IMPROVED TRANSFORMS FOR THE COMPRESSION OF COLOR AND MULTISPECTRAL IMAGES

Ricardo L. de Queiroz

[1] Xerox Corporation,
800 Phillips Rd., 128-27E, Webster, NY, 14580
queiroz@ieee.org

ABSTRACT

In the compression of color or multispectral imagery, intra pixel color transforms are usually employed to decorrelate planes. It is usually thought that plane decorrelation (such as provided by the KLT) may lead to higher compression. Spatial correlation, however, is usually not considered. We developed a method to devise a pixel-wise color transform that takes into acount spatial correlation and outperforms the KLT. This is done by considering space-color correlation. We aim at decorrelating the data across color planes, but at correlating the data spatially, so that spatial transforms can more easily decorrelate each of the color planes. Experiment results are shown to demonstrate the gains of the proposed transform.

1. INTRODUCTION

Multispectral image data is commonly generated by orbital remote sensing imaging devices, in which earth regions are imaged using multiple bands of the spectrum. Typical color images are also a form of multispectral imaging [1]. In simple terms, a color image is represented as a finite number of color image planes. Each color is obtained by filtering the image (pixel) spectrum and by measuring the resulting luminosity energy. In this way, each pixel color is represented by a few values, corresponding to a few filters. Usually, 3 filters (RGB) are used, but multi-spectral data as well as subtractive spaces such as CMYK use more than 3 channels. Digital color images are pervasive and universal, available to anyone who has ever used a computer to browse the web, for example.

A sampled color image is an array of N-tuples: every pixel is a vector. For a continuous image, another interpretation is that an image is a Riehmanian surface in $(N+2)$D space. For the typical 3-channel color case, the image is a 5D parametric surface, where the parameters may be chosen as the spatial coordinates xy. One and only one point is mapped to each point in the xy plane. To see this, imagine a scan line of the image (1D signal) and one single color signal (monochrome). Then, the "image" is a simple function. If the image is a single 2D plane (monochrome), it is a surface, i.e. it is parameterized in 2D. To add more than one color signal, imagine a 1D signal (scanline) and two color signals. The "image" would be a line in 3D. Extending the space to 2D and the number of color channels to 3, it is easy to conclude that a 3-color image is a surface in 5D parameterized by two out of five axes. This abstraction is useful to re-emphasize that there is correlation not only across color planes, or across space within a color plane, but the correlation has to be analyzed in a 5D space, across both color planes and space coordinates.

Furthermore, it help us analyze color images linearly, using the proper topological representation.

Of interest to us are the properties of color images and of other multispectral data related to compression. In particular, well correlated signals or regions without much detail tend to compress better than more noisy or detailed images, i.e. smooth regions compress better than regions involving too many edges. The objective of this paper is to derive an intra-pixel color transformation that is optimized for compression. In fact, we want to incorporate color and spatial correlation information into the design of the transform in order to outperform a usual decorrelating transform such as the Karhunen-Love transform (KLT) across color planes [1]–[3]. We are concerned with color transforms and not as much with space-color transforms such as those in [4],[5].

2. COLOR PLANES AND SPATIAL CORRELATION

Let us denote the pixel position with the usual vector representation $\mathbf{n} = (n_1, n_2)$. Let there be N planes and let $C_k(\mathbf{n})$ denote the k-th color plane at pixel position (n_1, n_2). Let also $\mathbf{c}(\mathbf{n})$ be a vector containing all such color planes, i.e.

$$\mathbf{c}^T(\mathbf{n}) = [C_0(\mathbf{n}), C_1(\mathbf{n}), \cdots, C_{N-1}(\mathbf{n})]. \qquad (1)$$

A composite vector can be formed as

$$\mathbf{u}^T = [\mathbf{c}^T(\mathbf{n}), \cdots, \mathbf{c}^T(\mathbf{n} + \mathbf{k}_i), \cdots], \qquad (2)$$

where $\mathbf{k}_i \in \chi$ ($1 \leq i \leq K$), where χ is a set determining a neighborhood of K pixels around the origin. A measure of correlation among neighbor pixels across color planes can be made via the autocorrelation matrix of $\mathbf{u}$, i.e.

$$\mathbf{R}_u = E\{\mathbf{u}\mathbf{u}^T\} = \begin{bmatrix} \mathbf{\Gamma}_0 & \mathbf{\Gamma}_{10} & \cdots & \mathbf{\Gamma}_{K0} \\ \mathbf{\Gamma}_{10} & \mathbf{\Gamma}_0 & & \\ \vdots & & \ddots & \\ \mathbf{\Gamma}_{K0} & & & \mathbf{\Gamma}_0 \end{bmatrix} \qquad (3)$$

Note that $\mathbf{\Gamma}_0$ is the plain cross correlation across color planes, while $\mathbf{\Gamma}_{ij}$ represents the cross correlation incorporating spatial displacement given by vector $\mathbf{k}_i - \mathbf{k}_j$.

The following example will illustrate the process. Consider the immediate horizontal, vertical and diagonal neighbors of a given pixel within an RGB image. Then, $\mathbf{c}(\mathbf{n}) = [r(\mathbf{n}), g(\mathbf{n}), b(\mathbf{n})]$, $\mathbf{\Gamma}_h = E\{\mathbf{c}(\mathbf{n})\mathbf{c}^T(\mathbf{n} + [1, 0])\}$, $\mathbf{\Gamma}_v = E\{\mathbf{c}(\mathbf{n})\mathbf{c}^T(\mathbf{n} + [0, 1])\}$, and $\mathbf{\Gamma}_d = E\{\mathbf{c}(\mathbf{n})\mathbf{c}^T(\mathbf{n} + [1, 1])\}$. The correlation matrix is

$$\mathbf{R}_u = \begin{bmatrix} \mathbf{\Gamma}_0 & \mathbf{\Gamma}_h & \mathbf{\Gamma}_v & \mathbf{\Gamma}_d \\ \mathbf{\Gamma}_h & \mathbf{\Gamma}_0 & \mathbf{\Gamma}_d & \mathbf{\Gamma}_v \\ \mathbf{\Gamma}_v & \mathbf{\Gamma}_d & \mathbf{\Gamma}_0 & \mathbf{\Gamma}_h \\ \mathbf{\Gamma}_d & \mathbf{\Gamma}_v & \mathbf{\Gamma}_h & \mathbf{\Gamma}_0 \end{bmatrix} \qquad (4)$$

.

To better illustrate the nature of these matrices, for a typical RGB image whose channels are shown in Fig. 1, the following correlation matrices were found:

$$\mathbf{\Gamma}_0 = \begin{bmatrix} 0.9755 & 0.8720 & 0.7850 \\ 0.8720 & 1.0000 & 0.8991 \\ 0.7850 & 0.8991 & 0.9155 \end{bmatrix}$$

$$\mathbf{\Gamma}_h = \begin{bmatrix} 0.9338 & 0.8348 & 0.7516 \\ 0.8348 & 0.9619 & 0.8649 \\ 0.7516 & 0.8649 & 0.8808 \end{bmatrix}$$

$$\mathbf{\Gamma}_v = \begin{bmatrix} 0.9121 & 0.8147 & 0.7336 \\ 0.8147 & 0.9436 & 0.8480 \\ 0.7336 & 0.8480 & 0.8659 \end{bmatrix}$$

$$\mathbf{\Gamma}_d = \begin{bmatrix} 0.8950 & 0.7999 & 0.7213 \\ 0.7999 & 0.9286 & 0.8353 \\ 0.7213 & 0.8353 & 0.8530 \end{bmatrix}$$

As expected, the intra-pixel cross correlation is the largest followed by the correlation among planes of horizontal or vertical neighbor pixels. If one models the image as an autoregressive process with interpixel correlation ρ, with the above data one might infer the correlation as around $\rho = 0.96, 0.94, 0.93$ for the h, v, d directions, respectively, for all 3 channels. That is a typical correlation value found in monochrome images. What is less expected is the large correlation among samples of different planes and pixels, which can be as much as 0.9. Our experience in compression is that decorrelation often leads to increased compression efficiency. In this example, of course, the RGB planes are clearly not decorrelated. However, one might argue that what really can improve compression is to have "smooth" color planes, i.e. color planes without much spatial detail, or highly correlated planes. Highly correlated images can be near-optimally decorrelated by spatial transforms such as the discrete cosine transform (DCT) [6]. Having said that, one still is left with the task of designing the above $\mathbf{\Gamma}$ matrices in order to improve compression.

3. COLOR TRANSFORMS

In order to decorrelate the samples of $\mathbf{u}$, a transformation $\mathbf{\Phi}$ can be used such that $\mathbf{y} = \mathbf{\Phi u}$ and

$$\mathbf{R}_y = E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{\Phi}\mathbf{R}_u\mathbf{\Phi}^T . \qquad (5)$$

The space-color KLT is the one that diagonalizes $\mathbf{R}_y$. This can be accomplished by choosing the rows of $\mathbf{\Phi}$ as being the eigenvectors of $\mathbf{R}_u$, such that $\mathbf{R}_y$ is a diagonal matrix. In this case, one achieves both intrapixel and spatial decorrelation, simultaneously. The multidimensional transform will outperform the compaction provided by separate color transforms followed by linear spatial transforms (such as DCT) of comparable sizes. However, the spatial and color transforms are often independent, so that each pixel is transformed individually. If the color transform is a linear matrix $\mathbf{Q}$, the process is reduced to constraining the space-color transform $\mathbf{\Phi}$ into a block diagonal matrix:

$$\mathbf{\Phi} = \text{diag}\{\mathbf{Q}, \mathbf{Q}, \dots, \mathbf{Q}\}. \qquad (6)$$

Figure 1. Color planes of an RGB image used as an example.

Each pixel $\mathbf{c(n)}$ is transformed as $\mathbf{d(n)} = \mathbf{Qc(n)}$. In this paper we refer to color transforms as pixel-wise transforms following (6).

Because of the block-diagonal nature of the color transform the correlation matrix $\mathbf{R}_y$ is also a block matrix. Let $\mathbf{R}_y$ have blocks $\mathbf{\Lambda}_{ij}$. Hence,

$$\mathbf{\Lambda}_{ij} = E\{\mathbf{d(n + k}_i)\mathbf{d}^T(\mathbf{n + k}_j)\} = \mathbf{Q\Gamma}_{ij}\mathbf{Q}^T. \qquad (7)$$

The well known intrapixel color KLT is obtained by setting $\mathbf{u} = \mathbf{c(n)}$, so that $\mathbf{R}_u = \mathbf{\Gamma}_0$ and $\mathbf{\Phi} = \mathbf{Q}$ is selected as the matrix containing the eigenvectors of $\mathbf{\Gamma}_0$. The KLT approach is general and should provide good performance for any color space, including multispectral data. The disadvantage of the KLT and similar methods is the fact that one needs to gather the statistics of the image ($\mathbf{R}_u$). For the specific case of color images, there are some useful transforms that provide reasonable plane decorrelation for most typical images. Perhaps, the most used transformation is the one that brings RGB into YCbCr, which is used in JPEG [7] and JPEG 2000 [8]. There are also some proposals for using conventional transforms such as DCT or wavelets to transform the samples across color planes (see for example [5]). Nevertheless, their efficacy is still being studied.

4. SPATIALLY CORRELATED PLANES

How do we chose $\mathbf{Q}$ in order to favor compression? As we discussed, in order to achieve higher compression we might not want $\mathbf{R}_y$ to be diagonal, but, rather, we would like to have all $\mathbf{\Lambda}$ matrices to be diagonal, i.e. minimal cross-plane correlation and large intraplane correlation. Let us assume that the image planes follow the autoregressive (Markov) process model. As we have seen in the example, there might be different correlation coefficients for different members of χ. Nevertheless, within this model we have that $E\{C^2(\mathbf{n})\} = \sigma_c^2$ and $E\{C(\mathbf{n+k}_i)C(\mathbf{n+k}_j)\} = \rho_{ij}\sigma_c^2$. Let us assume the transformed planes $\mathbf{d(n)}$ are Markov processes. We want to design $\mathbf{Q}$ such that $\mathbf{\Lambda}_0$ is diagonalized to some matrix $\mathbf{\Lambda}_0 = \mathbf{D}$. Thus, we would like to design $\mathbf{Q}$ so as to produce

$$\mathbf{R}_y = \mathbf{A} \otimes \mathbf{D} \qquad (8)$$

where $\mathbf{A}$ is a Toeplitz matrix generated by the autocorrelation coefficients ρ_{ij}. For the example we used before,

$$\mathbf{R}_y = \begin{bmatrix} \mathbf{D} & \rho_h\mathbf{D} & \rho_v\mathbf{D} & \rho_d\mathbf{D} \\ \rho_h\mathbf{D} & \mathbf{D} & \rho_d\mathbf{D} & \rho_v\mathbf{D} \\ \rho_v\mathbf{D} & \rho_d\mathbf{D} & \mathbf{D} & \rho_h\mathbf{D} \\ \rho_d\mathbf{D} & \rho_v\mathbf{D} & \rho_h\mathbf{D} & \mathbf{D} \end{bmatrix} \qquad (9)$$

where ρ_h, ρ_v, ρ_d are the correlation coefficients of the transformed planes in the horizontal, vertical and diagonal directions.

The equations we want to satisfy are

$$\rho_{ij}\mathbf{D} = \mathbf{Q\Gamma}_{ij}\mathbf{Q}^T \qquad (10)$$

for $0 \le (i,j) \le K$. We can rewrite them as $\mathbf{D} = \mathbf{Q}\left(\frac{1}{\rho_{ij}}\mathbf{\Gamma}_{ij}\right)\mathbf{Q}^T$ and clearly one matrix $\mathbf{Q}$ cannot always simultaneously diagonalize all the normalized correlation matrices. If one cannot satisfy all conditions, a recourse is to try to satisfy the weighted average. Let

$$\mathbf{G} = \sum_{j=0}^{K}\sum_{i=0}^{K}\frac{1}{\rho_{ij}}\mathbf{\Gamma}_{ij}. \qquad (11)$$

By finding a matrix $\mathbf{Q}$ that diagonalizes $\mathbf{G}$ one diagonalizes the weighted average of the $\mathbf{\Lambda}_i$. The set ρ_{ij} can be derived from the correlation or can be adjusted, i.e. one would diagonalize the following:

$$\mathbf{G} = \sum_{i,j=0}^{K}\alpha_{ij}\mathbf{\Gamma}_{ij}. \qquad (12)$$

We are then left with the task of finding the set of input weights $\{\alpha_{ij}\}$ that maximizes the compression benefits.

5. EXAMPLE

For the example we have been using, the matrix to be diagonalized is

$$\mathbf{G} = \mathbf{\Gamma}_0 + \alpha_h\mathbf{\Gamma}_h + \alpha_v\mathbf{\Gamma}_v + \alpha_d\mathbf{\Gamma}_d. \qquad (13)$$

In this example, the neighborhood defines a 2×2-pixel block. Even though we only consider an intra-pixel (pixel-by-pixel) color transform, it is also necessary to spatially decorrelate the block. For such a small size, we can use a Haar transform (or 2×2 DCT) on the 2×2-pixel block. For 4 pixels, each one represented by 3 planes, so that $\mathbf{u}$ is transformed by a separable space-color transformation: $\mathbf{Q}$ is used intrapixel across planes and the Haar is used intraplane across pixels.

We compare the proposed method, in which $\mathbf{Q}$ is the matrix that diagonalizes $\mathbf{G}$ as in (13) (i.e. $\mathbf{Q}$ contains the eigenvectors of $\mathbf{G}$), with the intrapixel color KLT method (where $\mathbf{Q}$ diagonalizes $\mathbf{\Gamma}_0$). Compression performance was estimated based on the coding gain provided by the non-orthogonal transformation of the 12-tuple vector $\mathbf{u}$[9]. We assumed the signal to be stationary with given autocorrelation matrix $\mathbf{R}_u$ and using a non-orthogonal transformation $\mathbf{\Phi}$. In the proposed scheme, the following weights were found to be optimal: $\alpha_h = -0.2122, \alpha_v = 0.5335, \alpha_d = -1.3503$. While the proposed separable space-color method yields a coding gain of 29.80 dB, the color KLT method only yields a coding gain of 28.57 dB. Note that this is more than 1.2 dB improvement over what was previously thought to be the ideal color transform!

6. CONCLUSIONS

The paper introduces a method to devise a color (intra-pixel) transform for compression that takes into acount the spatial correlation of the image planes. The transform aims at decorrelating the color planes of multispectral image data, while increasing the spatial correlation within each plane. The idea is to increase spatial correlation and leave up to the spatial transform to remove it. Even though spatial characteristics are taken into consideration in the design, the transform is applied only on a pixel by pixel basis. We have shown that it is possible to find a color transform that can outperform the color KLT. Significant improvement was obtained over what was previously thought to be the ideal color transform! Even though in our example, we could achieve more than 1 dB improvement over the KLT. It is clear that non-separable space-color transforms can achieve much better results. As a reference the coding gain for using an optimal non-separable space-color KLT (12×12) would be around 36.87 dB for the same $\mathbf{R}_u$. This is an upper bound for all transforms. It is unclear at this point, however, what the upper bound of a separable transform would be. This is a direction of further exploration. Another direction of further exploration is the direct design of $\mathbf{Q}$ given not only $\mathbf{R}_u$ but also using coding

gain as a cost function and incorporating the spatial transform into the overal design.

REFERENCES

[1] W. Pratt, *Digital Image Processing,* John Wiley and Sons, New York, NY, 1978.

[2] P. De Neve, W. Philips, K. Denecker, I. Lemahieu, "Introducing a decorrelated color space in the lossy compression of pre-press applications," *Proc. 5th Color Imaging Conference,* Scottsdale, AZ, pp. 88-92, 1997.

[3] D. Tretter and C. Bouman, "Optimal transforms for multispectraland multilayer image coding," *IEEE Trans. on Image Processing,* Vol. 4, pp. 296–308, March 1995.

[4] M. Jovovic, "Space-color quantization of multispectral images in hierarchy of scales," *Proc. IEEE Intl. Conf. Image Processing,* 2001.

[5] D. Markman and D. Malah, "Hyperspectral image coding using 3D transforms," *Proc. IEEE Intl. Conf. Image Processing,* 2001.

[6] K. R. Rao, and P. Yip, *Discrete Cosine Transform, Algorithms, Advantages and Applications,* Academic Press, San Diego, CA, 1990.

[7] W. P. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard,* Van Nostrand-Reinhold, 1993.

[8] D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals Standards and Practices,* Kluwer Academic Publishers, 2001.

[9] A. Gersho and R. Gray, *Vector Quantization and Signal Compression,* Kluwer Academic, Norwell, MA, 1992.