1-4-1995

Losslees compression of RGB color images

Nasir D. Memon

Khalid Sayood

Lossless compression of RGB color images

Nasir D. Memon, MEMBER SPIE
Northern Illinois University
Department of Computer Science
DeKalb, Illinois 60115
E-mail: memon@cs.niu.edu

Khalid Sayood
University of Nebraska
Department of Electrical Engineering
Lincoln, Nebraska 68588-0511

Abstract. Although much work has been done toward developing lossless algorithms for compressing image data, most techniques reported have been for two-tone or gray-scale images. It is generally accepted that a color image can be easily encoded by using a gray-scale compression technique on each of the three (say, RGB) color planes. Such an approach, however, fails to take into account the substantial correlations that are present between color planes. Although several lossy compression schemes that exploit such correlations have been reported in the literature, we are not aware of any such techniques for lossless compression. Because of the difference in goals, the best ways of exploiting redundancies for lossy and lossless compression can be, and usually are, very different. We propose and investigate a few lossless compression schemes for RGB color images. Both prediction schemes and error modeling schemes are presented that exploit interframe correlations. Implementation results on a test set of images yield significant improvements.

Subject terms: data compression; lossless image compression; color images; RGB images; scan models; error modeling; composite source models.

Optical Engineering 34(6), 1711-1717 (June 1995).

1 Introduction

Data compression techniques generally consist of two distinct and independent components—modeling and coding.[1] The model captures the inherent structure in the source and the encoder maps the source sequence to a binary string based on the statistics provided by the model. The more efficient data compression schemes encode without any alphabet extension, that is, the source string is encoded symbol by symbol. At every step, the model provides to the encoder a probability distribution for the next symbol, based on which the encoding is performed. Coding schemes are known that perform at or close to the zero-order entropy of the source, and hence the critical task in data compression is that of modeling. The model that is imposed on the source effectively determines the rate at which we are able to encode a sequence emitted by the source. Naturally, the model is highly dependent on the type of data being compressed.

In lossless image compression, the task of modeling is usually split into two stages. In the first stage, a prediction model is used to predict pixel values and replace them by the error in prediction, often called the prediction residual. The resulting image is called the residual image or error image. If the prediction is based on previously transmitted values, then knowing the prediction error and the prediction scheme, the receiver can recover the current pixel and hence the entire original image exactly. Further, if the prediction is somewhat accurate, the residual image consists of pixel values that are independent and identically distributed (IID). Hence, we can convert the two-dimensional array of prediction residuals to a sequence $\{E(t)\}$ by scanning the image in some systematic manner. If the distribution of the residual values can be estimated accurately then the sequence $\{E(t)\}$ can be encoded into a binary sequence by using any of the standard variable length coding techniques, such as Huffman coding or arithmetic coding.

Unfortunately, the residual sequence obtained after prediction is generally not IID. In practice, the distribution of the residual sequence is highly correlated with the spatial content of the image. The distribution of prediction residual is sharply peaked at zero in the more uniform areas of the original image, whereas in more active regions the residual value is different from zero with high probability. Hence, in order to encode the residual image efficiently we need to construct better models that capture the structure that remains after prediction. This step represents the second stage of modeling for lossless image compression and is often referred to as error modeling[2] and the model constructed is called an error model. The error model essentially drives a variable length encoder for coding prediction errors.

Although much work has been done toward developing techniques for each of the two steps mentioned above, i.e., prediction and error modeling, most techniques reported have been for two-tone or gray-scale images (for a survey and classification see Ref. 3). It is generally accepted that a color image can be easily encoded by using a gray-scale compres-

Paper 21084 received Aug. 17, 1994; revised manuscript received Jan. 3, 1995; accepted for publication Jan. 4, 1995.

sion technique on each of the three (say, RGB) color planes. Such an approach, however, fails to take into account the substantial correlations that are present between color planes. Several lossy compression schemes that exploit such correlations have been reported in the literature. But we are not aware of any such techniques that have been reported for lossless compression. Because of the difference in goals, the best ways of exploiting redundancies for lossy and lossless compression can be, and usually are, very different. In this paper we propose and investigate a few simple lossless compression schemes for color images.

In the next section we review some common representations of color images and examine their properties. In Sec. 3 we propose prediction schemes that take into account spectral correlations present in color image data. Section 4 investigates schemes for error modeling that take into account prediction errors incurred at spatially as well as spectrally adjacent pixels. Implementation results of the techniques developed are presented and it is seen that significant improvements in compression ratios can be obtained by the schemes that have been proposed.

2 Color Images

According to the classical and well established tri-chromatic model for color, all colors are seen by humans as variable combinations of the three primary colors, red (R), green (G), and blue (B). A digital color image is formed by three sensors responding to a narrow band of wavelengths centered around 700 nm (red), 546.1 nm (green), and 435.8 nm (blue), respectively. The output of the sensors is digitized by sampling and quantization to give three different digital images. When superimposed for display purposes, these images yield the original color image. Since three signals are digitized simultaneously to yield a color image, any point (pixel) in a color image is determined by three components. Thus, a color image P of size $M\times N$ is an array of vectors

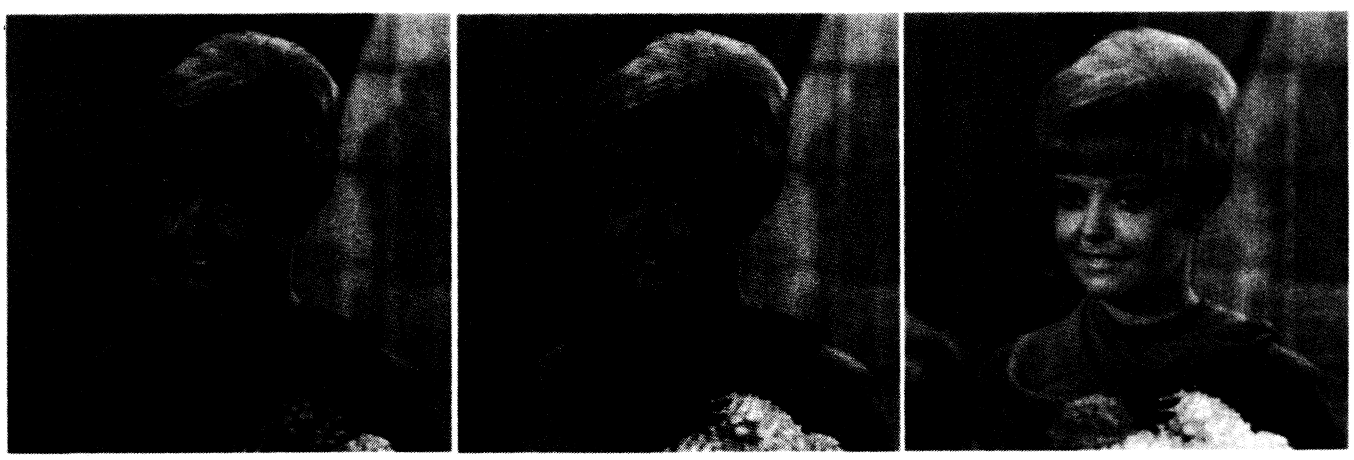
$$P[i,j]=\begin{bmatrix} R[i,j] \\ G[i,j] \\ B[i,j] \end{bmatrix} ,$$

where R, G, and B denote the red, blue, and green frames, respectively.

For the purposes of lossless compression a color image is often viewed as three individual monochromatic images that are compressed independently by a suitable still picture compression technique. Such an approach, however, fails to utilize the significant correlations, which we refer to as spectral correlations, that are present between the three individual color frames. In Fig. 1 we show the R, G, and B frames of the ''USC-Girl'' image. Clearly, the three frames are highly correlated. A number of compression schemes that exploit spectral correlations in color images have been reported in the literature. However, these techniques have all been for lossy compression. Because of the difference in goals, the best way of exploiting spatial and spectral redundancies for lossy and lossless compression is usually very different. Color images, for example, are often transformed to the (YIQ) or (LUV) domains before compression. This is because there is less correlation between components in these domains as compared to the device-oriented RGB domain. In Fig. 2 we show the same ''USC-Girl'' image represented in the LUV space. The interframe correlations have been substantially removed. Further, the human eye is less sensitive to chrominance signals, and hence with lossy compression the IQ or UV components are encoded at lower resolution. Such transformations to a more suitable space are often irreversible due to effect of finite precision arithmetic. Hence, though useful for lossy compression, for lossless compression, irreversible transformations may not be permissible.

Lossy compression techniques that operate in the RGB domain are also of little help when performing lossless compression. Lossy techniques are generally designed to minimize the mean square error between the original and reconstructed images. What is more important in lossless compression is the zero-order entropy of prediction residuals. In addition, the most popular lossy compression techniques that have been used in the RGB domain are based on DCT[4] and vector quantization,[5–8] which are of little utility for lossless compression.

In the rest of this section we propose and investigate some simple techniques for lossless compression of color images



(a) (b) (c)

Fig. 1 "USC-Girl" image: red (a), green (b), and blue (c) planes.

(a) (b) (c)

Fig. 2 "USC-Girl" image in LUV space.

in the RGB domain that exploit both spectral and spatial correlations. We look at images only in the RGB domain, since most imaging sensors output a color image in the RGB format and conversion to another domain usually involves loss of data. Hence, the need for lossless compression of color images is mostly in the RGB domain. However, the techniques that we present are general and could be applied to other domains as well.

3 Prediction Schemes for RGB Images

Given a sequence of data points $x_1, x_2, ..., x_n$, the essential problem in prediction is to predict the next outcome x_{t+1}, for each $t = 1, 2, ..., n$ based only on the data points $x_1, ..., x_t$. Two popular predictive models for lossless coding of images are linear predictive models and context models. In context-based models, data are partitioned into contexts and statistics are collected for each context. These statistics are then used to perform prediction. One problem with such schemes when compressing gray-scale images is that the number of different contexts grows exponentially with the size of the context. Given the large alphabet size of image data, such schemes quickly become infeasible even for small context sizes. Some clever schemes have been devised for partitioning the large number of contexts to a more manageable size.[9,10] Despite such schemes, the complexity of context-based schemes is much higher than linear predictive schemes, which often give comparable performance.

In linear predictive coding, a linear combination of neighboring pixels is taken as the prediction for the current pixel. Although techniques for extracting optimum coefficients with respect to a given linear model are known, for lossless image compression it has been observed in practice that some very simple prediction schemes, such as taking the average of two or three neighboring pixels as the prediction value for the current pixel, give comparable performance to using optimal coefficients with respect to a linear model. The JPEG standard for still image compression uses such simple linear predictors in its lossless mode,[11] shown in Table 1. In Table 2 we show the sum of the zero-order entropies of prediction residuals after using the JPEG prediction schemes on the RGB frames of a test set of 256×256 images taken from the USC database. The RGB frames have been processed independently and the zero-order entropies computed separately for each frame and summed up to give a rate in terms of bits per color pixel. The color pixel is coded in the original image using 24 bits.

Despite the success of linear predictive techniques and context-based techniques in exploiting spatial correlations in an image, both techniques are of little utility for exploiting spectral correlations present in color images. Since spectral dependencies are typically nonlinear, linear techniques are unable to capture such dependencies. Although context-based techniques are not restricted to linear dependencies, the size of the context needed to capture both spatial and spectral correlations gets to be prohibitively large in terms of the resulting complexity, when dealing with gray-scale image data.

In the rest of this section we present a couple of techniques that exploit spatial correlations as well as correlations present between color frames and give significantly lower entropies compared to those shown in Table 2.

Table 1 JPEG predictors for lossless coding.

Mode	Prediction for $P[i,j]$
0	0 (No Prediction)
1	$P[i-1,j]$
2	$P[i,j-1]$
3	$P[i-1,j-1]$
4	$P[i,j-1]+P[i-1,j]-P[i-1,j-1]$
5	$P[i,j-1]+(P[i-1,j]-P[i-1,j-1])/2$
6	$P[i-1,j]+(P[i,j-1]-P[i-1,j-1])/2$
7	$(P[i,j-1]+P[i-1,j])/2$

Table 2 Entropy of error image using JPEG predictors.

Image	JPEG 0	JPEG 1	JPEG 2	JPEG 3	JPEG 4	JPEG 5	JPEG 6	JPEG 7
USC-Girl	19.26	15.03	15.06	15.87	15.32	14.69	14.79	14.28
Couple	18.14	14.47	14.01	15.67	13.82	13.70	13.44	13.68
Girl	16.80	11.49	12.49	12.95	12.23	11.40	12.04	11.51
Lady	21.30	14.59	13.03	15.27	13.09	13.51	12.60	13.27
House	19.20	14.05	15.25	16.03	14.11	13.62	14.14	13.75
Tree	21.52	16.84	17.79	18.06	17.40	16.80	17.04	16.51
Average	**19.37**	**14.41**	**14.61**	**15.64**	**14.32**	**13.95**	**14.01**	**13.83**

3.1 *A Technique Based on Scan Models*

The notion of "adjacency" between pixels in an image is often based on the 4-neighborhood model or the 8-neighborhood model, the adjacency graphs of which, A_4 and A_8, are shown in Fig. 3. An image P induces a weighting function on the edges of an adjacency graph if we assign the weight on an edge to be the difference in intensity values of the two pixels corresponding to the vertices incident upon the edge. We call the weighted version of an adjacency graph induced by an image P to be the difference graph of P. An image and its difference graph using the 4-neighborhood model are shown in Fig. 4.

Given an adjacency graph, we call any spanning tree of the graph a scan model. In Fig. 5 we show two scan models for the image in Fig. 4. Starting from the pixel $P[1,1]$, if we traverse the tree in some fixed order, say depth-first order, then a scan model specifies an order for traversing the pixels of an image, for the given neighborhood scheme. The two scan models shown in Fig. 5 specify two different traversal schemes. It should be noted that standard traversal schemes like the raster scan and the Hilbert scan are special cases of a scan model.

A scan model can also be viewed as a noncausal prediction model for an image. For example, the scan model on the left in Fig. 5 specifies that the prediction for pixel (1,2) should be the intensity value of its neighbor on its left; and the right neighbor of pixel (1,3) is to be used as a prediction for its value and so on. In a similar manner the prediction scheme for each pixel is specified. Hence, given an image P and a scan model T, the set of weights induced by P on the edges of T can be viewed as prediction errors or prediction residuals resulting from using the prediction model corresponding to T on the image P.

It can be seen that finding a scan model that minimizes the absolute weight of prediction errors, which we call a MAW scan model, involves computing a minimum-weight spanning tree of the difference graph that can be computed in time $O(MN \log \log MN)$ for an $M \times N$ image. Furthermore, since a difference graph is sparse, a minimum-weight spanning tree can be computed in time $O(MN \log^* MN)$,[12] which for all practical purposes amounts to $O(MN)$.

Hence, given an image one can efficiently compute a MAW scan model and represent the image by an encoding of a MAW scan model followed by an encoding of the prediction errors incurred by using the scan model. The drawback with this approach when coding single images is that due to the very large number of scan models, the cost of encoding the prediction model may offset the savings made due to better prediction. In Ref. 13 we identify a trade-off between minimizing the number of bits needed to encode the model and minimizing the entropy of prediction errors. Depending on the application, there are many ways of striking a favorable trade-off, leading to novel image compression schemes.[13–16] In the rest of this subsection we present one such technique that is suitable for compression of color images.

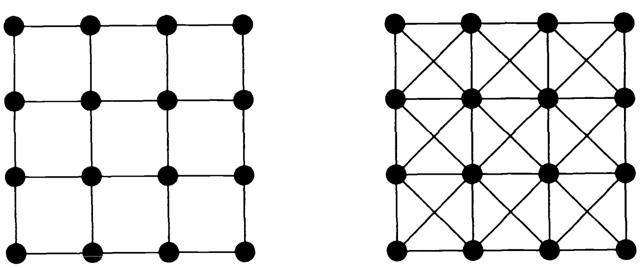


Fig. 3 Adjacency graphs A_4 and A_8 for the 4- and 8-connected models.

1	0	2	1
0	1	2	3
2	3	1	2
1	0	0	1

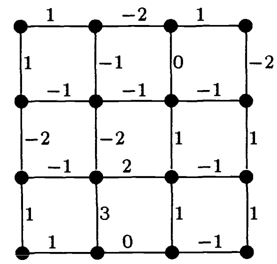


Fig. 4 An image and its difference graph.

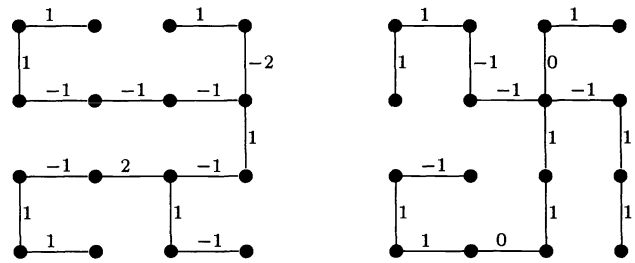


Fig. 5 Two scan models for the image in Fig. 4.

As mentioned before, color images have both spatial and spectral correlations. Correlations between color frames are a result of the fact that the frames are imaging the same physical structures. Thus, while pixel values in a neighboring frame may be very different, the relationships between a pixel and its neighbors may be very similar in adjoining frames. This relationship information is captured well in the scan model. Given a sequence of images that possess a similar structure, we can reasonably assume that optimal scan models for "neighboring" images will be very similar. Therefore, we could use an optimal scan model for one image on the next image in the sequence without significant loss in performance when compared to using optimal models for each image. The first image in the sequence is compressed and transmitted by any conventional method, and subsequent to that, we encode and transmit the residual image obtained by using a MAW scan of the previous image on the current image. While the scan model is being used to exploit spatial correlations, it is being constructed by using the correlations present along the spectral dimension. Hence, we have a simple and efficient backward adaptive technique that exploits both spatial and spectral correlations. By backward adaptive we mean an adaptive technique in which both the transmitter and receiver are in possession of the information necessary for adaptation. This happens when the output of the transmitter (which is also available to the receiver) is used for future adaptation. This has the advantage of obviating any necessity for transmission of additional or "side" information. However, as the current information can only be used for future adaptations, there is necessarily a delay in the adaptation process.

The above scheme was tried on the test images and the results are shown in Table 3. Column 2 gives the zero-order entropy of prediction errors after using the JPEG 7 predictor shown in Table 1. For the test set of images, it gives the best performance among the eight predictors specified by JPEG. Column 3 gives the zero-order entropy of prediction errors when the MAW scan model for the red plane was used on the green plane and the MAW scan model for the green plane was used for the blue plane. JPEG 7 was used for predicting the red plane. We see that almost 1 bit per color pixel savings can be achieved by exploiting spectral redundancies in addition to spatial redundancies.

3.2 *The Previous Best Predictor Technique*

Although we see that computing a minimum-weight spanning tree can be done efficiently,[12,17] in some applications the cost of doing so might be prohibitive. In such situations it may be advantageous to use a simple technique, even at the cost of loss in performance in terms of compression. An example is provided by the previous best predictor (PBP) technique.

The PBP technique is simple. Given a sequence of correlated images and a set of prediction schemes, in order to predict the value of a pixel in the current image, we use the prediction scheme from the set that gives the minimum prediction error when used on the corresponding pixel in the previous image of the sequence. The reason behind using this for color images is clear. We expect the relationships between a pixel and its neighbors to be very similar in adjoining frames of a color image. Hence, a prediction scheme that works well on a particular pixel in one frame will with high probability make an accurate prediction for a pixel at the same spatial location in an adjoining frame.

For RGB images, we first used JPEG 7 to predict pixels in the R frame. Each pixel in the G frame was then predicted by using the best predictor, from a given set of prediction schemes, for the corresponding pixel in the R frame. Similarly the B frame was encoded by using the G frame. Column 4 of Table 3 gives results by using the eight JPEG predictors listed in Table 1 as the set of predictors and selecting the best one in the manner specified above. We see that we can get about 60% of the coding gain obtained by using the MAW scan. We can try to improve the results by determining the predictor from different predetermined sets. Experiments with different sets of predictors were tried. None of them led to any appreciable improvement in performance.

4 Modeling Prediction Errors

If the residual image consisting of prediction errors is treated as an IID source, then it can be efficiently coded using any of the standard variable length entropy coding techniques, such as Huffman coding or arithmetic coding. Unfortunately, even after applying the most sophisticated prediction techniques, generally the residual image has an ample structure that violates the IID assumption. Hence, in order to encode the residual image efficiently we need a model that captures the structure that remains after prediction. Most lossless compression techniques that perform error modeling use a composite source model to capture the structure in the residual image.

Table 3 Entropy of prediction errors with different prediction schemes.

Image	JPEG 7	Scan Model	Prev. Best JPEG
USC-Girl	14.28	13.95	14.24
Couple	13.68	12.62	13.03
Girl	11.51	10.18	10.60
Lady	13.27	11.86	12.29
House	13.75	13.05	13.39
Tree	16.51	15.51	15.90
Average	**13.83**	**12.86**	**13.24**

The notion of a composite source model was first described by Berger.[18] In composite source modeling, we model the data as an interleaved sequence generated by multiple sources, called subsources. Each subsource has its own model and associated probability distribution. Each subsequence is treated as an IID sequence generated by the corresponding subsource and can be encoded at rates very close to the zero-order entropy by techniques such as Huffman coding or arithmetic coding. Error modeling schemes that use a composite source model typically consist of the following two components either in an implicit or explicit manner:

1. a family of probability mass functions, one for each subsource, that is used to construct a variable length code
2. a switching scheme that indicates which particular distribution is to be used to encode a particular pixel (or block of pixels).

We have a number of options for each component. In terms of the probability mass function, we could use the same set of probability mass functions for all images. This would be a static approach. We could adapt the probability mass functions after we encounter each symbol. The same information would also be available to the receiver/decoder so it could be adapted in the same manner. This is referred to as an online or backward adaptive technique. Finally, we could scan the entire image, and construct probability mass functions for each subsource, which could then be sent to the decoder as side information. This is referred to as an off-line or forward adaptive technique. The development of probability mass functions is often integrated with the encoding function, as in the case of adaptive Huffman coding and adaptive arithmetic coding.

Similarly, we can make switching decisions based only on what is available to both encoder and decoder (on-line or backward adaptive), or we could make switching decisions based on information available only to the encoder. The switching decisions could then be provided to the decoder as side information.

The switching techniques studied in this work were all backward adaptive. The reason for this choice was the wealth of information, in the form of previously encoded bands, available to both the encoder and decoder. The family of probability mass functions can be developed either in a forward or backward adaptive manner. The results reported here are in terms of the entropy of the composite source.

We constructed a composite source model consisting of eight sources. Each source represents a different level of activity that could be present in a given region of an image.

We used four different measures of activity. Recall that in the more active regions of the image the magnitude of the prediction residuals will be large, while in the quasi-constant regions of the image the magnitude of the prediction residuals will be smaller. The four different measures used to check the activity level were:

- HVN: the average magnitude of the prediction residual immediately above and the prediction residual immediately to the left of the pixel being encoded
- HPF: magnitude of the prediction residual at same spatial location as the current pixel, but in the previous frame
- COMB: average of HVN and HPF
- HPB: average magnitude of a $k \times k$ block of prediction residuals in the previous frame, centered around the spatial location of the current pixel.

These measurements are compared to a set of eight predetermined thresholds $T_1 < T_2, < ..., < T_8$. Subsource i is assumed to be active if the activity measure was less than or equal to T_i but greater than T_{i-1}.

In Table 4 we show the composite source entropy achieved with different switching techniques for encoding the test images. The first column labeled SSE gives the single source entropy, that is, the sum of the zero-order entropy of the three frames after using the scan-model-based prediction scheme described in the previous section. The rest give results for the composite source model with eight subsources, using the four different activity measures to perform the switching. In all cases the thresholds were picked to be 0, 1, 2, 4, 8, 16, 32, and 256. These thresholds were picked rather arbitrarily, and we expect to get better performance with a more intelligent choice of thresholds.

We see from Table 4 that more than half a bit improvement can be obtained by the use of a composite source model as opposed to using a single distribution for modeling the source. About half of this gain is obtained by simply going to a composite source mode, while the remaining half comes from the use of information in the previous frame. This gain is on top of the gain obtained from using neighboring frames to generate predictions. We would like to repeat here that the numbers given in Table 4 correspond to using information about prediction errors in a neighboring frame for the green and blue frames only. The red frame was encoded using only intraframe correlations (HVN technique). This is because exploiting spectral correlations is only possible for two of the three frames given the backward adaptive nature of our techniques. In our work we have assumed that the red frame is decoded first and hence information about prediction errors in other frames is not available for improved error modeling of prediction errors in the red frame.

Table 4 Entropy of prediction errors with different error modeling schemes.

Image	SSE	HVN	HPF	COMB	HPB
USC-Girl	13.95	13.62	13.56	13.49	13.42
Couple	12.62	12.28	12.14	11.95	11.89
Girl	10.18	9.90	9.76	9.70	9.63
Lady	11.86	11.59	11.51	11.53	11.49
House	13.05	12.67	12.71	12.52	12.30
Tree	15.51	15.10	14.87	14.85	14.60
Average	**12.86**	**12.53**	**12.43**	**12.34**	**12.22**

5 Summary and Conclusions

In this paper we have proposed and investigated some simple techniques for lossless compression of color images in the RGB domain that exploit both spectral and spatial correlations. Techniques for prediction and error modeling were presented that made use of information in the neighboring color frame to give enhanced performance. The prediction technique made use of spectral correlations by using a scan model to capture the inherent structure present in the image. A simple approximation to this scheme was also presented. Error modeling was done by using a composite source model and using prediction errors in neighboring color frames to switch between subsources. Implementation results with the proposed prediction and error modeling techniques gave a total improvement of around 1.5 bits per color pixel over the lossless JPEG standard for a test set of images. Several aspects of the proposed techniques are ad hoc and can be improved upon. This is especially true in the error modeling step. However, the results do show that there are significant advantages to be gained from taking into account spectral correlations in the lossless compression of color images.

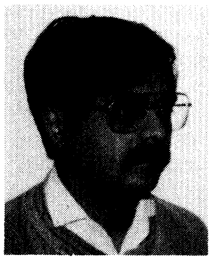
Acknowledgments

The first author wishes to acknowledge the support of the Arkansas Science and Technology Authority through grant 94-B-37. The second author was supported by the Goddard Space Flight Center (NAG 5-1612).

References

1. J. J. Rissanen and G. G. Langdon, "Universal modeling and coding," *IEEE Trans. Information Theory* **27**(1), 12–22 (1981).
2. P. Howard and J. S. Vitter, "Error modeling for hierarchical lossless image compression," in *Proc. Data Compression Conference,* J. A. Storer and M. C. Cohn, Eds., pp. 269–278, IEEE Computer Society Press (1992).
3. N. D. Memon and K. Sayood, "A taxonomy for lossless image compression techniques," in *Proc. Data Compression Conference,* p. 526, IEEE Computer Society Press (1994).
4. A. Zaccarin and B. Liu, "A novel approach for coding color quantized images," *IEEE Trans. Image Processing* **2**(4), 442–453 (1993).
5. Y. S. Feng and N. M. Nasrabadi, "Dynamic address-vector quantisation of RGB color images," *IEE Proceedings-I* **138**(4), 225–231 (Aug. 1991).
6. H. H. Hang and B. G. Haskell, "Interpolative vector quantization of color images," *IEEE Trans. Communications* **36**(4), 465–470 (1988).
7. C. Y. Wang and L. W. Chang, "Color image coding using variable blocksize vector quantization in (R, G, B) domain," *Visual Communications and Image Processing '92, Proc. SPIE* **1818,** 512–523 (1992).
8. H. Yamaguchi, "Efficient encoding of colored pictures in R, G, B components," *IEEE Trans. Communications* **32**(11), 1201–1209 (1986).
9. P. G. Howard and J. S. Vitter, "New methods for lossless image compression using arithmetic coding," in *Proc. Data Compression Conference,* J. H. Reif and J. A. Storer, Eds., pp. 257–266, IEEE Computer Society Press (1991).
10. S. Todd, G. G. Langdon, and J. J. Rissanen, "Parameter reduction and context selection for compression of gray scale images," *IBM J. Res. Develop.* **29**(2), 188–193 (1985).
11. G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM* **34**(4), 31–44 (1991).
12. M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM* **34**(3), 596–615 (1987).
13. N. D. Memon, K. Sayood, and S. S. Magliveras, "Scan models for image compression," submitted for publication.
14. N. D. Memon, K. Sayood, and S. Magliveras, "A simple technique for enhancing the performance of lossy plus lossless image compression schemes," *J. Electronic Imaging* **2**(3), 245–252 (1993).

15. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless compression of multispectral image data," *IEEE Trans. Geosciences and Remote Sensing* **32**(2), 282–289 (March 1994).
16. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless image compression with a codebook of block scans," *IEEE J. Selected Areas of Communications* **13**(1), 24–31 (1995).
17. D. Cheriton and R. E. Tarjan, "Finding minimum spanning trees," *SIAM J. Computing* **5**(4), 724–741 (1976).
18. T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression,* Prentice-Hall, Englewood Cliffs, NJ (1971).



Nasir D. Memon received his BE in chemical engineering and MSc in mathematics from the Birla Institute of Technology, Pilani, India, in 1982. He received his MS and PhD degrees from the University of Nebraska, both in computer science, in 1989 and 1992, respectively. He was an assistant professor in the Department of Computer Science, Mathematics and Physics at Arkansas State University from 1992 to July 1994. He joined the Computer Science Department at Northern Illinois University in August 1994, where he is currently serving as an assistant professor. His research interests include data compression, data encryption, and communications networks.

Khalid Sayood received his undergraduate education at the Middle East Technical University, Ankara, Turkey, and the University of Rochester, New York. He received the BS and MS degrees from the University of Rochester and the PhD degree from Texas A&M University, College Station, Texas, in 1977, 1979, and 1982, respectively, all in electrical engineering. He joined the Department of Electrical Engineering at the University of Nebraska-Lincoln in 1982, where he is currently serving as a professor. His current research interests include data compression, joint source/channel coding, communications networks, and biomedical applications.