Segmentation of hyperspectral images based on histograms of principal components

J. Silverman[a], S.R. Rotman[a,b] and C.E. Caefer[a]
[a]Air Force Research Laboratory, Hanscom AFB; [b]Ben-Gurion Univ. of the Negev

ABSTRACT

Further refinements are presented on a simple and fast way to cluster/segment hyperspectral imagery. In earlier work, it was shown that, starting with the first 2 principal component images, one could form a 2-dimensional histogram and cluster all pixels on the basis of the proximity to the peaks. Issues that we analyzed this year are the proper weighting of the different principal components as a function of the peak shape and automatic methods based on an entropy measure to control the number of clusters and the segmentation of the data to produce the most meaningful results. Examples from both visible and infrared hyperspectral data will be shown.

Keywords: Hyperspectral, segmentation, histograms, principal components, entropy

1. INTRODUCTION

Hyperspectral imagery can be overwhelming given the large quantity of data obtained in both the spectral and spatial domains. One way to overcome this deluge of information is to convert the data to its principal components; in this way, an entire datacube may be summarized in a few images.

Often, one wishes to go one step further; one desires to reduce the entire data cube to a single segmented image in which one of a small number of digital grey levels is assigned to each pixel. Last year[1], we showed that we could do this by taking two of our principal components images and process them as follows:

1. Produce a two-dimensional histogram where the values of each axis are determined by one of the principal components images.

2. Locate the N peaks in the histogram and assign a digital value from 1 to N.

3. Associate each pixel to one of the peaks based on the minimum Euclidian distance and use the corresponding peak digital value as the segmentation level for that pixel.

In addition to the obvious extension to N-components, several questions arise when refining this algorithm.

1. First of all, how should we define a peak in the histograms when implementing the peak determination stage? In our initial algorithm, we insisted that a peak be a maximum of at least height 2 within its own three pixel by three pixel neighborhood . How will other definitions of a peak affect the results?

2. How should we model the peak? If we assume that peaks are Gaussian, then the relevant parameters needed are the mean and variances of the peaks in N dimensions. The means of the peaks can be assumed to be the location of the maximum values of the peak. However, how can we obtain the width of a peak?

3. When forming our histograms from the floating point values of the principal components, how should the data be partitioned into bins? In our earlier work[1], we show that the number of final segmentation levels increases (up to a point) with the number of integer bins. Here we explore

Imaging Spectrometry VIII, Sylvia S. Shen, Editor, Proceedings of SPIE Vol. 4816
(2002) © 2002 SPIE · 0277-786X/2002/$15.00

non-linear mappings from the principal component values to integers under the guidance of an entropy criterion.

This paper will address these issues and refinements.

2. CLUSTERING MECHANISM

The best way to explain the clustering/segmentation technique is by an example. Let us start with a datacube taken from our AFRL/Solid State Scientific Corporation chromotomographic visible/near infrared hyperspectral imaging sensor (CTHIS).[2] Fig. 1a is a single band image of the 74-band data cube. The first two principal components are shown in Figs. 1b and 1c. These floating point continuous values must be binned into some range of integers by a suitable scaling technique (see Section IV). If we plot the resulting integer values into a 2-dimensional histogram, we get the image shown in Fig. 1d. If we then pick out the peaks in these images (where a peak is defined as having at least two pixels and is the maximum in a 3 by 3 area), we obtain the image shown in Fig. 1e. We can weed out redundant maxima, not allowing any to be within two pixels or less of each other; this is the result shown in Fig. 1f. A template is then built in which every possible combination of pixel values for the first and second principal component images are assigned to one of the peaks (Fig. 1g). The segmented image to 14 levels on the basis of this assignment is shown in Fig. 1h.

Fig. 1a. Sample band from visible hyperspectral data cube

Fig. 1b. First principal component of visible data cube

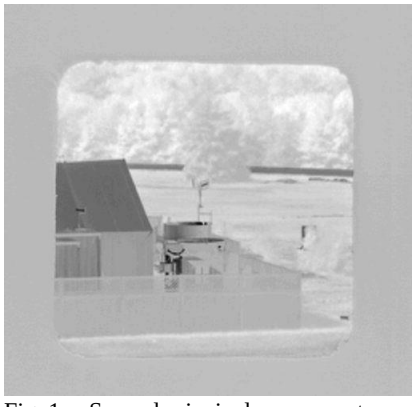

Fig. 1c. Second principal component.

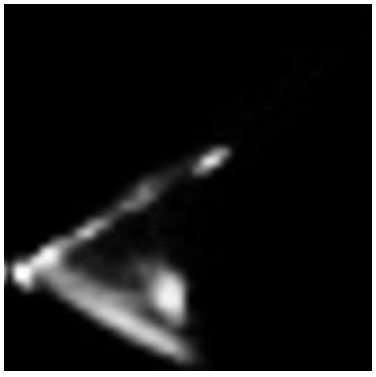

Fig. 1d. Two dimensional histogram.

Fig. 1e. Peaks of two-dimensional histogram

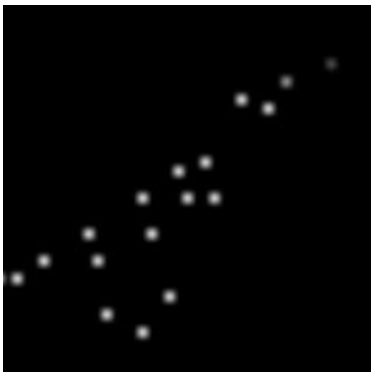

Fig. 1f . Weeded peaks of two-dimensional histogram

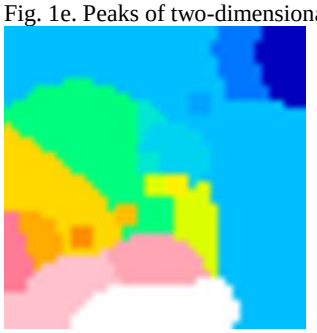

Fig. 1g. Template

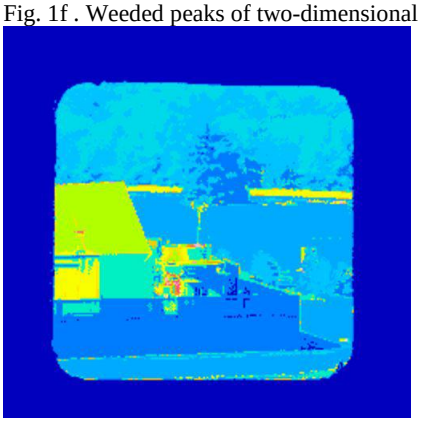

Fig. 1h. 14-level segmented image.

3. TWO-DIMENSIONAL CO-HISTOGRAM

We use the two-dimensional co-histogram to model our peaks; the co-histogram is one type of cooccurence matrix.[3] The co-histogram plots the association of grey-level values of a central pixel with the grey-level values of its 3 by 3 neighbors. Take the abscissa and ordinate as the central and neighboring grey level values. A value of 20 (for example) at coordinate (40,45) means that there are 20 occurrences in the image of a pixel of grey level 40 having a pixel of level 45 as a neighbor. Regions in the image with different noise characteristics form unique regions in the co-histogram. Note that the point (45,40) will also be 20 leading to mirror symmetry along the diagonal.

Fig. 2a and 2b show the resulting two-dimensional co-histograms, respectively, of the first two principal component images from the data cube described in Fig. 1. The peaks in general occur along the diagonal. This is reasonable, since pixels of like values will tend to group together. If we think of each line as representing the statistics of a single grey level, we can see that certain levels have characteristic large variations in their neighborhood, while others have smaller variations. The statistics of each line, i.e. the variance of the values along the line, can determine the natural width of the changes that we can expect in the neighborhood of these pixels.

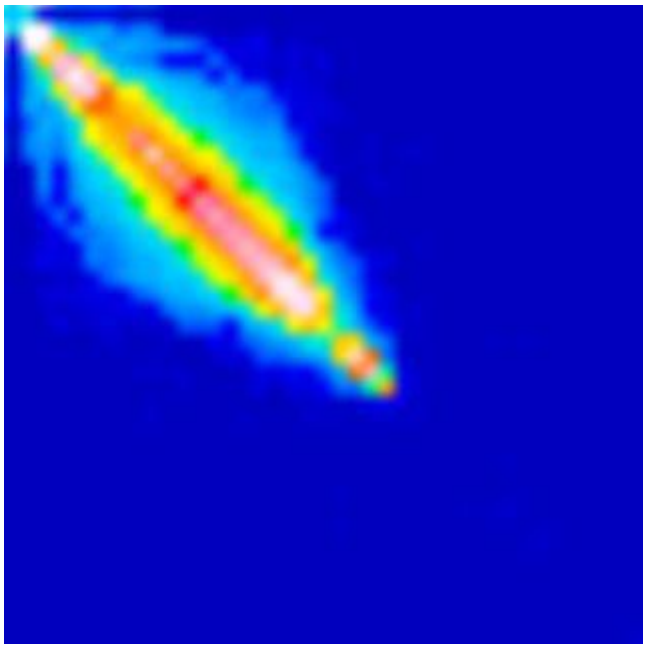

Fig. 2a. Co-histogram of the first principal component

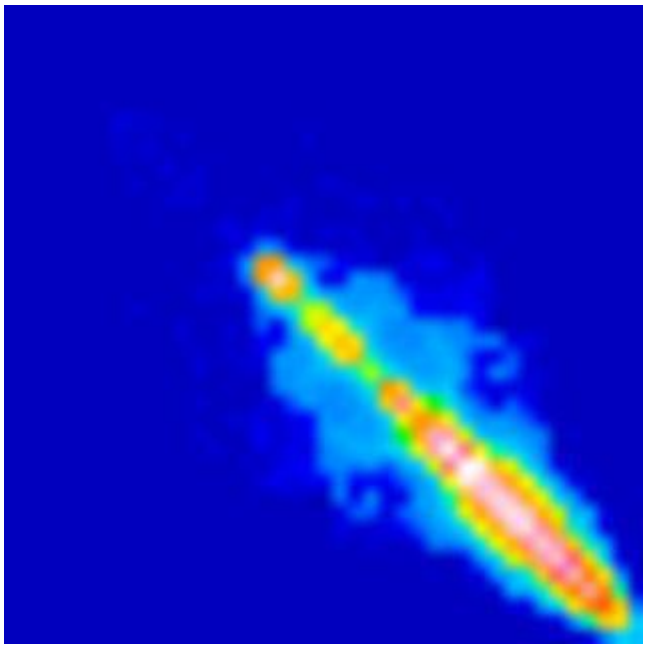

Fig. 2b. Co-histogram of the second principal component.

With this we can now address the second issue listed above. Let us start with the one- dimensional problem, i.e. segmenting on the basis of a single component image. If we have two Guassian distributions stemming from two peaks in the histogram to which a single pixel could belong, then the maximum likelihood estimate for the correct class would be based on

$$Ae^{-(x-m_A)^2/2\sigma_A^2} \underset{<_B}{>^A} Be^{-(x-m_B)^2/2\sigma_B^2} \qquad (1)$$

where A and B are the peak values of the two distributions, m_A and m_B are the means of the distributions and σ_A and σ_B are the variances. A more convenient mathematical formulation is to assign pixel A to the peak with minimum value for the experession:

$$(x-m_A)^2 - 2\sigma_A^2 \ln(A) \qquad (2)$$

The variable A for each distribution is assumed to be proportional to the number of points at the center of the peak on the N-dimensional histogram described in the first section. The mean m_A is the value of the peak location. The variance σ_A can be calculated as the variance of the distribution given for the central pixel with grey level value m_A on the co-histogram. A wider distribution, with larger values of σ_A and A, favors assigning pixels with distant values. The narrower and smaller the peak, the more we will restrain the distribution and demand that only pixels with closer/similar values should be allowed to join.

This approach increases in importance for 2D-histograms or greater. Assume that N equals two, i.e. we are trying to segment the image on the basis of two component images. If one image is very noisy and the other is not, we want to segment on the basis of the non-noisy image. This will occur automatically when we consider our desired minimized Equation 2 generalized to two dimensions:

$$\frac{(x_1-m_{A1})^2}{\sigma_{A1}^2} + \frac{(x_2-m_{A2})}{\sigma_{A2}^2} - 2\ln A \qquad (3)$$

Here the values of σ will be found from the co-histograms of each of the components separately. If the second component image is much noisier than the first image, then the value of $\sigma_A(2)$ will be much greater than $\sigma_A(1)$. The second component will not be as significant as the first component when the assignments are considered.

To demonstrate this dramatically, we show a case of a clean first component image with a very noisy second component image. Figs. 1b and 3a show the first and seventh principal component images, respectively; Figs. 2a and 3b show the resulting co-histograms from these components. As can be seen, the distributions in Fig. 3b are considerably wider than the distributions in Fig. 2a; hence the expression in Equation 3 will be much more dependent on variations in the first component image than the second component image. Fig. 3c is the resulting image using minimum Euclidian distance without considering the factors in Equation 3; Fig. 3d is the image when incorporating Equation 3. The resulting improvement is quite evident.

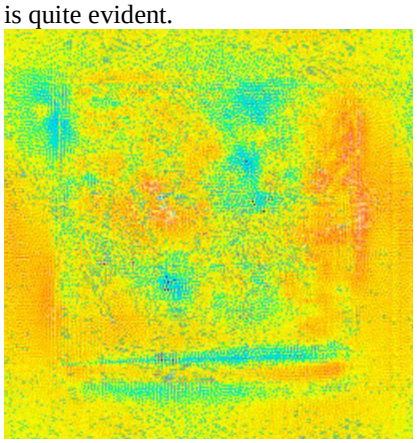

Fig. 3a. Seventh principal component

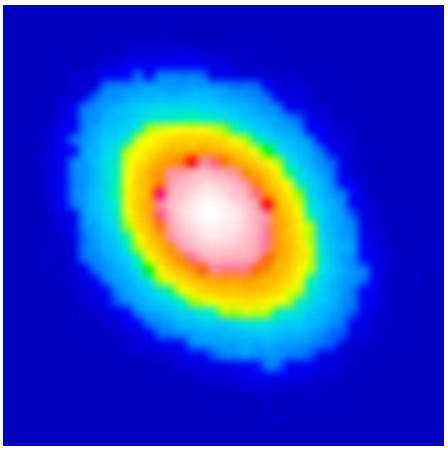

Fig. 3b. Co-histogram of seventh principal component

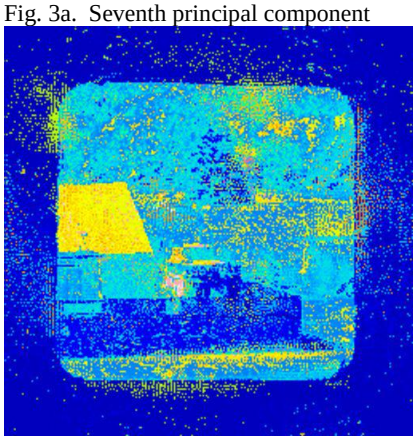

Fig. 3c. Segmentation without taking into account the quality of the principal components.

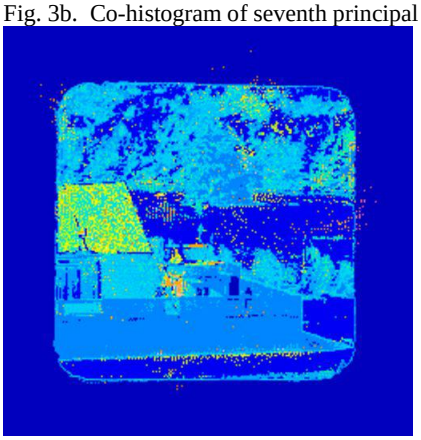

Fig. 3d. Segmentation using cohistograms to measure the quality of the principal components.

4. ENTROPY-GUIDED MAPPINGS

We next address the third issue raised in our introduction: namely, how to map the floating point principal component data into the integer bins needed to generate the histograms. This issue becomes critical when the desired number of segmentation levels is small, say < 15, and the image consists mostly of slowly spatially varying intensity backgrounds but with occasional significant rapidly varying intensity regions at levels removed from the background. The IR hyperspectral image, whose first two principal components are shown in Fig. 4a and b, where the building is such a high rapidly varying region, will be our example for exploring this issue.

In our initial formulation of the histogram segmentation technique[1], we used linear scaling of the principal component floating point numbers into a specified range of integer bins and iterated this range until the user-desired number of segmentation levels was achieved. The only other user specified parameter was the setting of the minimum number of pixels at a maximum location accepted as a peak (taken as two in our earlier work). The effect of changing this peak definition is illustrated in Fig. 4c and d where segmentation to 38 levels at peak min 2 is compared to peak min 5 with 22 levels; the change in peak min has removed 16 levels. Nevertheless, the overall segmentation results look similar. The consequence is to decrease the number of segmentation levels assigned to the building while maintaining the same discrimination in the background portions. However choosing the peak definition that gives the most pleasing segmentation is both subjective and empirical and we prefer the alternative scaling technique described next.

In past research on the display of infrared imagery[5], which treats mapping from high dynamic range histograms (the "raw" histogram) to 8-bits (the display histogram), we introduced a non- linear mapping, Histogram Projection (HP) which treats each occupied level of the raw histogram equally. This was in contrast to the standard technique of Histogram Equalization (HE) that allots integer range in the display histogram in proportion to histogram height in the raw histogram. The HE mapping tends to lump high frequency spatial regions at sparsely occupied levels into a few display levels. We further introduced a more generalized mapping, Plateau Equalization (PE), in which, by introducing a plateau or saturation level into the raw histogram computation, we generate intermediate mappings to those of HP (plateau = 1) and those of HE (plateau above the maximum in the raw histogram).

We have adopted the plateau mapping into the present scaling problem as follows. First, we linearly scale the floating point numbers into a large integer range (typically 0 to 1000) that retains the basic statistics of the initial floating point numbers. Then we employ PE to map from the "raw" histogram of 1000 to the much lower integer ranges needed to form the histograms used in the segmentation techniques (ranges from 10 to 70 integer bins are typical for the 2D histograms.) To automate the process of choosing the plateau level that gives the "best" result at a user-specified number of final segmentation levels, we employ an entropy measure. Our experience is that the first component of a Principal Component analysis generally provides a high contrast view of the scene with reasonable balance between high and low spatial frequency regions. Assuming we desire segmentation to N levels, we linearly scale the first component to N levels in order to compute the entropy E of this component by the expression:

$$E = \sum_{0}^{N-1} -\left(f_i \ln\left(f_i\right)\right) \tag{4}$$

where f_i is the fraction of pixels at level i. We next run our algorithm over a range of plateau levels, typically 1, 5, 10, 15, 20, 25 and 30. At each plateau level, we generate the histograms produced by the corresponding PE mapping and iteratively change the final number of integer bins until we obtain our N level (or just below) segmentation. The entropy of the final segmentation is computed as above and the preferred final result is that of the plateau level whose final segmentation entropy is closest to the starting entropy of the first Principal Component.

In Fig. 4e and f, we show the results of this technique for the plateau = 1 result (close to linear scaling if the principal component floating point numbers form a smooth continuum) as compared to the plateau level result with the best entropy match. In Fig 4f with the plateau level equal to 30, the entropy guides us to an image in which the fundamental segmentation of the image into field stop, sky, building, trees, and ground is clear and similar to that in the first principal component image. In Fig. 4e with the plateau level equal to 1, the sky, trees and part of the ground are all combined into a single segment.

Fig. 4a. First principal component of infrared image.

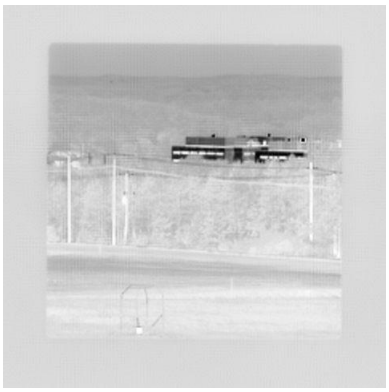

Fig. 4b. Second principal component of infrared image

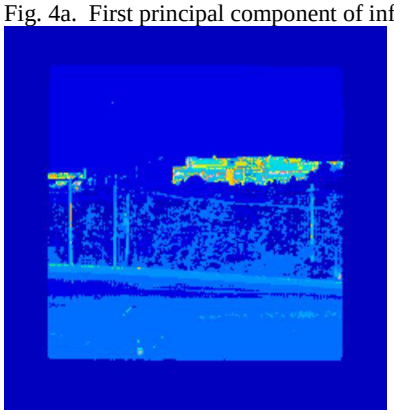

Fig. 4c. 38 level segmentation with a peak minimum value of 2.

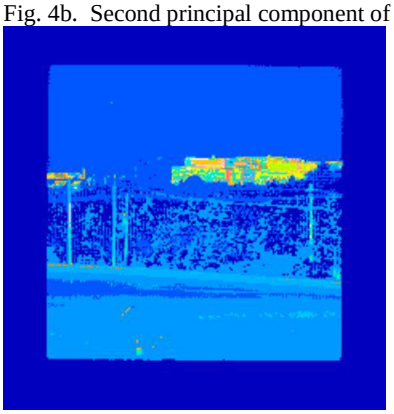

Fig. 4d. 22 level segmentation with a peak minimum value of 5.

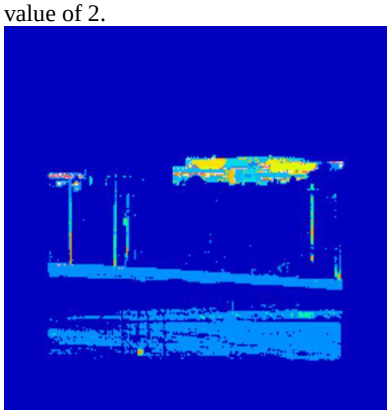

Fig. 4e. Segmentation at 12 levels inputting the data linearly and without using the entropy metric.

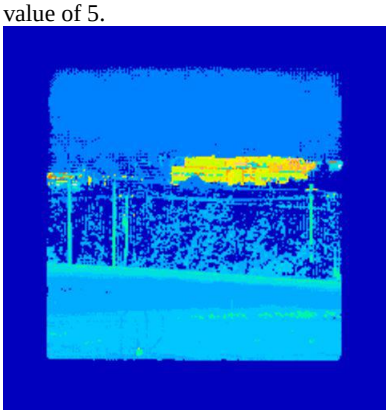

Fig. 4f. Segmentation at 12 levels with non-linear inputting of the data by using the entropy metric.

5. CONCLUSIONS

In this paper, we have described refinements to a method of segmenting hyperspectral images on the basis of the most significant principal components. Two major refinements have been incorporated. Moving away from a minimum distance assignment to peaks in our histograms, we statistically characterize each peak by means of the co-histogram of the integer-binned principal components. Secondly, using the technique of PE, we map the floating point numbers from their initial large integer mapping into the range required for the histograms. The plateau level of choice is automatically chosen through the best entropy-measure match between the final segmentation and the first component.

Several different applications can be considered, given this capability. Standard automatic target recognition algorithms use segmented images to identify targets. Also, the larger clusters can be used to generate endmembers that represent the backgrounds; this is often a first step toward detecting pixels with

spectral anomalies that deviate from the standard background signatures. For further details on follow-up applications, see Ref. 5.

ACKNOWLEDGMENTS

The cameras used to collect the visible and MWIR data were designed and fabricated by William Ewing, Toby Reeves and Steven DiSalvo of our laboratory. We are grateful to our Linda Bouthillette for graphic assistance. This work was carried out under Air Force Task 2305BN00. We'd like to acknowledge partial support of the Paul Ivanier Center for Robotics and Industrial Production, Beer-Sheva, Israel. This work was performed while one of the authors (SRR) held a National Research Council Research Associateship Award at the Air Force Research Laboratory at Hanscom AFB.

REFERENCES

1. J. Silverman, C. E. Caefer, J.M. Mooney, M.M. Weeks, and P. Yip, "An automated clustering/segmentation of hyperspectral images based on histogram thresholding", in *Imaging Spectrometry VII*, Michael R. Descour and Sylvia S. Shen, Editors, Proceedings of SPIE Vol. 4480, 65-75 (2002).

2. J.E. Murguia, T.D. Reeves, J.M. Mooney, W.S. Ewing, F.D. Shepherd, and A.K. Brodzik, "A compact visible/near infrared hyperspectral imager", in *Infrared Detectors and Focal Plane Arrays*, E.L. Dereniak and R.E. Sampson, Editors, Proc. of SPIE Vol. 4028, 457-468, 2000.

3. R. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification", IEEE Trans. Syst. Man Cyber. **3,** 610-621 (1973).

4. V.E. Vickers, "Plateau equalization algorithm for real-time display of high-quality infrared imagery", *Opt. Eng.* **35(7),** pp. 1921-1926 (1996).

5. J. Silverman and S.R. Rotman, "Segmentations of hyperspectral imagery: techniques and applications", Proc. SPIE 4820, (to be published).