Syracuse University

SURFACE

Electrical Engineering and Computer Science College of Engineering and Computer Science

Summer 7-5-2012

A Concurrent Spectral-Screening PCT Algorithm For Remote Sensing Applications

Tiranee Achalakul
Syracuse University, tachalak@syr.edu

Stephen Taylor
Syracuse University, Scalable Concurrent Programming Laboratory, steve@scp.syr.edu

Recommended Citation

Achalakul, Tiranee and Taylor, Stephen, "A Concurrent Spectral-Screening PCT Algorithm For Remote Sensing Applications" (2012). *Electrical Engineering and Computer Science*. 62.
https://surface.syr.edu/eecs/62

A CONCURRENT SPECTRAL-SCREENING PCT ALGORITHM FOR REMOTE SENSING APPLICATIONS[1]

Tiranee Achalakul

2-106 CST building, Syracuse University
Syracuse, NY 13244, USA
Tel: 315-443-2226
Fax: 315-443-2126
tiranee@scp.syr.edu, www.scp.syr.edu

Stephen Taylor

2-106 CST building, Syracuse University
Syracuse, NY 13244, USA
Tel: 315-443-2134
Fax: 315-443-2126
steve@scp.syr.edu

[1] This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract N66001-99-1-8922.

ABSTRACT

The paper presents a concurrent algorithm for remote sensing applications that provides significant performance and image quality enhancements over conventional uniprocessor PCT techniques. The algorithm combines spectral angle classification, principal component transform, and human centered color mapping. It is evaluated from an image quality perspective using images collected with the Hyper-spectral Digital Imagery Collection Experiment (HYDICE) sensor, an airborne imaging spectrometer. These images correspond to foliated scenes taken from an altitude of 2000 to 7500 meters at wavelengths between 400nm and 2.5 micron. The scenes contain mechanized vehicles sitting in open fields as well as under camouflage. The algorithm operates with close to linear speedup on shared memory multiprocessors and can be readily extended to operate on multiple, low-cost PC-style servers connected with high-performance networking. A simple analytical model is outlined that allows the impact on performance of practical, application-specific properties to be assessed. These properties include image resolution, number of spectral bands, increases in the number of processors, changes in processor technology, networking speeds, and system clock rates.

Keywords: Image fusion, principal component transform, concurrent algorithm, color mapping scheme

1. Introduction

This paper describes and evaluates a *concurrent spectral-screening PCT algorithm* that can be used for hyper-spectral image fusion in remote sensing applications. The algorithm combines the Principal Component Transform (PCT) with spectral angle classification [7] and human-centered color mapping. In this paper, the algorithm is applied to a 210-channel hyper-spectral image collected with the Hyper-spectral Digital Imagery Collection Experiment (HYDICE) sensor, an airborne imaging spectrometer. The performance and image-quality characteristics of the algorithm are explored on shared memory multiprocessors.

The PCT is a standard tool used in many areas of remote sensing, including hyper-spectral data compression [18], information extraction and fusion, [14] and change detection [8, 12, 21]. The essence of the idea is to summarize and de-correlate a collection of images by removing redundancy and packing the residual information into a small set of images, termed *principal components*. The components are rank ordered by the magnitude of their variances (eigenvalues); therefore, most of the spectral contrast is pushed forward to the first few components, with the additional advantage of noise reduction.

When applied to remote sensing, PCT encounters the problem that it often finds variations that dominate numerically – thus if a variation is common, but not significant, PCT will tend to highlight it; for example, trees in a hyper-spectral image of a forest. To alleviate this problem, we augment the PCT with spectral angle classification prior to the de-correlation process. This has the effect of reducing the importance of an object that occurs frequently in a scene. The classification scheme operates by characterizing objects with a spectral signature and detecting the similarity between these signatures. Signatures that vary significantly are used in the de-correlation process while similar signatures are used only once. For example, a foliated region would have a distinct signature that would be significantly different from that of a mechanized vehicle. Since each is used only once, both trees and vehicles will have equal significance in the resulting principal components.

The final step of the algorithm is to generate a color-composite image from a collection of principal components. To achieve this, we use a human-centered approach that attempts to match the spatial-spectral content of the output image with the spatial-spectral processing capabilities of the human visual system. This approach uses a detection model [9] to map spectral variation into the achromatic, red-green, and blue-yellow opponent channels of the human visual system. Unfortunately, subjective inspection of the output image does not provide a useful quantitative method for comparing image quality. Thus we employ the signal-to-noise ratio (SNR) [17] for quantifying the amount of useful information in the resulting image.

2. Related Research

Hyper-spectral image fusion is the process of combining images from different wavelengths to produce a unified color-composite image, removing the need for frame by frame evaluation to extract important information. Image fusion can be accomplished using a wide variety of techniques that include pixel, feature, and decision level algorithms [5, 6]. At the pixel level, raw pixels can be fused using image arithmetic, band-ratio methods [13], wavelet transforms [10], maximum contrast selection techniques [24], and/or the principal/independent component transforms [2, 18, 22]. At the feature level, raw images can be transformed into a representation of objects, such as image segments, shapes, or object orientations [5, 6]. At the decision level, images can be processed individually and an identity declaration is performed to fuse the results [5, 6].

Many techniques have been explored for remote sensing applications. One of the common ones is image arithmetic (addition, subtraction, multiplication, and division) on the pixel intensity from two or more bands. Band differences or band ratios is the most useful of these approaches. It is often used to enhance spectral reflectance differences for rocks, soils, and vegetation. Unfortunately, it is unclear how to define effective fusion arithmetic for a large number of bands. Empirical selection of arithmetic rules may introduce losses in pixel contrast and important spectral information is thus lost. An alternative fusion technique uses maximum contrast selection. It involves a contrast measurement calculation for each pixel, at each scale and orientation in all spectral bands. Unfortunately, this is impractical for large data sets such as HYDICE. Another method of interest is Wavelet-based image fusion, which requires all spectral bands to be transformed into spatial-frequency domain. Representations in this domain are then combined using algebraic rules to form a single fused data. The results are then inversely transformed to obtain a final visible image. The algebraic rules typically weight images in the source according to the relative importance of specific spectra. One such rule is the maximum selection rule introduced by Burt for combining the coefficients of Wavelet Transform [15]. Once again, the Wavelet image fusion scheme is most effective when used on a small number of images as it is less clear how to define fusion rules for a large number of images. Moreover, since the source images are transformed individually, the computational cost becomes significant.

In many remote sensing data sets, such as HYDICE, although the data set possesses different information in each band, there is high inter-band correlation due to the common features in a scene (e.g. forests, roads, rivers). Therefore, data processing involves a significant degree of redundancy. The PCT algorithm is an approach that may utilize either a correlation matrix or a covariance matrix to de-correlate the source images and thus remove this redundancy [3, 4]. The correlation matrix tends to prevent features with large numerical values from dominating the resulting bands. Although producing unbiased eigenvalues, this often distributes variation over a larger number of the resulting components than the covariance matrix. Our goal is to pack as much information as possible into the first few principal components. Thus in this paper, we choose to work with the covariance matrix.

Although both the PCT and spectral angle classification concepts have been used for multi-spectral imaging, the applications have tended to post-process small numbers of spectra without concern for the time taken to process the data. This goal of the research in this paper is to lay the foundation for utilizing low-cost off-the-shelf multi-processor servers, connected through high-performance networking, to increase the overall throughput to the point where real-time remote-sensing applications are possible.

3. Spectral Angle Classification and PCT

Spectral angle classification is a technique that measures the similarity between spectral signatures of objects in a scene. In an N-dimensional hyper-spectral space, the similarity between two signatures can be determined by calculating the angle between the two associated pixel vectors, X and Y as shown in Figure 1(a). The spectral signatures can then be separated from one another if there is a sufficient difference in their angles as shown in Figure 1(b).

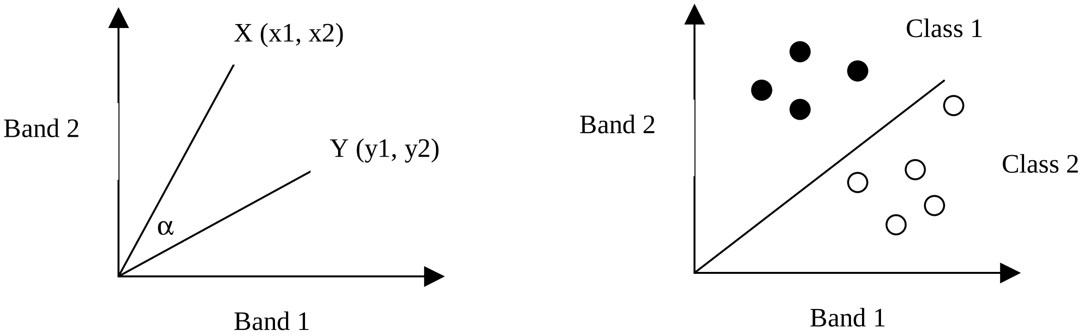


Figure 1. (a) Spectral angle for a two bands image. (b) Classifying spectral space.

The calculation of the spectral angle can be performed by taking the arccosine of the dot product of the two n-dimensinal pixel vectors as shown in the equation below.

$$\alpha(x,y) = \cos^{-1}\left(\frac{x \bullet y}{\|x\| \bullet \|y\|}\right) = \cos^{-1}\left(\frac{\sum_{i=1}^{n} x_i y_i}{\left(\sum_{n=1}^{n} x_i^{\ 2} \sum_{n=1}^{n} y_i^{\ 2}\right)^{1/2}}\right)$$

where n is the number of spectral bands, x = [$x_{1,}$ x_2, x_3, …, x_n] and y = [y_1, y_2, y_3, …, y_n].

For a given spectral angle threshold, α_{thr}, a set of unique spectral signatures is formed by calculating the spectral angle between all the pixel vectors in a hyper-spectral image using dotproduct. The unique signatures set starts with just one pixel vector. Then, for each remaining pixel vector, the spectral angle is calculated with the vectors in a unique set. If all the angles exceed, the threshold the pixel vector is added into a set. It is discarded otherwise. On completion of the process, a unique set of spectral signatures is determined in which the spectral angle between every pair of pixel vectors is more than the threshold, α_{thr}. This unique set is then used, instead of the entire collection of pixel vectors in the hyper-spectral image, in the spectral de-correlation process. The number of unique spectral contained in the final set may vary greatly with different threshold (α_{thr}) used. The appropriate value of α_{thr} is highly application dependent. In our experiment different values of have been explored, and α_{thr} = 6 degree gave the most satisfactory result.

By adding this screening method, we are assured a variation that dominates numerically (backgound) in the original hyper-spectral image, will not dominate the resulting image; small objects in the scene will have an equal chance of being pushed into the front principal components. Recall that the PCT is a linear transformation that de-correlates and compresses a hyper-spectral image into a small set of spectral components. Generally, the covariance matrix is calculated based on all pixel vectors in an hyper-spectral image. Instead we choose to calculate the covariance matrix based on pixel vectors associated with the unique spectral signatures derived through spectral angle classification.

4. Concurrent Algorithm

Recall that spectral angle classification requires the computation of a dotproduct for every pair of pixel vectors in the hyper-spectral image, in the worst case (n^2) vector operations. Moreover, unlike Fourier, Walsh, or Hadamard transforms, the PCT transformation matrix is not separable, and thus, no high performance uniprocessor algorithm exists [16]. We therefore perfer a concurrent approach to performance enhancement. The approach described here is based on *multi-processing,* however, it is our expectation that the concepts can be readily extended to a collection of multi-processing servers connected through high-performance networking. The approach minimizes thread creation overheads by creating an initial set of threads proportional to the number of processors available in a particular machine, and not to the problem size. The concurrent-PCT algorithm described in [19] is then extended to accommodate the spectral screening process as follows:

1. **Spectral classification:** Pixel vectors in a hyper-spectral image are taken as a sequence from the top left to bottom right. The sequence is divided into *P* parts, where *P* is number processors in the system, using integer division. Each part is allocated to a processor. Each processor proceeds concurrently to form a unique spectral set. The operation starts by each proccesor put the first pixel vector in its part into its unique set. The angle calculation is then performed between the remaining vectors and the vectors reside in a unique set. The angle calculation is an arccosine of dotproduct of all pixel vectors pair as shown below.

```
for all p=1 to P conccurently
        for all pixel vectors i in each p {
            for all pixel vectors j in each unique set {
                $\alpha(i, j) = \cos^{-1}(i \bullet j / \|i\| \bullet \|j\|)$

                if(all ($\alpha$(i, j)) > $\alpha_{thr}$ )
                    add(i)
            }
        }

where P = number of parts
      i, j = pixel vector pair
```

2. **Merge unique sets:** The P unique sets are then combined a pair at a time using the algorithm in step 1. Upon completion there will be one unique set left with *K* pixel vectors.

3. **Mean vector:** The n elements in a mean vector of n-bands image are divided into P parts. Each part is allocated to a different processor. Each processor is then calcuated the mean independently. The mean vector is formed once all the processor finish its part.

4. **Covariance Sum:** All the pixel vectors in a unique set are divided into *P* parts, and allocated to *P* processors. Each processor then executes the following code to form a covariance sum:

```
for all p = 1 to P concurrently {
        $sum_p = 0$
        for all pixels (i, j) in each p {
                $C_{ij} = I_{ij} I_{ij}^T - mm^T$
                $sum_p = sum_p + C_{ij}$
        }
}

where P = number of parts
     $sum_p$ = the matrix sum of the covariance in each p
     $I_{ij}$ = pixel vectors in the unique set
```

5. **Covariance matrix:** The covariance matrix is the average of all the matrices calculated in step 4, and is calculated sequentially since its complexity is related only to the number of threads rather than the image size.

6. **Transformation matrix:** The eigenvectors of the covariance matrix are calculated and sorted according to their corresponding eigenvalues which provide a measure of their variances. As a result, the high spectral content is forced into the front components. Since the degree of data dependency of the calculation is high, but its complexity is related to the number of spectral bands rather than the image size, this step is done sequentially.

7. **Transformation of the data:** Each pixel vector, Is_{ij}, in the original hyper-spectral image can be transformed independently. Therefore, pixel vectors are divided equally among all processors, and are transformed concurrently as follows:

```
for all p = 1 to P concurrently
        for all pixels (i, j) in each p
                $Cs_{ij} = A(Is_{ij} - m)$

where P = number of parts
      $Is_{ij}$ = pixel vectors in the original image
      $Cs_{ij}$ = transformed pixel vectors
```

5. Color Mapping

Once the spectral-screening PCT is completed, the next step is to display the results on a monitor. As discussed in the previous sections, the first three Principal Components contain more information than any other combinations of any three bands. For example, in the HYDICE data set the first three components contain more than 97% of the total variance. The task in this step is to assign the first three Principal Components to human color space and then convert these to monitor RGB values.

A large number of color spaces have been proposed in the literature in color vision [23]. In this paper, we choose to work with the luminance/chrominance model favored by Peterson et al. 1993, the YOZ model. The outputs of three cones in the human visual system can be transformed to a Luminance band (Y) and two color-opponent bands (O and Z). The information bandwidth of the human color channels is unequal. The spatial frequency bandwidth of the Luminance channel is much greater than the color opponent channels [1]. This suggest that mapping the first Principal Component into the luminance channel and the second and third Principal Components into chromatic channels of the visual system will provide an efficient utilization of visual system bandwidth. After assigning the Principal Components to YOZ, we transform the YOZ values to monitor RGB values to generate the displayed image. The details of this transform are given below.

The YOZ color matching functions can be directly derived from the standard color matching functions termed XYZ, developed by CIE in 1931 using the transform given in Peterson et al. 1993:

$$[YOZ] = [XYZ]\begin{bmatrix} 0 & 0.47 & 0 \\ 1 & -0.37 & 0 \\ 0 & -0.10 & 1 \end{bmatrix}$$

The luminance channel is Y and the blue channel is just Z. The red-green channel (O) can be given by the following equation, O = 0.47X – 0.37Y – 0.10Z.

A color monitor displays an image in the RGB space, thus, another step required is the YOZ-RGB mapping. According to Boynton, the color space mapping matrix, k, can be derived from the spectral power distribution of the display (intensity at each wavelength), and a color matching function as follows [23]:

$$k = [T\ P]^{-1}$$

where T is any color matching function, and P is the spectral power distribution of the monitor.
In our experiments, we have used the YOZ color matching functions described above, for matrix T and the spectral power distribution of a typical RGB monitor for matrix P. The resultant matrix k is shown below:

$$k = \begin{bmatrix} 0.0079 & 0.0551 & 0.0012 \\ 0.0090 & -0.0156 & -0.0015 \\ -0.0024 & 0.0013 & 0.0093 \end{bmatrix}$$

Using matrix k to convert color space, however, may produce RGB values that are not physically realizable such as negative values. To prevent this problem, we have to scale and normalize the matrix to guarantee valid resulting display values. Since the intensity range along each dimension (Y, O, and Z) is different, each dimension is scaled separately to provide maximum contrast. Firstly, the mapping matrix is scaled to a unit axis along all three dimensions. This is done by calculating the largest scale factor such that the sum of a middle display value (128) and stimulus (Y, O, or Z) fall on the edge of the display range. The scale factors are then used to scale the mapping matrix as shown below.

$$k\_unitAxis = 255 * k * scaleFactor$$

Secondly, the largest possible out of range values in RGB space is calculated from the whole range along every dimension in YOZ space. The maximum value is then used as a global scale to normalize all dimensions. The final mapping matrix, k_final, can be calculated as follows:

$$k\_final = 128 / globalScale * k\_unitAxis = \begin{bmatrix} 0.4387 & 0.4972 & 0.0641 \\ 0.4972 & -0.1403 & -0.0795 \\ -0.1355 & 0.0116 & 0.4972 \end{bmatrix}$$

Differential YOZ input values are used because negative O value indicates green color. The final equation for YOZ to RGB mapping can be stated as follows:

$$[RGB] = (128 + (k\_final * ([YOZ] - 128))) / 256$$

6. Experiments and Discussion

To assess the capability of the algorithm, it was applied to a variety of hyper-spectral image sets generated using Hyper-spectral Digital Imagery Collection Experiment sensor (HYDICE). The experiments used the algorithm to summarize the information from 210 spectral bands in a single, color-composite image; presenting as much information as possible to human observer. The goal of the experiments was to quantify the quality of the resulting image and the performance of the algorithm.

The HYDICE sensor is a 210 channel, 0.4 to 2.5 micron, push-broom airborne hyper-spectral imaging spectrometer. The sensor has been used to collect images at discrete spectral resolution ranges from 3 nm (for the short wavelength) to 10-20 nm (for the longer wavelength). The ground sample distances of the flight range from 0.75 to 3.75 meters with 1:1 aspect ratio at an operating altitude of 2000 to 7500 meters. All of the hyper-spectral images were arranged into a three dimensional data structure of two spatial dimensions and one spectral dimension. The data set presented in this paper is a representative example of a variety of scenes used in our experiments and is a 16 bit integer, 320x320-pixel sub-scene provided by the Spectral Information Technology Applications Center (SITAC).

Figure 2 shows a single hyper-spectral image via a representative sample of frames picked from the 210 spectral bands. This particular scene was chosen because it contains a significant mix of forestry, fields, and roads. In addition, the scene contains both uncovered and camouflaged mechanized vehicles. Notice that at the 524nm there is an image with significant contrast on the forestry and camouflaged vehicles, however, since this image is embedded in a data set of 210 frames, we seek an automated method to obtain this information in a single visible image.

Figure 3 shows a plot of the variance in pixel intensity (contrast) in each principal component generated for this data set with both the standard PCT algorithm and the spectral screening PCT. This plot is typical of wide variety of experiments on other scenes in the data set. Using the spectral-screening PCT, almost 80% of the variance is pushed into the first principal component compared to just 60% for the standard PCT. In both cases, after the first three components there is no significant variance. Thus, it is possible to use only the first three bands to generate the final resulting image.

To quantify the quality of the resulting images, we consider only the first principal. In the human-centered mapping, this component will be mapped directly to the achromatic channel of the human visual system which is the primary channel for distinguishing contrast. Thus consideration of this component yields an effective method for quantifying the level of contrast that will be perceived. To measure this perceived contrast, we follow Ready and Wintz [17] by calculating a relative signal-to-noise ratio:

$$\Delta SNR = SNR_{pct} / SNR_{org}$$

$$SNR_{org} = \sigma_x^2 / \sigma_N^2$$

$$SNR_{pct} = \lambda_1 / \sigma_N^2$$

Thus:

$$\Delta SNR = \lambda_1 / \sigma_x^2$$

where

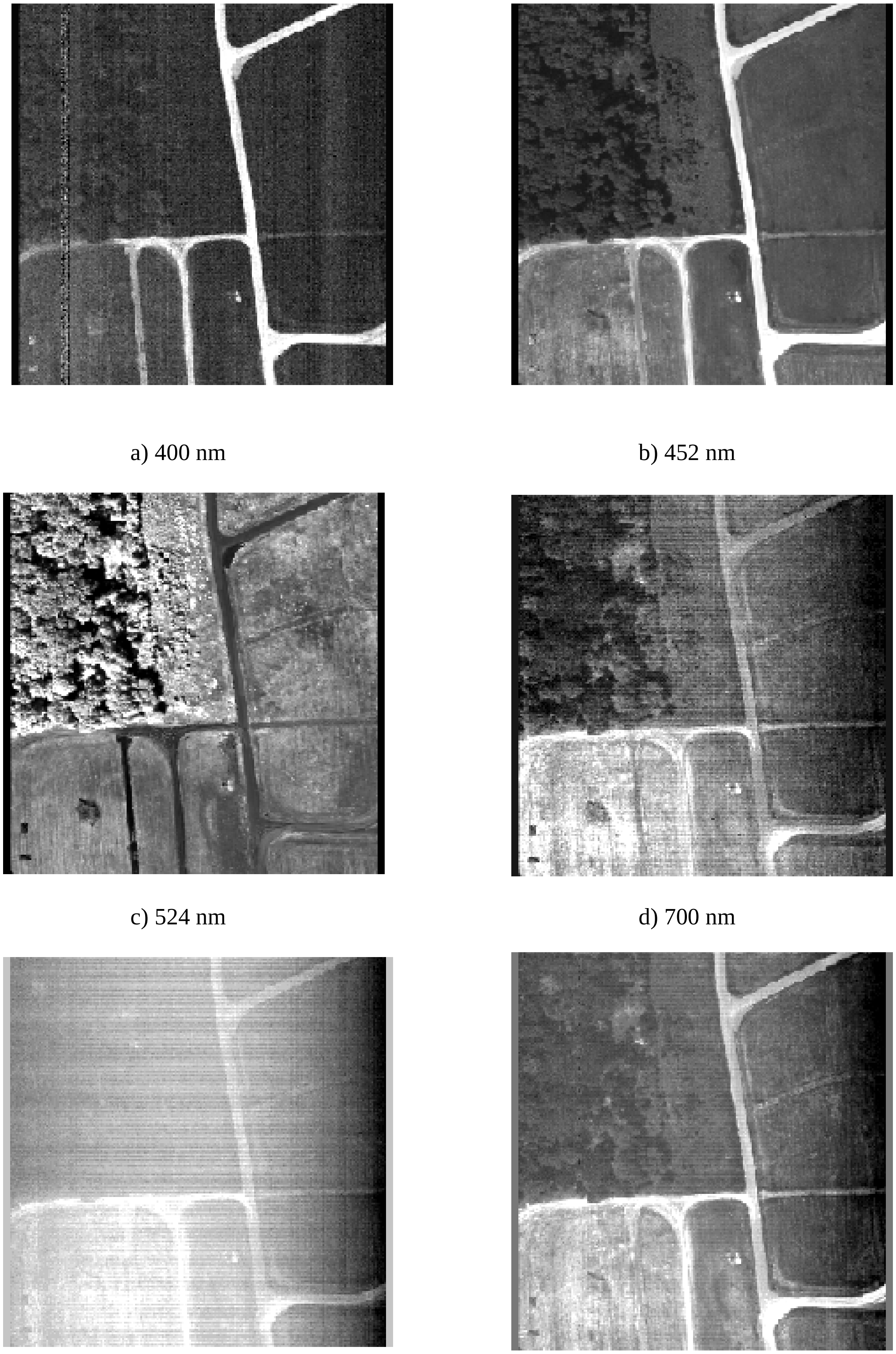
$$\sigma_x^2 = \max(\sigma_1^2, \sigma_2^2, \ldots.., \sigma_n^2)$$

σ_N = variance of noise in an image.

λ_1 = eigenvalues (variance) of the first Principal Component.

Figure 4 shows the first principal component computed using the standard PCT and the spectral screening PCT. The values of the relative SNR were calculated for each image and were 17.79db and 19.58db respectively. The 1.79 dB improvement gained by spectral-screening PCT provides a significant improvement in contrast.

Figure 5 shows the resulting color-composite image after applying the spectral-screen PCT to the full 210 frame data set using two alternative color mapping functions. Figure 5a demonstrates false color mapping in which the first principal component is mapped to red, the second to green, and the third to blue. Figure 5b shows the alternative human-centered mapping presented in section 5; recall that this maps the first principal component to achromatic, the second to red-green opponency, and the third to blue-yellow opponency. The latter picture, when viewed on a high-quality monitor, shows significantly improved contrast levels. The forested areas show significantly improved detail and the camouflaged vehicle in the lower left corner is significantly enhanced against its background. Postprocessing steps can subsequently be applied to detect edges in the image and use structural information to detect and classify the vehicles.



a) 400 nm

b) 452 nm

c) 524 nm

d) 700 nm

e) 997 nm

f) 1998 nm

Figure 2. A set of sample frames from the original hyper-spectral image

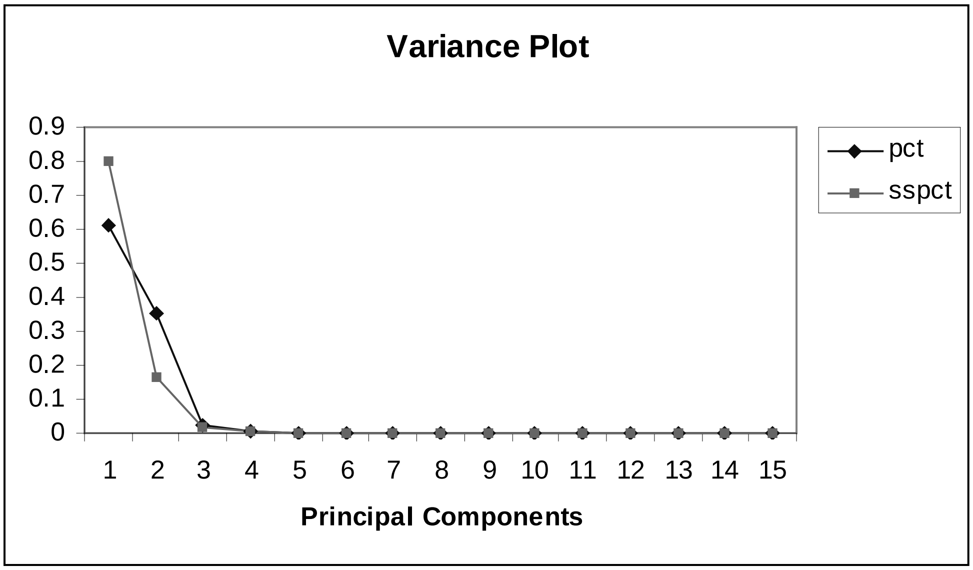
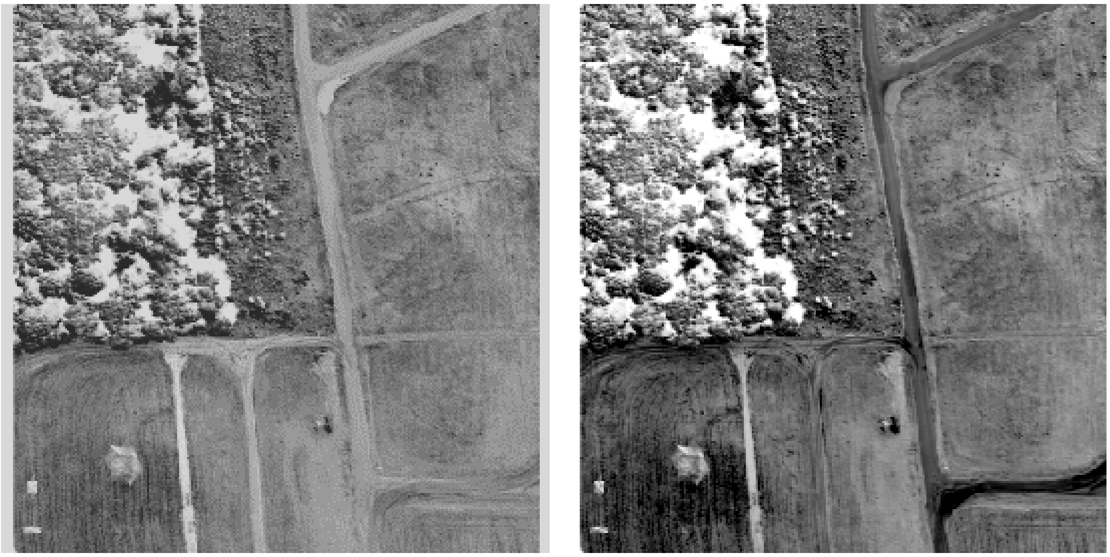


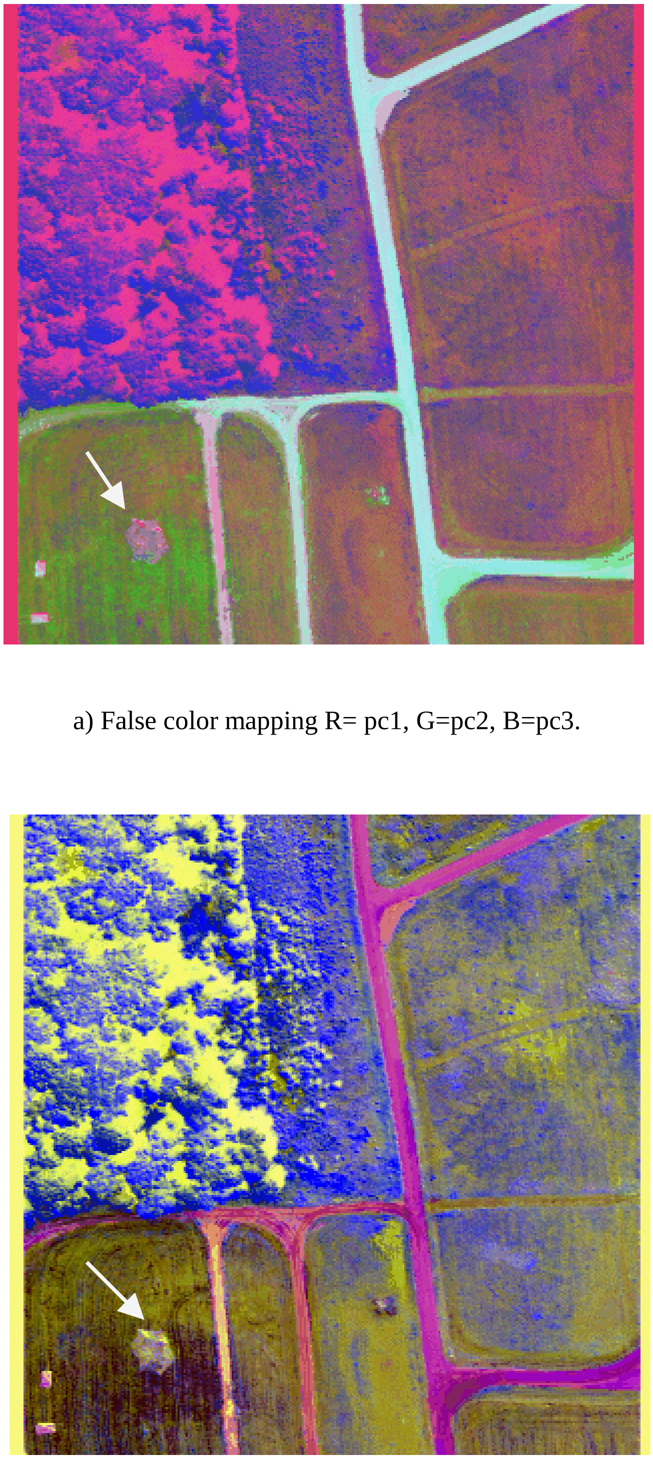
Figure 3.Variances Plot



a) SNR = 17.79 dB

b) SNR = 19.58 dB

Figure 4. The first Principal Component of a) PCT b) spectral-screen PCT



a) False color mapping R= pc1, G=pc2, B=pc3.

b) Color mapping method presented in section 5.

Figure 5. Color mapping result

7. Performance Evaluation and Prediction

The performance of the algorithm when generating the results presented in section 6 were measured on an Intel 8-processor shared-memory PC-server running at 550 MHz. Figure 6 shows the speed up gained as a function of the number of processors, plotted against the ideal speedup. The algorithm operates close to linear speedup with only a 5% degradation at 8 processors. This speed degradation was caused by the sequential code used in the algorithm for finding the eigenvalues (step 6). The eigenvector calculation is related to the number of spectral bands used in the problem. Although the algorithm has a complexity of O(n^3), at the typical problem size of less than 210 frames, the time used is not dominant. Note that this speed degradation *decreases* when either the number of spectral bands or the image resolution is increased. Our experiments show that even for small numbers of spectral bands (8-12) the sequential overhead does not exceed 10%.

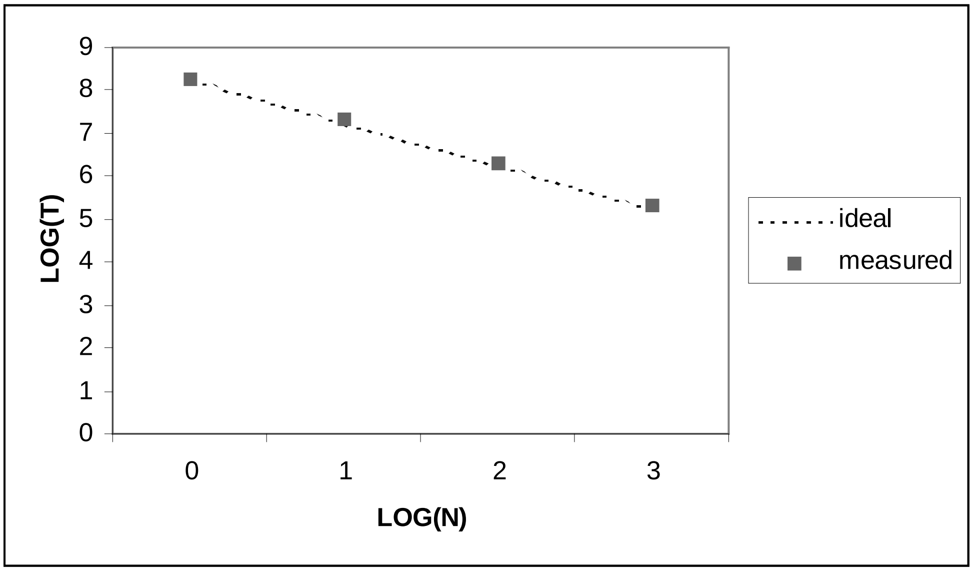


Figure 6. Performance Chart

It can be shown, using techniques developed by Foster et.al. [11, 20], that the performance of the spectral-screening PCT can be described analytically with the following predictive model:

$$T_c = C_1 \frac{snm^2}{p^2} + C_2 \frac{(p-1)s}{p} + C_3 \frac{sn}{p} + C_4 \frac{n^2 s}{p} + C_5 n^2 p + C_6 n^3 + C_7 \frac{n^2 m^2}{p} + C_8$$

where T_c is the total execution time on an n-band hyper-spectral image of size mxm and s is the number of unique spectra in the spectral screening. Each step in the algorithm is represented here as a term in the equation that expresses its general computational complexity. The weighting factors C_1 through C_8 can be obtained experimentally through a sequence of regression tests and least-squares fitting. This describes the gross behavior of the algorithm when executed on a multi-processor system and can be extended though successive refinements to include multiple servers connected through high-performance networking.

This simple model allows the performance of the algorithm to be accurately assessed for changes in image resolution, number of spectral bands, increases in the number of processors, changes in processor technology, networking speeds, and system clock rates; in short, all of the interesting properties of the algorithm on a wide variety of remote-sensing applications. Figure 8 demonstrates the prediction capability of the model. Two problem sizes are shown, n=27 and n=210 spectra. The algorithm readily scales to 128 processors, the size of multi-processor PC architectures that can be expected to emerge over the next 3 to 5 years. Notice that the measured values fall directly on the anticipated curves for the 210 band HYDICE data set (n=210).

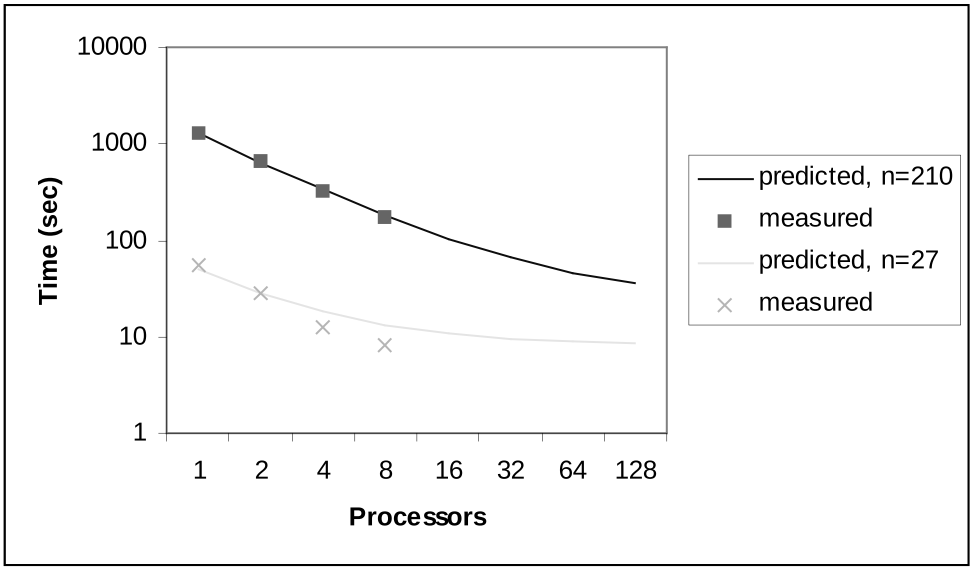


Figure 8.Performance prediction

8. Conclusion

The paper has presented a concurrent spectral-screening PCT algorithm that provides significant performance and image quality enhancements over conventional uniprocessor PCT techniques. The algorithm combines three basic concepts: spectral angle classification, principal component transform, and human centered color mapping. It can be used in a wide range of hyper-spectral image fusion and remote sensing applications and can form a basis for both distributed and real-time remote sensing technologies. An analytical model is outlined that has been validated through extensive tests and that can be used to predict the algorithms performance. It accurately characterizes all of the interesting practical consequences of the algorithm including changes in machine architecture and image characteristics. The algorithm was applied to a 210-channel hyper-spectral image, and the results quantify the improvements in image quality and performance. The results presented here are representative of a wide range of experiments on urban as well as foliated scenes from this data set.

9. References

1. A. B. Poirson, B. A. Wandell, Appearance of Colored Patterns: Pattern-color Separability, Journal of the Optical Society of America, 10, (1993), 2458-2470.

2. A. Mackiewicz and W. Ratajczak, Principal Components Analysis (PCA), Computers & Geosciences, 19, (1993), 303-342.

3. A. Singh and L. Eklundh, A Comparative Analysis of Standardised and Unstandardised Principal Components Analysis in Remote Sensing, International Journal of Remote Sensing, 14, (1993), 1359-1370.

4. A. Singh and A. Harrison, Standardized Principal Components", International Journal of Remote Sensing, 6, (1985), 883-896.

5. D. L. Hall, Mathematical Techniques in Multisensor Data Fusion, MA:Artech House, Boston, 1992.

6. D. L. Hall, An Introduction to Multisensor Data Fusion, in: Proceedings of The IEEE, 85, (1997), 6-23.

7. F. A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and F. H. Goetz, The spectral Image Processing System (SIPS) – Interactive Visualization and Analysis of Imaging Spectrometer Data, Remote Sensing Environment, 44, (1993), 145-163.

8. G. F. Byrne, P. F. Crapper, and K. K. Mayo, Monitoring Land-Cover Change by Principal Component Analysis of Multitemporal Landsat Data, Remote Sensing of Environment, 10, (1980), 175-184.

9. H. A. Peterson, A. J. Ahumada, and A. B. Watson, An Improved Detection Model for DCT Coefficient Quantization, SPIE, 1913, (1993), 191-201.

10. H. Li, B. A. Manjunath, and S. K. Mitra, Multisensor Image Fusion Using the Wavelet Transform, Graphical Models and Image Processing, 57, (1995), 235-245.

11. I. Foster, W. Gropp, and R. Stevens, The Parallel Scalability of the Spectral Transform Method, Monthly Weather Review, 1992.

12. J. A. Richards, Thematic Mapping from Multitemporal Image Data Using the Principal Components Transformation, Remote Sensing of Environment, 16, (1984), 35-46.

13. J. A. Richards, and X. Jai, Remote Sensing Digital Image Analysis: An Introduction, Springer, New York, 1998.

14. J. Lee. and K. Hoppel, Principal Components Transformation of Multifrequency Polarimetic SAR Imagery, IEEE Transactions on Geoscience and Remote Sensing, 30, (1992), 686-696.

15. P. J. Burt, and R. J. Loczynski, Enhanced image capture through fusion, in: Proceedings of the Fourth International Conference on Computer Vision, Berlin, Germany, May 1993, pp. 172-182.

16. P. M. Pardalos, A. T. Phillips, J. B. Rosen, Topics in Parallel Computing in Mathematical Programming, Science Press, New York/Beijing, 1992.

17. P. Ready , and P. Wintz, Information Extraction, SNR Improvement, and Data Compression in Multispectral Imagery, IEEE Transactions on Communications, 21, (1973), 1123-1130.

18. R. C. Gonzalez, R. E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, Inc., New York, 1993.

19. T. Achalakul, P. D. Haaland , S. Taylor, Mathweb: A Concurrent Image Analysis Tool Suite for Multi-spectral Data Fusion, in: SPIE Sensor Fusion: Architectures, Algorithms, and Applications III, Orlando, FL, April 1999, pp351-358.

20. T. Fahringer, Automatic Performance Prediction of Parallel Programs, Kluwer Academic Publishers, Boston/London, 1996.

21. T. Fung. and E. LeDrew, Application of Principal Components Analysis to Change Detection, Photogrammetric Engineering and Remote Sensing, 53, (1987), 1649-1658.

22. T. Lee, Independent Component Analysis: Theory and Applications, kluwer Academic Publishers, Boston, 1998.

23. T. M. Boynton, Human Color Vision, Rinehart, and Winston, New York, 1979.

24. T. Peli, E. Peli, K. Ellis, R. Stahl, Multi-Spectral Image Fusion for Visual Display, in: SPIE Sensor Fusion: Architectures, Algorithms, and Applications III, Orlando, FL, April 1999, pp359-368.