Estimating True Color Imagery for GOES-R

Michael D. Grossberg[a], Fazlul Shahriar[b], Irina Gladkova[a], Paul K. Alabi[a], Donald W. Hillger[c], and Steven D. Miller[d]

[a]City College of New York, 160 Convent Avenue, New York, NY 10031;
[b]CUNY Graduate Center, 365 Fifth Avenue, New York, NY 10016;
[c]NOAA/NESDIS, Regional and Mesoscale Meteorology Branch, in Fort Collins, CO;
[d]Cooperative Institute for Research in the Atmosphere, Fort Collins, CO;

ABSTRACT

The Advanced Baseline Imager (ABI) on GOES-R will help NOAA's objective of engaging and educating the public on environmental issues by providing near real-time imagery of the earth-atmosphere system. True color satellite images are beneficial to the public, as well as to scientists, who use these images as an important "decision aid" and visualization tool. Unfortunately, ABI only has two visible bands (cyan and red) and does not directly produce the three bands (blue, green, and red) used to create true color imagery.

We have developed an algorithm that will produce quantitative true color imagery from ABI. Our algorithm estimates the three tristimulus values of the international standard CIE 1931 XYZ colorspace for each pixel of the ABI image, and thus is compatible with a wide range of software packages and hardware devices. Our algorithm is based on a non-linear statistical regression framework that incorporate both classification and local multi-spectral regression using training data. We have used training data from the hyper-spectral imager Hyperion. Our algorithm to produce true color images from the ABI is not specific to ABI and may be applicable to other satellites which, like the ABI, do not have the ability to *directly* produce RGB imagery.

Keywords: GOES-R, ABI, CIE-XYZ, Hyperion, green, color, regression, true color

1. INTRODUCTION

Advanced Baseline Imager (ABI) on the future GOES-R satellite will not have a channel that senses in the green range. It has only two channels, 470nm and 640nm in the visible range.[1] As a result, ABI cannot directly produce color images despite the widespread demand for such images for use as "decision aids" by meteorologists and for visualization by the public. NOAA's goal of integrating earth observations requires the ability to provide ABI data in a form that can be compared with other earth observations such as conventional color images. We address this problem by fusing ABI data in multiple visible and near visible spectral bands to best quantitatively estimate a true color image product.

Sources that may be used to estimate this true color product include MODIS data from the Terra and Aqua satellites and data from other imagers on polar orbiters. In addition to these sources, accurate wide band true color regression models can be constructed using offline data sources that provide high spectral resolution reflectance. In this paper we use empirical data from the hyper-spectral imager Hyperion, as offline training and verification data sets.

Prior work focused on estimating a spectrally narrow green band response to produce a color image. This is a crude approximation to a perceptually accurate RGB image, so instead we have statistically reconstructed an X, Y, and Z channel based on the standardized CIE 1931 XYZ color space defined by CIE (International Commission on Illumination).[2] Thus, our algorithm directly minimizes the true color error, not only for improving visualization applications but also potentially allowing for accurate fusion of ABI data with RGB images taken on the ground.

Further author information: (Send correspondence to M.G.)
M.G.: E-mail: grossberg@cs.ccny.cuny.edu, Telephone: 212-650-6166

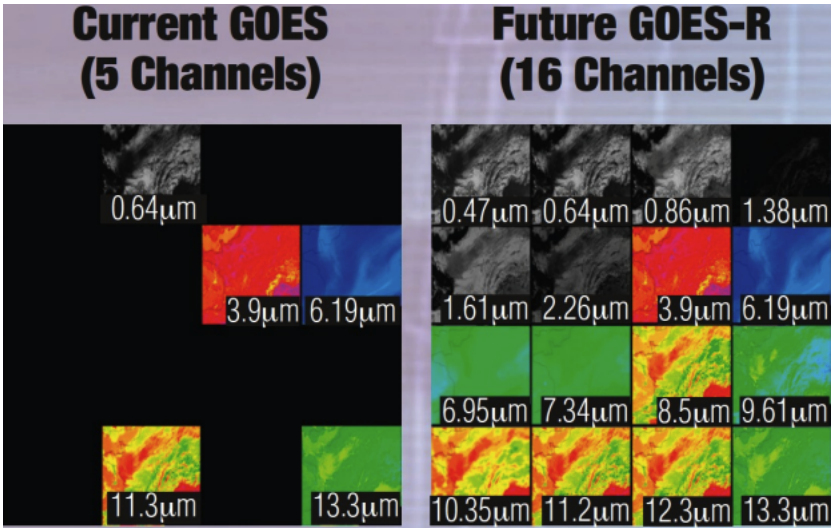


Figure 1: Current GOES Imager vs ABI GOES-R bands. Image courtesy of the Cooperative Institute for Meteorological Satellite Studies (CIMMS)

Because an accurate RGB image is based on broad, rather than narrow, spectral bands, simply choosing bands "close to" R, G and B is insufficient and more spectral parameters are needed. Because the number of entries in look-up tables grows exponentially with the number of parameters, the straightforward use of traditional look-up tables becomes impractical. Simple approaches will not produce satisfactory results in all situations due to the underlying non-linearity of the data, and the relationship among the spectral reflectances of different materials. For instance cloud footprint spectra will be radically different from that of desert or forest. Non-parametric methods such as look-up tables, nearest neighbor regression, kernel methods, and cluster analysis can be used to capture these complex relationships. Unfortunately non-parametric methods can require more data than is practical. In this paper we present one approach which combines a $K-$means clustering with a multi-linear regression for each cluster. This provides a compromise between the flexibility of non-parametric methods while still limiting the need for training data. We will present an evaluation showing that this compromise provides good performance on estimation of the XYZ true color.

2. BACKGROUND

The issue of generating a synthetic green band has been investigated by both CIMSS and CIRA using a look-up table (LUT) method.[3,4] The approach they took is to derive this LUT statistically. The LUT is designed to take as input, values for the 470 nm cyan band, 640nm red band, and the 860 nm NIR band and to output a 550nm green band value. The LUT is built from a large set of modeled and measured data. From this data many pixels with the coincident reflectances of the four bands, cyan, red, NIR, and green in images are collected. Each entry in a 250×250×250 data cube array represents a bin in the cyan, red, and NIR reflectance space. The value assigned to each bin is the average green band value for that bin as determined across the whole data collection. The lookup table is applied to new data by finding the nearest cyan, red and NIR reflectance from the ABI, and looking up the corresponding green values. The values from the estimated green, cyan and red bands are then Rayleigh corrected and stretched through a log enhancement. They are then combined to form an RGB true color value for visualization. The result shown in Figure 2 is an example from D. Hillger at CIRA using simulated ABI data generated at CIMSS.

While the LUT method above does provide a good statistical regression of the 550 nm green channel, it does not take into account that human perception is not based on a narrow band red, green and blue channel responses. The human visual system does not respond to only narrow red, green and blue bands, but has three broadband color responses; these responses have been standardized and quantified through the CIE 1931 XYZ color space.[2] The XYZ color space provides three functions which when applied to the measured spectral radiance curve, give

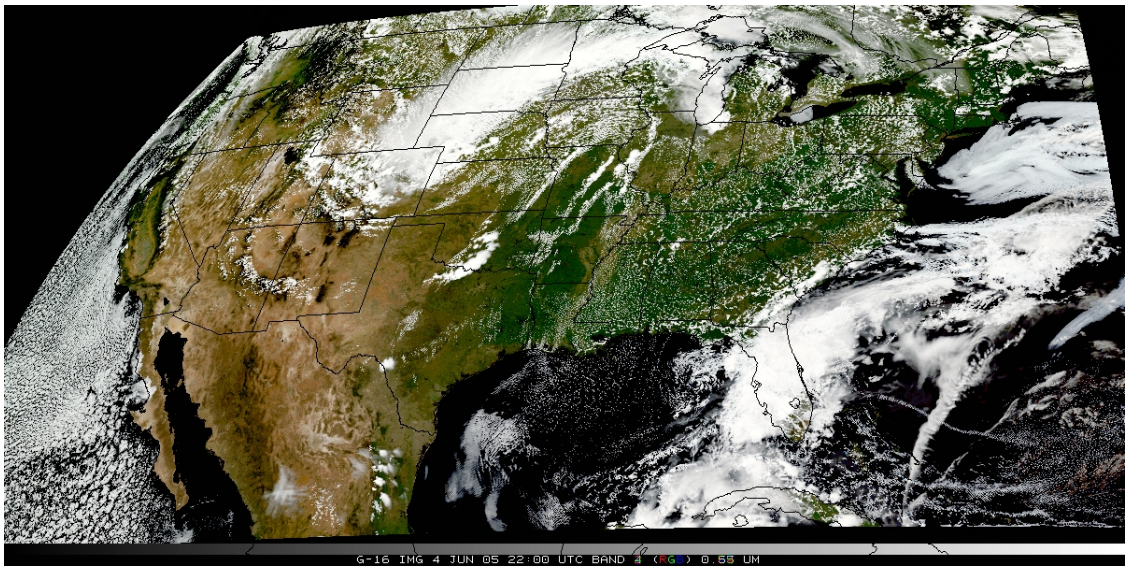


Figure 2: True color image created at CIRA using simulated ABI data from CIMSS.

parameters X, Y, and Z called the *tristimulus values.* Loosely speaking, X, Y, and Z are closely related to red, green and blue, respectively. However, unlike a loose definition of red, green and blue, XYZ are precisely defined quantities.

The tristimulus values are derived from broad spectral band responses applied to the spectral radiance distribution for each point. This spectral radiance distribution could be based on a model or come from measured data. If the spectral radiance distribution $I(\lambda)$ is given in $watts/meter^2$ per unit wavelength $\Delta\lambda$, then the tristimulus values $X(I)$, $Y(I)$, $Z(I)$ are linear functionals of I. They were chosen based on experimental measurements with human subjects to establish an international standard (color) observer. If any normal observer is asked to distinguish two patches of color, having spectral radiance I_1 and I_2 respectively, they will be impossible to distinguish if and only if $X(I_1) = X(I_2)$, $Y(I_1) = Y(I_2)$, and $Z(I_1) = Z(I_2)$. The tristimulus values are defined by three color matching functions (densities) x, y, z so that

$$X = \int_0^\infty I(\lambda)x(\lambda)d\lambda, \qquad Y = \int_0^\infty I(\lambda)y(\lambda)d\lambda, \qquad Z = \int_0^\infty I(\lambda)z(\lambda)d\lambda \tag{1}$$

where the curves are shown in Figure 3. The XYZ colorspace is an international standard for perceived color that is incorporated into most professional design tools. Calibration tools are also widespread for measuring XYZ, as are tools that can convert XYZ to the RGB of a specific computer monitor, or the CMYK (Cyan, Magenta, Yellow and Black) of a specific printer. Thus the determination of the XYZ color for an ABI granule, unlike the estimation of a narrow green band, makes it possible to able to produce an accurate color representation for any viewing device. The goal of this paper is to quantitatively estimate the XYZ values of a scene given ABI radiances.

3. ALGORITHM

To address the problem of estimating a true color image product from the ABI, our approach is to statistically train from a pool of representative examples. We use these examples to develop a training set consisting of many pixels where we have coincident values for X, Y, and Z as well as values for the 6 visible and near visible bands of ABI. We then apply a non-linear regression method which produces a predictor of the values of X, Y, and Z from the ABI data. We then can use this predictor to produce X, Y, and Z, given new ABI data. The outline of the approach is presented in the diagram shown in Figure 5. The training data is input into a non-linear multi-parameter regression algorithm to estimate a set of predictor parameters. The predictor takes ABI visible

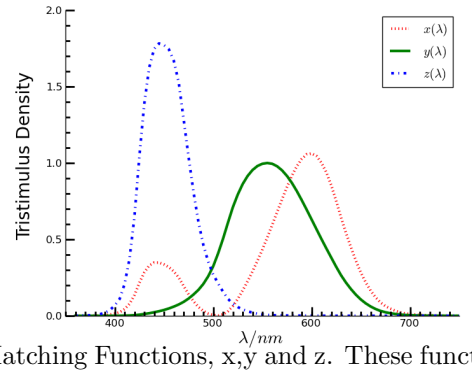


Figure 3: CIE 1931-XYZ Color Matching Functions, x,y and z. These functions when multiplied by the observed spectral radiance distribution, and integrated, yield the broad-band X,Y, and Z tristimulus values.

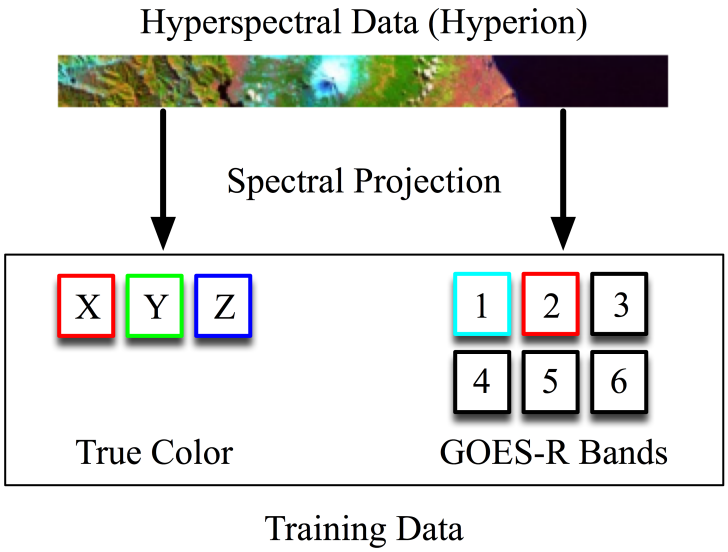


Figure 4: Diagram showing the construction of training data. Both the tristimulus values, represented by the boxes labled X,Y, and Z, and the visible and near visible ABI simulated bands (1-6) are produced from a spectral radiance distribution at each pixel. The spectral radiance distributions were derived from from Hyperion data.

and near-IR images as input, and produces true color images as output. After training, the predictor is applied to independent testing data to evaluate the accuracy of the predicted values. Hence the output of the final algorithm will be a true color predictor function, which produces estimates of true color images for the ABI.

The first component is obtaining and building ground truth training and testing data sets. One way to do this is to start with spectral radiance distributions $I(\lambda)$ for a set of representative scene points. These could be obtained from physical modeling and simulation, or from measurements. In this paper we derive our spectral radiance distributions using data from a hyper-spectral imager, the Hyperion on the Earth Observing Mission 1 satellite (EO-1). Suppose that at each footprint p associated with a pixel in the imager, the spectral distribution is I_p. We can then obtain the values of X_p, Y_p, and Z_p by integrating the spectral distribution against the matching functions, using equation 1. In a similar way we can obtain the co-incident values for the 6 visible, and near visible bands of the ABI, $B_{p,1}, \ldots B_{p,6}$, from the ABI specified spectral quantum efficiency curves $b_1, \ldots, b_6$ in place of the matching functions x, y and z in equation 1. Together, the spectral projections of the XYZ values and the 6 ABI bands applied to the hyper-spectral data at each pixel, as illustrated in Figure 4, result in a large training set.

The second component of the algorithm, illustrated in Figure 5, is the development of an effective parameterized predictor, along with an efficient method of estimating the predictor parameters. One simple approach is the

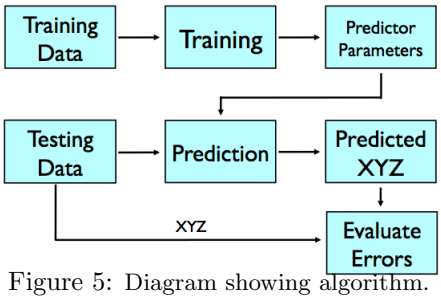


Figure 5: Diagram showing algorithm.

construction of a look-up table (LUT). For each quantized set of inputs in the input bands, $B_1, \ldots, B_6$ we would need a large number of examples in the training data data. The average values of X, Y and Z, respectively, for that input are the expected values for XYZ and would be used as the estimates. This is essentially the method used by Miller et al.[4] The LUT method has the advantage of being non-linear and purely data driven. One problem is that unlike in Miller et al[4] which used 3 input bands for estimation, we are using 6 input bands. The lookup table grows exponentially in the number of input bands, as does the need for training data. In the case of 6 bands, even in the best (and unrealistic) case where input all the pixel values were uniformly distributed this would require the equivalent of hundreds of millions of granules for training. Without this much data, a lookup table risks over-fitting. In other words, there are not enough examples to smooth the natural (uninformative) variation in the data. As a result the lookup table would just summarize past data and generalize poorly on future data.

Another very simple method of estimating a predictor, is to estimate the values as a multi-linear function of the inputs. That is if $w = [X, Y, Z]$ is thought of as a 3×1 column vector and If $v = [B_1, \ldots, B_6]$ is a 6×1 column vector of ABI inputs then a XYZ multi-linear predictor is a $3x6$ matrix A (6 dim input 3 dim output) so that

$$Av = w \tag{2}$$

with $w = [X, Y, Z]$ the predicted values of X, Y, and Z. There are several numerically stable methods of finding a least-squares solution to equation 2, minimizing the square error. The square error is given by

$$\text{Error}_S(A) = \sum_{p \in S} ||Av_p - w_p||^2 \tag{3}$$

over all the p in the training set S. Unlike a lookup table, this approach requires much less data, and is not subject to over fitting problems since we only need to estimate the 18 coefficients of A.

Unfortunately, the multi-linear model is too simplistic to provide satisfactory results. What is required is a compromise between the simple multi-linear model, and a LUT that allows us to add as much non-linear flexibility as we can, given the training data, while not overfitting. On further analysis, the multi-linear model does not fit the data well because for different materials and atmospheric constituents there are different relationships between the bands. Instead of trying to fit a single multi-linear relationship for all the data, if we can partition the input data space into related data clusters, we can provide a separate model for each cluster. Rather than arbitrarily partition the data we use a simple clustering method, $K-$means to break the data up into clusters.[5] The number of clusters is a free parameter K. For a fixed value, the K-means clustering algorithm is applied to the training data to break it up into K clusters $S_1, \ldots S_K$ which are represented by their K means, $m_1, \ldots, m_K$. For each cluster S_i we separately fit a multi-linear model A_i. For each new set of input values v the first step is to find the nearest mean m_i out of the means $m_1, \ldots m_K$. Then the predicted values for $w = [X, Y, Z]$ are given by $w = A_i v$.

Clustering breaks the data into separate data clusters. Since this means we fit a separate multi-linear model per cluster, which is a subset of the data, the fitting (training) error will at worst be the same, and typically will degrease. The fitting error will continue to decrease with an increasing number of clusters. Potential over-fitting can be evaluated by testing the model on test data produced in the same way as for the training data, but not used for fitting. This error of the fit on this test approximates the *generalization error*, ie, the error of the model

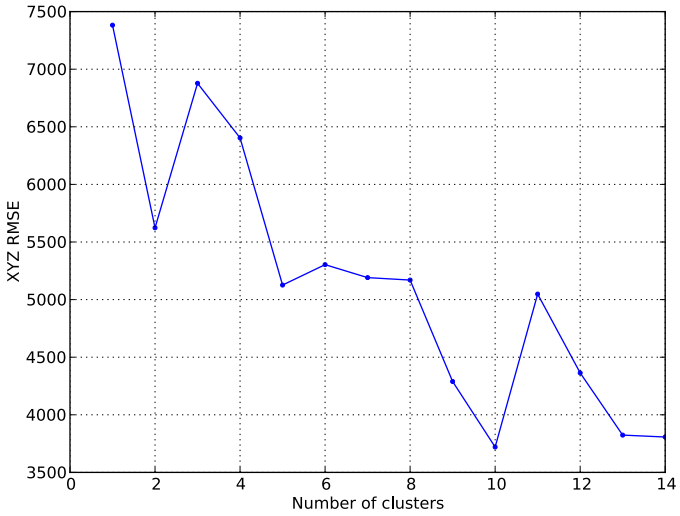


Figure 6: Root mean square error on testing data (not used to fit the model) plotted against an increasing number of clusters.

on new data. What we expect is that as we increase the number of clusters K the generalization error will, in general, drop for a while, and then as we begin over-fitting, the generalization error will not improve or even climb.

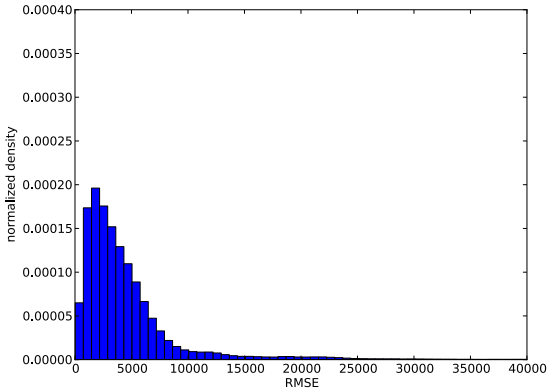
Figure 6 shows the root mean square error for XYZ color estimation on testing data using the method of K-means clustering, followed by K models. Because K-means is a random seeded iterative algorithm, the clustering followed by fitting was run on training data many (100) times, and the best result was used. This should be seen as part of the optimization. The Root Mean Square Error (RMSE) shown in the figure was taken from training data. Too little data was available for the result of Figure 6 to be considered statistically representative. The local increases and decreases in the RMSE have to do with the peculiarities of the data set and the interaction of the local models with the changes in the clustering. In general, as expected, there is a decrease in RMSE with increasing number of clusters. It can be seen, however, that after 10 clusters little or no improvement is seen. This suggests using more than 10 clusters would result in over-fitting.

3.1 Results

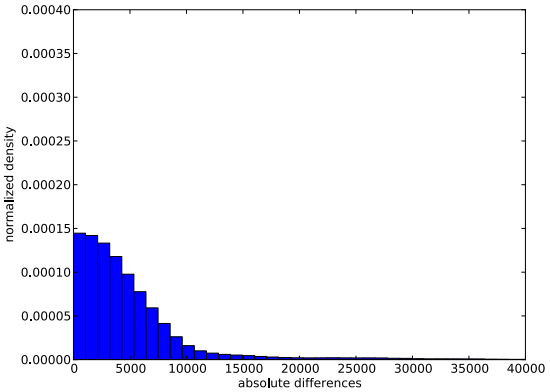
We implemented the algorithm using the Python programming language, the SciPy and Numpy libraries, as well as the Matplotlib library for visualization.[6–8] We used Hyperion sample granules from the EO-1 obtained from the USGS website. To evaluate the algorithm we used one large continuous region of the Hyperion granule for training and a separate, non-overlapping region for testing. In Figure 7(a) we measure the scaled histogram (probability) in root mean square errors (RMSE) in radiance between the true $v_p = [X_p, Y_p, Z_p]$, and the predicted $v'_p = [X'_p, Y'_p, Z'_p]$ at a pixel p, is given as

$$\text{XYZError}(p) = ||v'_p - v_p|| = \sqrt{\frac{1}{3} * \left((X_p - X'_p)^2 + (Y_p - Y'_p)^2 + (Z_p - Z'_p)^2.\right)} \tag{4}$$

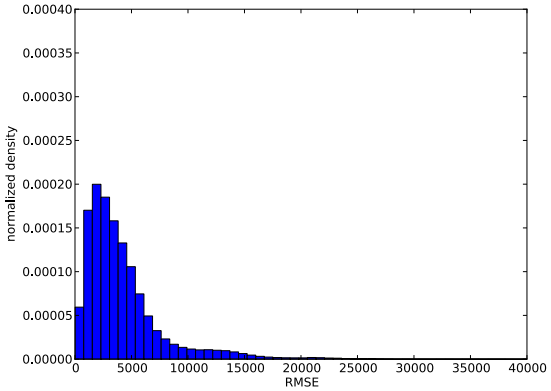
Note that the XYZ are broadband responses, thus that radiance levels, and consequently the errors, are much larger than radiance levels for a narrow band responses. The errors shown in Figure 7(a) are for a single multi-linear predictor (1 cluster.) If we just consider the Y response the absolute difference errors histogram are very similar, as shown in Figure 7(b). Both the XYZ error and the absolute errors on just the prediction of Y gradually improve when a second cluster is considered, as shown in Figures 7(c,d). We can see in Figure 7(e,f), that the distribution of errors improve much more when 10 clusters are used.



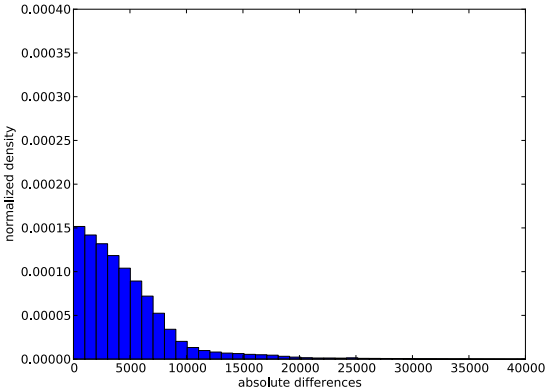
(a) XYZ (1 cluster)



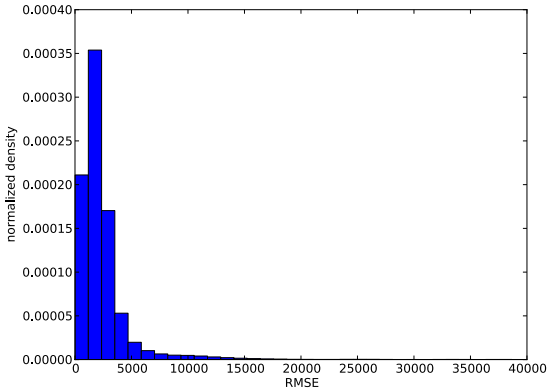
(b) Y (1 cluster)



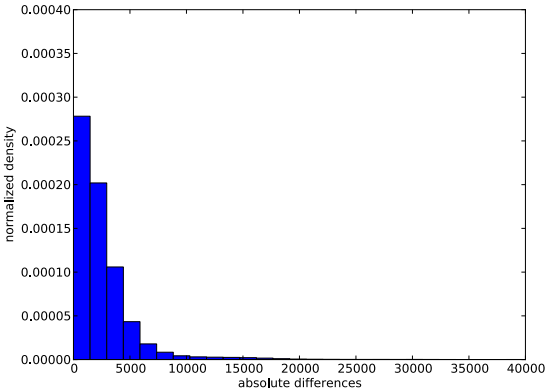
(c) XYZ (2 clusters)



(d) Y (2 clusters)



(e) XYZ (10 clusters)



(f) Y (10 clusters)

Figure 7: Histogram of RMSE

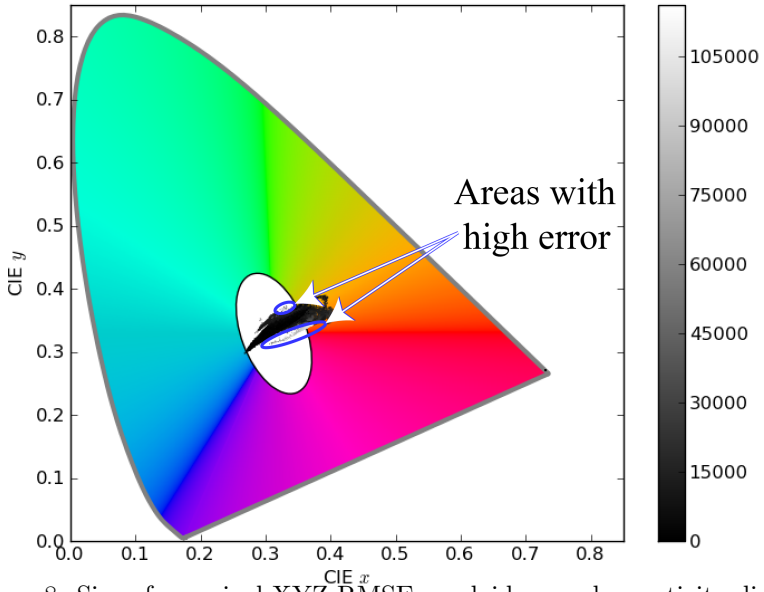


Figure 8: Size of per pixel XYZ-RMSE overlaid on a chromaticity diagram.

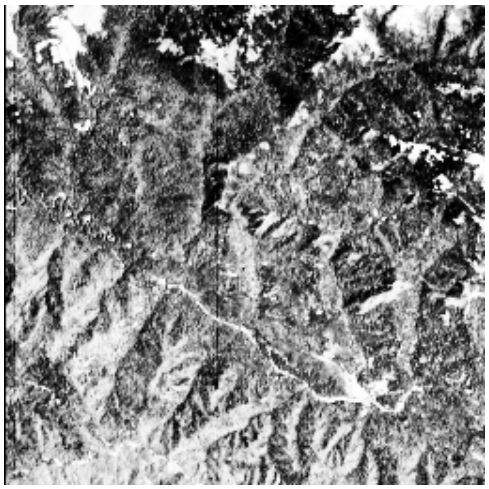
To get a better understanding of what colors are most problematic it is instructive to move the errors to the 2-dimensional chromaticity. This space normalizes by the luminance or more informally the "grey component". This is done by projecting the XYZ into two variables x, and y where

$$x = \frac{X}{X+Y+Z}, y = \frac{Y}{X+Y+Z}. \tag{5}$$

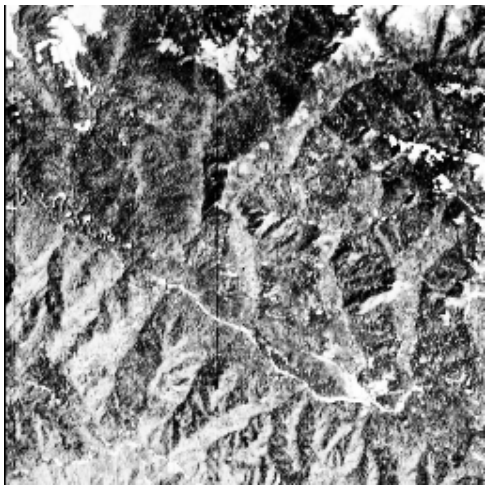
The plot of color pixels into this color space is called the *chromaticity diagram*, as shown in figure 8. The curve bounding the color region shows the limits of color perception. The bounding curve represents pure spectral curves where the center fades to a "white point" where the colors are desaturated. Overlaid on the chromaticity diagram are the x, and y values computed from the actual XYZ values of our data. The dots are colored by the RMSE of the predicted XYZ values. As we can see in figure 8, the large errors are concentrated in the desaturated red/purple region with some deep green pixels also being challenging. Still the majority of the color have small error.

In Figure 9 we see each component of X (top row), Y (middle row), and Z (top row) in the original image (left column) and predicted (right column). To produce the predicted images, 10 clusters were identified with K-means, one run was used rather than the 100 runs used to produce each clustering of Figure 6. The images were histogram-equalized so that the contrast between the predicted and original images would be more pronounced. The algorithm predicts the bands well although, in the middle row, we can see that the region within the circle marked A in the original image, shown in Figure 9(c) is noticeably lighter than in the predicted image Figure 9(d). Since the Y roughly corresponds to the green channel and the X and Z in this same region are nearly unchanged, once converted to RGB, this region is somewhat greener in the original image than in the predicted image. Nevertheless, in other regions it is nearly identical even for the Y value.

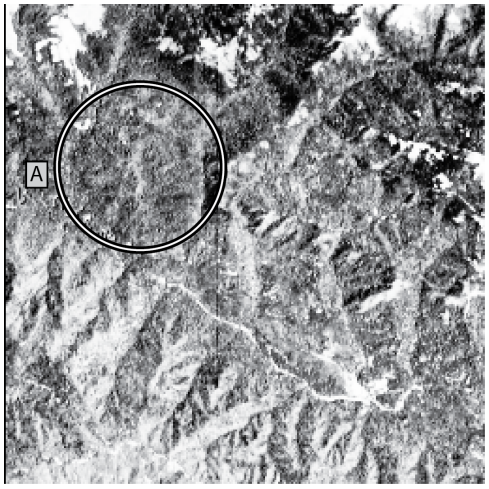
Since the quantity Y contains more sensitivity in the region where data is missing, that of green, than X and Z, the errors there are most relevant. To show how performance in Y is affected by increasing numbers of clusters, Figure 10 shows the improvement in the image with increasing numbers of clusters. In Figure 10(a) the original image is shown with two regions marked A and B for reference. Note that in Figure 10(b), prediction of Y with a single cluster the water region A is somewhat lighter than it is in Figure 10(a), while region B is somewhat darker. As we increase the number of clusters from 2 to 10, shown in the Figures 10(c,d), the intensities more nearly match regions A and B in the original image Figures 10(a).



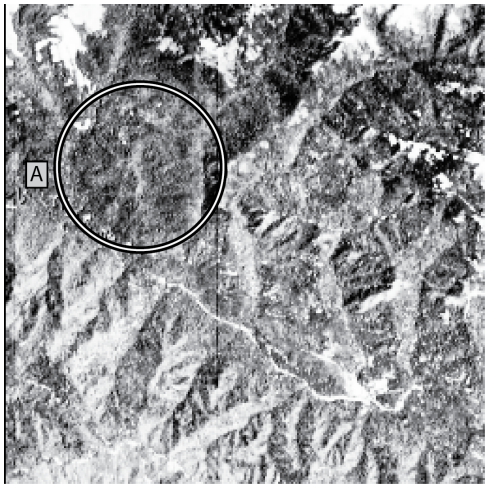
(a) Original X



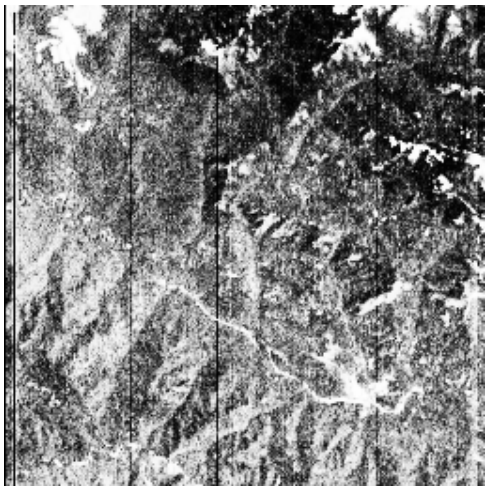
(b) Predicted X



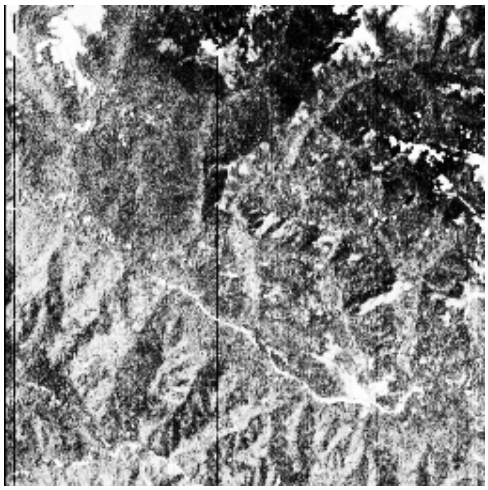
(c) Original Y



(d) Predicted Y



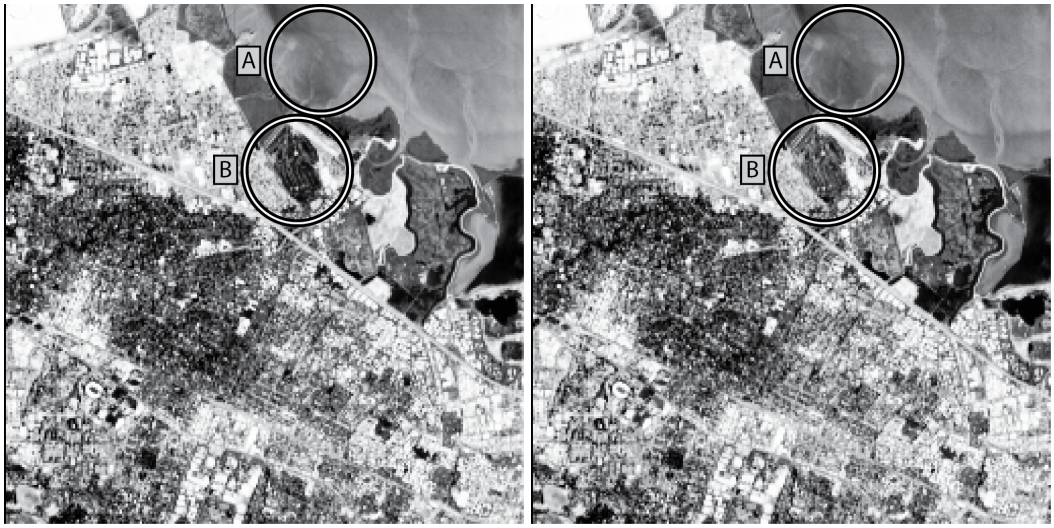
(e) Original Z



(f) Predicted Z

Figure 9: histogram-equalized XYZ images for 10 clusters.

(a) Original (b) 1 Cluster (predicted)



(c) 2 Clusters (predicted) (d) 10 Clusters (predicted)

Figure 10: Original Y shown with predicted Y for three different numbers of clusters. Region A and B are marked for easier comparison across the image.

To render a highly accurate color satellite image to a computer display or to a printer, device specific profiles must used to implement the transformation from XYZ to RGB. Since we intended to equalize the images anyhow, to emphasize the contrast between original and predicted, we converted XYZ to RGB using a generic profile. Figure 11 shows three original-predicted image pairs using this profile followed by histogram equalization of each band (independently) to maximize the visual differences. As can be seen in the pairs Figure 11(a,b) and Figure 11(c,d) the agreement for RGB is excellent although the irrelevant equalization process does introduce some unnatural colors (purple haze). The image pair in Figure 11(e,f) shows a failure case. While in most parts of the image the predicted image is quite close, the region which is brown in the middle right of the image in Figure 11(e) is yellow in the image in Figure 11(f). One problem is that because the K-means is performed independent of the multi-linear prediction, pixels requiring different predictors are sometimes grouped together.

4. CONCLUSION

In this work we have shown that despite the fact that the future ABI on GOES-R will not have a 550nm green band, it is possible to obtain a good approximation to the CIE 1931 XYZ color space using all of the visible and Near IR bands. We achieved this by combining a multi-linear regression, with $K-$means clustering to achieve selectable level of flexibility. This provided a compromise between the robust multi-linear regression whose performance is limited due to its inability to adapt to different kinds of scenes, with a multi-dimensional lookup table which can provide arbitrary flexibility but whose data requirements explode with the number of input variables. We note that this same method could be used to approximate a virtual sensor with any desired spectral response. In future work we hope to evaluate the method using more data including other hyper-spectral imaging data sets, as well as those derived from model simulations. Would would like to determine how to address the color regions where our current XYZ fails. We would also like to evaluate our XYZ method, for matching to ground based conventional broadband photography to which it is more similar than the narrow band RGB methods. There is nothing fundamentally specific to our algorithm to ABI. It could be applied to other satellites which, like ABI, do not directly produce RGB images.

ACKNOWLEDGMENTS

The authors would like to thank Tim Schmit of NOAA/STAR for his input as well as Alfred Powell and Ingrid Guch of NOAA/STAR for their support. This study was partly supported and monitored by National Oceanic and Atmospheric Administration (NOAA) under ISET Grant # NA06OAR4810187

REFERENCES

[1] Schmit, T. J., Gunshor, M. M., Menzel, W. P., Li, J., Bachmeier, S., and Gurka, J. J., "Introducing the Next-generation Advanced Baseline Imager (ABI) on GOES-R," *Bull. Amer. Meteor. Soc.* **8**, 1079–1096 (August 2005).

[2] Broadbent, A. D., "A critical review of the development of the cie1931 rgb color-matching functions," *Color Research Application* **29**(4), 267–272 (2004).

[3] Hillger, D., Grasso, L., Miller, S., Brummer, R., and DeMaria, R., "Synthetic advanced baseline imager (ABI) true-color imagery," *Accepted for publication by to J. Applied Remote Sensing* (2011).

[4] Miller, S., Schmidt, C., Schmit, T., and Hillger, D., "A Case for Natural Colour Imagery from Geostationary Satellites, and an Approximation for the GOES-R ABI," *Submitted to to International Journal of Remote Sensing* (2011).

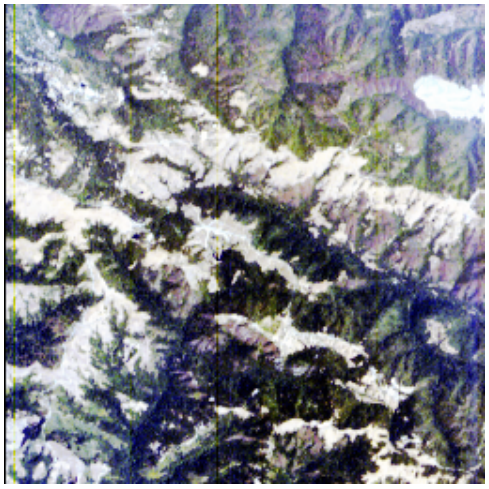
[5] Duda, R. O., Hart, P. E., and Stork, D. G., [*Pattern Classification, S.E*], Wiley Interscience (2000).

[6] van Rossum, G. and Drake, F. L., [*PYTHON 2.6 Reference Manual*], CreateSpace, Paramount, CA (2009).

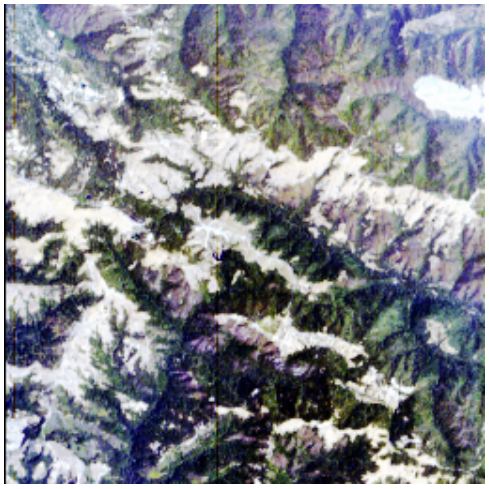
[7] Jones, E., Oliphant, T., Peterson, P., et al., "SciPy: Open source scientific tools for Python," (2001–).

[8] Hunter, J. D., "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering* **9**, 90–95 (May-Jun 2007).

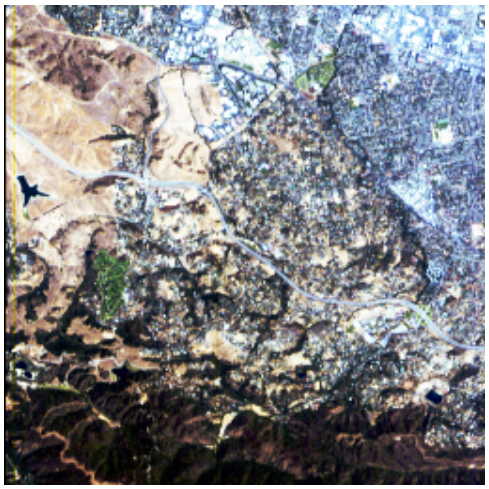
[9] Grasso, L. D., Sengupta, M., Dostalek, J. F., Brummer, R., , and DeMaria, M., "Synthetic satellite imagery for current and future environmental satellites," *Int. J. of Remote Sensing* **29**(15), 4373–4384 (2008).



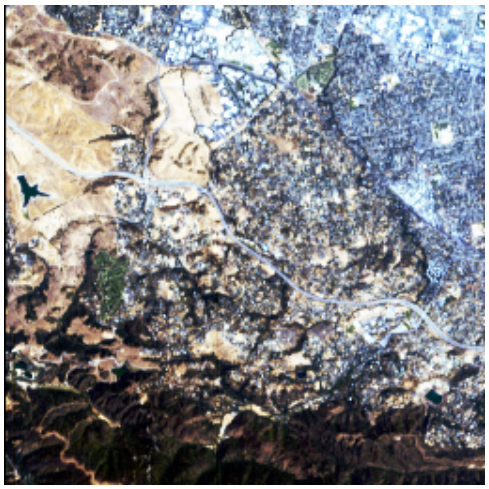
(a) Original



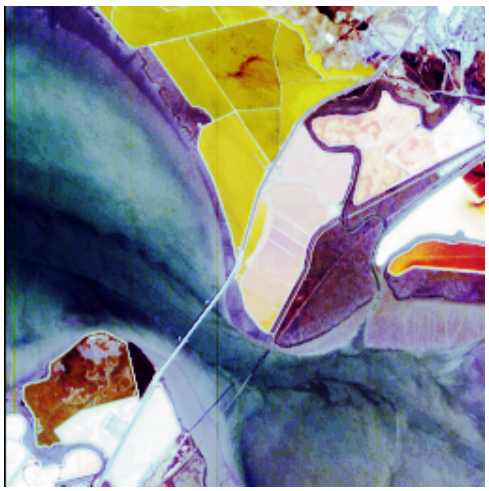
(b) Predicted



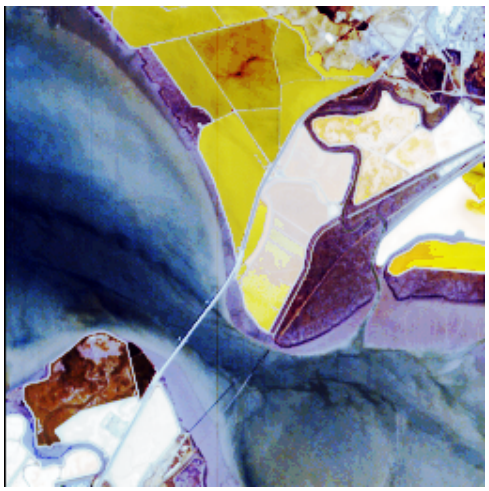
(c) Original



(d) Predicted



(e) Original



(f) Predicted

Figure 11: Original and predicted XYZ images converted to RGB then color equalized to increase contrast. The pairs (a)-(b) and (c)-(d) show excellent aggrement while the pair (e)-(f) was a challenging case.