compressive projection algorithm seems to be able to accurately mix the spectra, as seen by the low-ordered off-diagonal elements in Fig. 7, which was the purpose of this end-to-end test at this early stage in the HIP development. Several hardware and calibration issues exist, partly with the HIP and partly with the imaging spectrometer. For example, there is noticeable spatial stray light from the HIP and dark counts from the CCD, and these have not been well-corrected. This may explain why the measured spectra do not go to zero where they should. There is also spectral stray light and extra intensity coming from the red end of the spectral engine DMD. There are many small known improvements to the HIP design that could help with this, and are being included in the next-generation prototype design. There is also vignetting in the spatial engine from mirror M2, for which there are known solutions yet to be implemented. Finally, the simple relative calibration procedure implemented for the imaging spectrometer could still leave residual spectral structure, and there has been no spectral or spatial stray-light correction for this instrument as well.

7. SUMMARY

We have developed a compressive projection algorithm for the HIP and used it to project and measure a simple test datacube with the visible-band breadboard HIP hardware. The results show image quality and spectral fidelity consistent with the current level of optical design of the hardware and the calibration of the test instruments, and are good enough to conclude that the compressive projection algorithm works successfully.

ACKNOWLEDGEMENTS

This work was funded in part by the Department of Defense (DOD) Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program through contract number MIPR8DDAT3A622, and in part by the NIST Office of Law Enforcement Standards. We also thank Bob Saunders and Steve Brown from NIST for providing their ASD LabVIEW software, and Bob Saunders for performing an integrating-sphere calibration of the HIP ASD spectrometer.

[*]Note: References are made to certain commercially available products in this paper to adequately specify the experimental procedures involved. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that these products are the best for the purpose specified.

REFERENCES

[1] Rice, J. P., Brown, S. W., Neira, J. E., and Bousquet, R. R., "A hyperspectral image projector for hyperspectral imagers," *Proc. SPIE* **6565**, 65650C (2007).

[2] Brown, S. W., Myers, B., Barnes, R. A., and Rice, J. P., “Characterization of Earth observing satellite instruments for response to spectrally and spatially variable scenes,” *Proc. SPIE* **6677**, 667705 (2007).

[3] Rice, J. P., Brown, S. W., and Neira, J. E., “Development of hyperspectral image projectors,” *Proc. SPIE* **6297**, 629701 (2006).

[4] Brown, S. W., Rice, J. P., Neira, J. E., and Johnson, B. C., “Hyperspectral image projector for advanced sensor characterization,” *Proc. SPIE* **6296**, 629602 (2006).

[5] Rice, J. P., Brown, S. W., Johnson, B. C., and Neira, J. E., “Hyperspectral image projectors for radiometric applications,” *Metrologia* **43**, S61-S65 (2006).

[6] Brown, S. W., Rice, J. P., Neira, J. E., Johnson, B. C., and Jackson, J. D., “Spectrally tunable sources for advanced radiometric applications,” *J. Res. Natl. Inst. Stand. Technol.* **111**, 401-410 (2006).

[7] Brown, S. W., Rice, J. P., Allen, D. W., Zuzak, K., Livingston, E., and Litorja, M., “Dynamically programmable digital tissue phantoms,” *Proc. SPIE* **6870**, 687003 (2008).

[8] Rice, J. P., Neira, J. E., Kehoe, M., and Swanson, R., “DMD diffraction measurements to support design of projectors for test and evaluation of multispectral and hyperspectral imaging sensors,” *Proc. SPIE* **7210**, 72100D (2009).

[9] Allen, D. W., Rice, J. P., and Goodman, J. A., “Hyperspectral projection of a coral reef scene using the NIST hyperspectral image projector,” *Proc. SPIE* **7334** (these proceedings) (2009).

[10] Gruninger, J., Ratkowski, A. J., and Hoke, M. L., “The sequential maximum angle convex cone (SMACC) endmember model,” *Proc. SPIE* **5425**, 1-14 (2004).