

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Journal Articles

Computer Science and Engineering, Department
of

6-28-1993

Simple method for enhancing the performance of lossy plus lossless image compression schemes

Nasir D. Memon
Arkansas State University

Khalid Sayood
University of Nebraska-Lincoln, ksayood1@unl.edu

Spyros S. Magliveras
University of Nebraska-Lincoln

Follow this and additional works at: <https://digitalcommons.unl.edu/csearticles>

Memon, Nasir D.; Sayood, Khalid; and Magliveras, Spyros S., "Simple method for enhancing the performance of lossy plus lossless image compression schemes" (1993). *CSE Journal Articles*. 224. <https://digitalcommons.unl.edu/csearticles/224>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Simple method for enhancing the performance of lossy plus lossless image compression schemes

Nasir D. Memon

Arkansas State University
Department of Computer Science
State University, Arkansas 72467-0070

Khalid Sayood

University of Nebraska—Lincoln
Department of Electrical Engineering
Lincoln, Nebraska 68588-0511

Spyros S. Magliveras

University of Nebraska—Lincoln
Department of Computer Science and Engineering
Lincoln, Nebraska 68588-0115

Abstract. *Lossy plus lossless techniques for image compression split an image into a low-bit-rate lossy representation and a residual that represents the difference between this low-rate lossy image and the original. Conventional schemes encode the lossy image and its lossless residual in an independent manner. We show that making use of the lossy image to encode the residual can lead to significant savings in bit rate. Further, the complexity increase to attain these savings is minimal. The savings are achieved by capturing the inherent structure of the image in the form of a noncausal prediction model that we call a prediction tree. This prediction model is then used to transmit the lossless residual. Simulation results show that a reduction of 0.5 to 1.0 bit/pixel can be achieved in bit rates compared to the conventional approach of independently encoding the residual.*

1 Introduction

Most of the work that has been performed on gray-scale image compression deals with lossy techniques. With lossy techniques the decompressed image is not the same as the original, but only a visual approximation. Achievable compression ratios depend on the quality of the reconstructed image. Current state-of-the-art lossy techniques easily achieve compression ratios between 10:1 and 20:1, on typical real-life images, without noticeable loss in fidelity (in a qualitative sense). For an excellent review of lossy image compression techniques, see Ref. 1.

Despite the existence of high-performance lossy compression techniques, there are many applications that require lossless compression. For example, in medical images no loss of information can be tolerated. This is because what may seem to be noise to a layman may well turn out to be very significant and life-saving information to the trained eye of an expert. The same situation exists with certain satellite and astronomical images in which weak signals are often a scientifically important part of the image. Unfortunately, compression ratios obtained with lossless techniques are not very high. Typically, depending on the image, compression ratios are between 3:1 and 1.5:1. Because of these compression ratios, for certain applications a two-tiered approach is being investigated, in which a lossy *preview* image or *browse* image is first made available to the user. This preview image is a low-rate lossy approximation of the original image, and can, therefore, be provided to the user with efficient use of transmission resources. If the user after preview must view the original image, a losslessly compressed version of the difference between the original and its lossy approximation is then made available. Such schemes are called *lossy plus lossless* (LPL) compression schemes.¹

Generally, LPL schemes do not give as good a compression as other standard methods. However, they are found to be very useful in certain applications. One application is when a user is browsing through a database of images, looking for a specific image of interest. For example, a data browse capability has been declared an essential feature of the Earth Observing System Data and Information Service (EOSDIS), which is projected to become NASA's primary data and information system for the earth sciences, by the turn of the

21	20	22	21	21	1	-2	1
20	21	22	23	1	-1	-1	-1
22	23	21	22	-2	-1	2	-1
21	20	20	21	1	1	0	-1
(a)				(b)			

Fig. 1 (a) A digital image and (b) its differential image.

century.² EOSDIS is unprecedented in size and complexity and will satisfy the needs of a diverse multidisciplinary scientific community. Incorporation of a browse capability will help users check data for presence and location of cloud cover and/or any other anomalies that might affect subsequent analysis. Hence, LPL compression schemes are actively being considered for EOSDIS.

Another application of LPL encoding, in medical imaging, is described by Rabbani and Jones.¹ This application involves patient referral by one physician to another, over a telephone line. Due to the low-channel bandwidth, a lossy approximation of a radiograph is first transmitted to serve the purpose of discussion. The exact image can then be sent at conclusion of the conversation, when the time taken is not critical. Hence, the LPL compression scheme is an ideal choice for such an application.

In this paper, we give a simple technique that can be used to enhance the performance of any LPL compression scheme.

2 Image Decorrelation

We consider a digital image P to be an $M \times N$ array of integers such that $0 \leq P[m,n] < L-1$ for $1 \leq m \leq M$ and $1 \leq n \leq N$. The elements of the array are called pixels and if $P[m,n] = l$, we say that the pixel at location (m,n) has intensity l . In the rest of this paper we shall use P to denote a digital image, M to denote the number of rows in P , and N to denote the number of columns.

The goal of lossless image compression is to represent a given image with the minimum number of bits. This is generally achieved in two steps (Ref. 3). In the first step, statistical redundancy in the image is removed and a residual image is obtained. This step is called decorrelation. For compression to be lossless, the decorrelation step needs to be reversible, that is, the original image should be exactly recoverable from the residual. In the second step, the residual image obtained after decorrelation is encoded into a binary string by means of a variable length code. This step is called coding (Ref. 3).

Most of the compression is normally attained in the decorrelation step. Further, if decorrelation is effective, then the residual image consists of independent and identically distributed pixel values, which can be optimally encoded by well-known coding techniques.^{4,5} Hence, research in lossless compression is focusing more and more on the development of better decorrelation algorithms.

Since neighboring pixels in an image are highly correlated, one simple yet surprisingly efficient way to decorrelate an image is to scan the image row by row going from left to right (raster scan order) and replace each pixel value by its difference with respect to the previous pixel. The first (left-most) pixel in the first (topmost) row is left, and for the remaining rows, the first pixel is replaced by its difference

with respect to the neighboring pixel above. The resulting image of difference values is called a differential image. In Fig. 1 we show a digital image and its differential image. It is clear that the original image can be exactly recovered from the differential image by scanning the differential image in the same order and adding each difference value to the previous pixel to reconstruct the current pixel.

If we assume that pixel values in an image are independent and identically distributed, then the average number of bits needed to losslessly represent the image is the zero-order entropy of the image. If the pixels $P[i,j]$ of an image P take on values from a set $\{a_1, a_2, \dots, a_n\}$, the zero-order entropy $H_0(P)$ of P is defined as

$$H_0(P) = - \sum_{k=1}^n p(a_k) \log_2 p(a_k),$$

where $p(a_k)$ is the probability that a given pixel has the value a_k . The zero-order entropy of the differential image is generally 1 to 2 bits/pixel less than that of the original. Further, it is well known that the distribution of pixel values in the differential image can be modeled well by a zero mean Laplacian distribution.⁶ This enables efficient encoding of the differential by means of a variable length code.

The technique described above is a special case of a larger family of decorrelation schemes known as lossless linear predictive techniques⁷ [also known as lossless differential pulse code modulation (DPCM)]. In predictive decorrelation, a prediction is made for the current pixel, based on the values of previously encountered neighboring pixels. The current pixel is then replaced by the prediction error. The resulting image of prediction errors is called the residual image. A good predictive decorrelation scheme will result in a large number of zeros and small values in the residual image. This also results in lower zero-order entropy for the residual image.

Now it is clear that LPL techniques essentially incorporate a predictive decorrelation step. The lossy image, which is first transmitted, serves as a prediction for the original image. The set of prediction errors, that is the residual image, is transmitted subsequently to achieve lossless reconstruction. In general, an LPL scheme consists of the following steps:

- First generate a low-bit-rate representation of the image by some lossy technique.
- Use this low-bit-rate approximation to decorrelate the image by forming a residual that represents the difference between the original image and the low-rate approximation.
- Transmit the lossy image and follow with variable length encoding of the residual.

LPL techniques that use a discrete-cosine-transform-based lossy step have been investigated in Refs. 1 and 8. The use of the Walsh-Hadamard transform and S transform in the lossy step was investigated in Ref. 9. Also investigated in the same study was subband coding by means of the Smith and Barnwell filter as well as the quadrature mirror filter for obtaining a low-bit-rate approximation of the image. Manohar and Tilton¹⁰ give a vector-quantization-based LPL technique. They found improved performance by iterating this process again on the residual by using a special codebook

for the residual image. They report best performance for three such iterations.

In all the schemes mentioned above, the lossy image and its residual are encoded independently as separate images. However, we show in this paper that information contained in the lossy image can be effectively utilized to encode the residual. Simulation results show that gains of 0.5 to 1.0 bit/pixel can be achieved by the proposed technique.

3 Prediction Trees

Looking at the differential image in Fig. 1, we see that for the pixel at location (2,3), taking the difference with respect to the top neighbor [location (1,3)] would have resulted in a difference value of zero at that location in the differential image instead of -1 . For better decorrelation, it would, therefore, have made more sense to use the pixel at location (1,3) for prediction than the pixel at location (2,2). Similarly, for the pixel at (2,2), the neighboring pixel at (3,3) is the best to take a difference with. Continuing in this fashion one can see that there are many different ways of taking differences, some better than others, with the best one dependent on the specific image being decorrelated. If we call the various differencing schemes prediction models, then for lossless compression, we would like to obtain a prediction model that minimizes the zero-order entropy of the differences. For a small image like the one shown in Fig. 1, we could do this by looking at all possible differencing schemes and selecting the best. However, for larger images, this would not be feasible. Fortunately it is possible to abstract the problem as a graph problem and construct prediction models which, though not guaranteed to have minimum entropy, result in significantly lower entropies than any of the standard prediction models.¹¹⁻¹⁴ Further, the computational costs for finding such models are minimal. In the rest of this section, we briefly describe some of the ideas developed. To do so, however, we need some elementary notions from graph theory. These are listed in Sec. 6, Appendix.

Given a digital image P , we construct its difference graph D by including a vertex for every pixel in P . We then connect vertices representing neighboring pixels with an edge that is weighted with the difference in the intensity values of the corresponding pixels. The difference graph of the digital image in Fig. 1 is shown in Fig. 2. The difference graph shown is obtained by using a four-neighborhood model, that is, pixels immediately to the left, right, above, and below are considered to be neighboring pixels. If we use an eight-neighborhood model then diagonally adjacent pixels are also considered neighbors and additional edges would be added between them.

It is clear that an image P is completely specified by the intensity at any pixel along with any spanning tree T of its difference graph. Hence, we call any spanning tree T , of D , a prediction tree of the image P . In Fig. 3 we give as examples two different prediction trees of the image in Fig. 1.

Any prediction tree T of an image yields a residual image E if we traverse the tree in some order, starting from the vertex representing $P[1,1]$, and replace the intensity at pixel (i,j) by the weight on the edge traversed to reach vertex (i,j) . Note that the residual image E obtained in such a manner from T is independent of the specific traversal that is made of T . Different traversals of T would result in the same residual

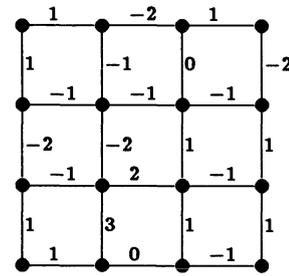


Fig. 2 The difference graph of the image given in Fig. 1.

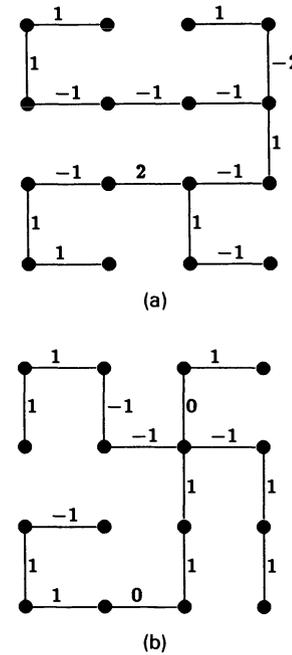


Fig. 3 Two prediction trees for the image in Fig. 1.

image E . In Fig. 4 we show the residual images obtained by using the prediction trees given in Fig. 3.

Now it should be clear that any prediction tree T of an image P defines a decorrelation scheme for P . Therefore, the set of prediction trees of an image form a family of decorrelation schemes for the image. A prediction tree can also be viewed as defining a noncausal⁶ prediction model for the image under consideration. For example, the prediction tree of Fig. 3(a) 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, the set of prediction trees can now also be viewed as a family of prediction models. Naturally we would be interested in a prediction model that maximizes decorrelation. In other words, a prediction model that results in a residual image with minimum entropy.

What is interesting is that given our formulation, the problems related to finding good models can be abstracted as graph problems, that is, problems that involve constructing a spanning tree of the difference graph, with desired properties. This is because the entries in the residual image re-

21	1	1	-2	21	1	0	1
1	-1	-1	-1	1	-1	-1	-1
-1	2	-1	1	1	-1	1	1
1	1	1	-1	1	0	1	1
(a)				(b)			

Fig. 4 Residual images obtained by using prediction trees in Fig. 3.

sulting from using prediction tree T are precisely the weights on the edges of T . Hence, the problem of finding a prediction model that results in a residual image of minimum entropy is equivalent to constructing a spanning tree of the difference graph such that the entropy of the weights on T is minimized. We call such a tree a minimum entropy prediction tree. Reference 11 shows that the problem of computing a minimal entropy spanning tree of a weighted graph is NP-Hard. However, we show that under certain reasonable conditions, minimizing the sum of the absolute weights of prediction errors results in a prediction tree with minimum entropy. We call such a tree a minimum absolute weight prediction tree or, in short, a MAW tree. A MAW tree can be computed in time linear in the size of the image.^{15,16}

A prediction tree that minimizes prediction errors will vary from image to image. Hence, to encode an image, we first encode an optimal prediction model and then encode the image with respect to this optimal model. The drawback with this approach when coding single images is that, due to the very large number of prediction trees,¹¹ the cost of encoding the prediction model may offset the savings made due to better prediction. Reference 11 identifies 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, each leading to a novel image compression scheme.^{11,17} In the next section we show how prediction trees can be effectively utilized to enhance the performance of any LPL compression scheme.

4 Application to Lossy Plus Lossless Compression

In LPL schemes, a low-bit-rate approximation of the image is first generated by some standard lossy technique. This lossy representation can be encoded very efficiently (i.e., at very low rates) and transmitted to the user. The rate of the lossy scheme could be determined by the fidelity and rate constraints of the particular application. The rate of the lossy coding algorithm could also be obtained as the lossy coding rate that would minimize the total (LPL) coding rate. In practice, the former approach would probably be more useful than the latter. After the low-bit-rate lossy approximation of the image has been transmitted, a variable length encoding of the difference between the original image and the approximation is transmitted.

In Table 1 we show the results of using the LPL scheme proposed by Rabbani and Jones¹ with a set of test images. The scheme uses the Joint Photographic Experts Group discrete cosine transform (JPEG-DCT) algorithm¹⁸ as the lossy coding algorithm. All the images are of size 256×256 and

have 256 intensity levels. The images (1) "USC girl," (2) "USC couple," (3) "house," (4) "tree," and (5) "lady" are taken from the University of Southern California database. These images are part of a standard test set used by the image compression research community. The original images are color images with three different color planes—red, green, and blue. We took the green color plane for all the images as our test image. The other three images were present locally. The reason for selecting them was to provide some variety to the test data. One of them is a satellite image and the other an x-ray image. All three are again 256×256 with 256 gray levels.

The particular implementation of JPEG used in this work was a public domain implementation provided by the Independent JPEG Group. This implementation provides an input parameter Q , which controls the quality and bit rate of the compressed image. A value of 10 was used for Q because this generally produced a one-quarter to one-half bit/pixel compressed image. With lower values of Q , the bit rate goes further down, but the quality of the compressed image becomes very poor. The bit rate (in bits per pixel) of the lossy coding scheme is listed in the first column. The second column contains the zero-order entropy $H_0(R)$ of the residual image R . For most of the images, there still may be considerable pixel-to-pixel correlation remaining. We can make use of this correlation by taking differences of neighboring pixels and transmitting the differences. We call the zero-order entropy of the differences, the first-order entropy $H_1(R)$ of the residual image. For the test set these are listed in column three. We see that for most images there is a substantial reduction in the zero-order entropy. This indicates that the decorrelation obtained by subtracting the lossy approximation from the original was not very effective. However, the use of more sophisticated 2-D prediction schemes³ on the residual did not result in a lower zero-order entropy. Column four gives the zero-order entropy of the residual obtained after the decorrelation of the original image by means of a simple but very effective 2-D predictor. The predicted value of each pixel was simply taken to be the average of the pixels above and to the left. It is surprising to see that quite often it may be as good or even better to simply retransmit a compressed version of the original image rather than forming a residual with respect to the lossy representation. This was found to be the case only at low bit rates. However, the figures obtained seem to indicate the futility of the conventional practice of independently encoding the difference between the lossy and the original images, especially at low bit rates. The reader should also observe that the sum of the lossy bit rate and the zero-order entropy of the residual always exceeds the zero-order entropy of the original image after decorrelation by the 2-D predictor. As we mentioned before, LPL schemes do not give as good a compression as the use of some standard technique on the original. However, as already stated, LPL schemes find interesting applications and, hence, are worthy of further study.

The question to ask then is whether the residual image representing the difference between the lossy representation and the original can be further decorrelated? The answer quite often seems to be in the affirmative. We illustrate the reason for this by means of an example. In Fig. 5(a) we show the original "USC-girl" image. The same image coded at a rate of 0.28 bit/pixel using the JPEG compression algorithm is

Table 1 Bit rates in bits per pixel obtained with LPL techniques.

Image	Lossy	$H_0(R)$ (total)	$H_1(R)$ (total)	$H_{2D}(P)$
USC-Girl	0.28	5.04 (5.32)	5.01 (5.29)	4.93
Girl	0.27	4.92 (5.19)	4.69 (4.97)	4.75
Lady	0.22	4.50 (4.72)	3.96 (4.18)	4.02
House	0.30	4.92 (5.22)	4.67 (4.97)	4.64
USC-Couple	0.29	4.88 (5.17)	4.72 (5.01)	4.27
Tree	0.48	5.68 (6.16)	5.65 (6.13)	5.66
Satellite	0.57	5.90 (6.47)	6.03 (6.60)	6.01
X-Ray	0.42	6.00 (6.42)	6.27 (6.69)	6.40

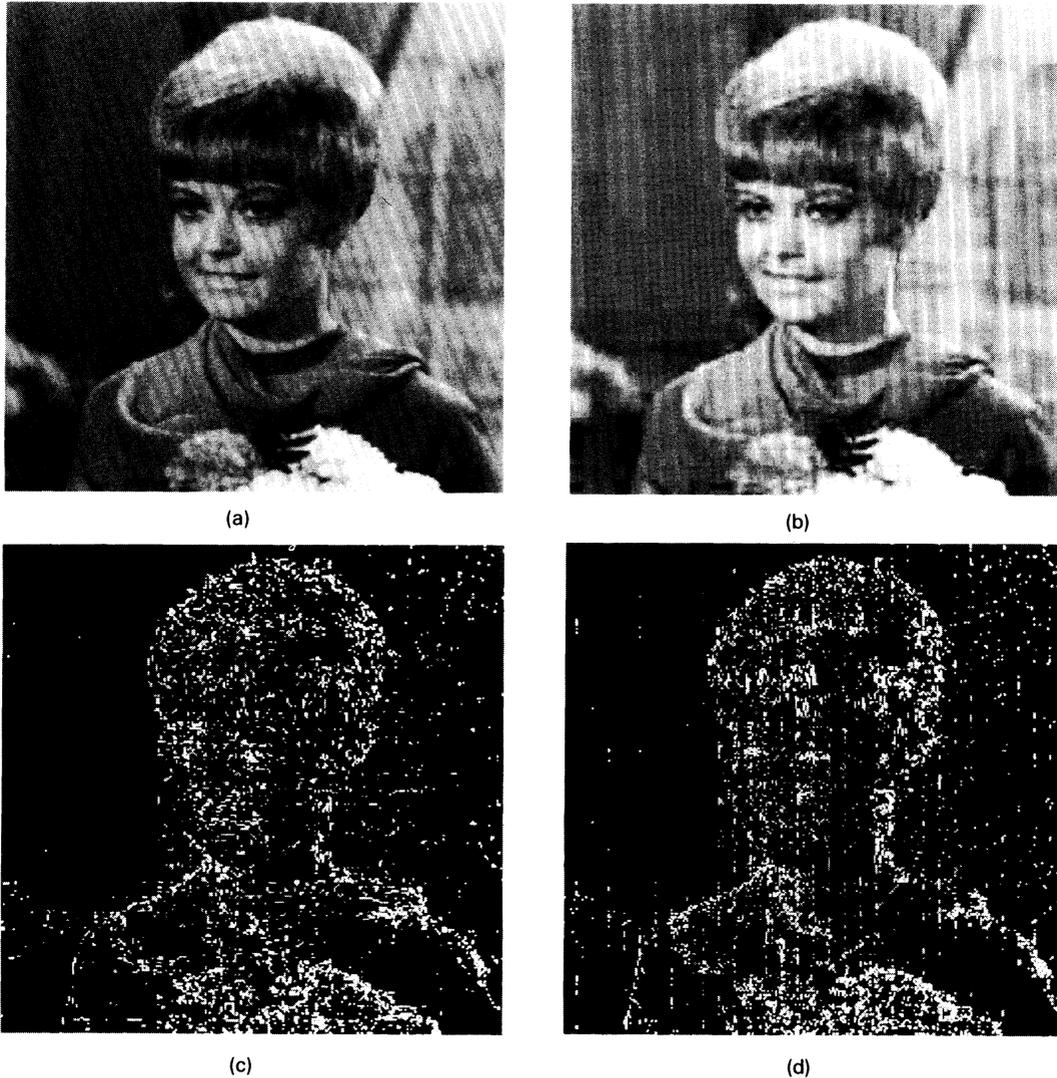


Fig. 5 (a) "USC-girl" original image, (b) lossy version at 0.28 bit/pixel, (c) residual image (thresholded), and (d) differential of the residual image (thresholded).

shown in Fig. 5(b). Although the lossy image is of poor quality, it would definitely serve the purpose of a browse quality image because the objects in the images are clearly recognizable. Based on this lossy representation, a user could choose to obtain the original image by requesting the residual image that is the difference between the original and its lossy reconstruction (assuming both the sender and receiver are

using exactly the same round-off conventions in computing the DCT). This residual image is shown in Fig. 5(c). To improve the presentation of the residual, it was thresholded to a binary image. Pixels with absolute value greater than 16 were mapped to 0 (white) and the ones below 16 were mapped to 1 (black). It is seen that the residual image still exhibits a lot of "structure" and cannot be characterized as a random

image consisting of independent and identically distributed values. The same can be said of the differential of the residual shown in Fig. 5(d).

The distribution of prediction errors in the residual image indicates that the use of the lossy image could lead to improved decorrelation. In Table 2 we show the results of the same experiment, except this time in coding the residual image we have made use of the fact that both transmitter and receiver have a copy of the low-rate lossy representation. Information contained in this lossy image can be utilized to effectively decorrelate and/or encode the difference image. Specifically, in the context of the decorrelation techniques we have presented, a prediction tree can be obtained from the lossy approximation. This tree can now be used to decorrelate either the original image or the difference image, prior to encoding and transmission. Since the lossy image contains the inherent structure present in the image, a MAW tree of it could be expected to decorrelate the original image (or even the difference image) quite effectively.

The results in Table 2 show that there are significant advantages to be gained from this approach. The first column in Table 2 again gives the bit rate for the lossy image obtained. The second column contains the entropy of the weights on the MAW tree computed from the lossy image and used on the original image. The third column contains the entropy of the weights on the same MAW tree used on the residual image. Comparing results from Table 1 and Table 2, we see that considerable savings can be achieved by using the MAW tree of the lossy image on the original image, compared to just transmitting the residual. Note that retransmitting the original image, by means of the MAW tree, gives better results than transmitting the residual in a similar manner. This can lead to some simplification at the receiver end if this approach is used in a browsing application. The receiver does not have to keep a copy of the lossy representation and add on the residual image. It can simply reconstruct the entire image.

As is the case with any lossless coding scheme, the gains of using this approach vary for different images in the test set. However, there always seem to be some savings, and the savings can be as large as 0.75 bit/pixel when compared to the conventional approach of independently encoding the residual image. At this point, we would like to point out that the proposed approach is essentially a new method for decorrelation of the residual obtained in a LPL compression scheme. Coding of this residual has not been addressed. Hence, we use the zero-order entropy as a measure of performance for the decorrelation step. As mentioned before, sophisticated coding schemes exist that can encode at rates close to or even lower than the zero-order entropy.^{1,4,5}

While in most applications the bit rate of the lossy representation would be dictated by the application, we might have a situation where it is possible to use the lossy bit rate that minimizes the total bit rate. Rabbani and Jones¹ report this rate to be around 0.75 bit/pixel. To see if this would hold when using the proposed approach we ran the same experiment with different values of Q on some of the images in the test set. The results obtained with the "USC-couple" image are shown in Table 3. The results for the conventional approach tend to confirm those reported in literature. However, with our approach, while the first-order entropy of the residual goes down with increasing Q , the decrease is not

Table 2 Bit rates in bits per pixel obtained with LPL techniques using prediction trees.

Image	Lossy	$H(T, P)$ (total)	$H(T, R)$ (total)
USC-Girl	0.28	4.84 (5.12)	4.82 (5.10)
Girl	0.27	4.08 (4.35)	4.38 (4.65)
Lady	0.22	3.73 (3.95)	3.77 (3.99)
House	0.30	4.55 (4.85)	4.54 (4.84)
USC-Couple	0.29	4.24 (4.53)	4.27 (4.56)
Tree	0.48	5.44 (5.92)	5.47 (5.95)
Satellite	0.57	5.92 (6.49)	5.87 (6.44)
X-Ray	0.42	5.03 (5.45)	6.07 (6.49)

enough to offset the increase in the lossy coding rate. Even though the total (LPL) rate is still less than the rate from the conventional approach for values of Q below 50, the advantage of the proposed approach goes down with increasing Q . In fact, for values of Q greater than or equal to 50, the conventional approach is slightly better than the proposed approach. This is because as the bit rate for the lossy representation increases, the residual image consists more and more of random noise, and the proposed approach, which depends on residual correlations, loses its advantage. From this we can see that the proposed approach works best in situations that require low rate "browse" images. Because this is generally the case, the proposed approach would perform well in most applications.

5 Conclusions

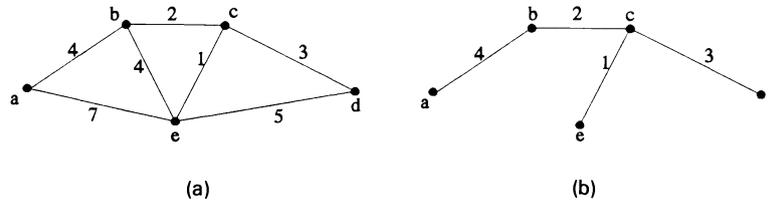
LPL schemes provide an attractive option in applications that have need for quick transmission on the one hand and exact reconstruction on the other. A number of such schemes have been proposed and studied in the literature. However, all schemes known to the authors treat the lossy image and its lossless residual as independent entities. In this paper we have shown that taking into account the lossy image while transmitting the lossless residual can lead to significant savings in coding rates. The savings are achieved by capturing the inherent structure of the image in the form of a noncausal prediction model, which we call a prediction tree. This prediction model is then used to transmit the lossless residual. Implementation results show that significant improvements can be attained by our approach, with a minimal increase in complexity.

6 Appendix

Here we review some elementary notions from graph theory as presented in Ref. 19. A graph G is a finite set of objects called vertices (the singular being vertex) together with a (possibly empty) set of unordered pairs of disjoint vertices of G called edges. The vertex set of G is denoted by $V(G)$, while the edge set is denoted by $E(G)$. If $e = uv$ is an edge of a graph G then u and v are said to be adjacent vertices, while u and e are incident, as are v and e . A graph G is a weighted graph if each edge e is assigned a real number called the weight of e and denoted by $w(e)$. A graph can be described pictorially by representing each vertex by a point (or circle) and each edge $e = uv$ by a line segment (or curve) joining the points u and v . If the graph is weighted then the lines are labeled by the weight of the corresponding edge. Figure 6(a)

Table 3 Bit rates in bits per pixel with different Q values for the "USC-couple" image.

Q	Lossy Bpp	$H_0(R)$ (total)	$H_1(R)$	$H_0(T, P)$ (total)	$H_0(T, R)$
10	0.29	4.88 (5.17)	4.72	4.24 (4.53)	4.27
20	0.42	4.48 (4.90)	4.58	4.15 (4.57)	4.16
30	0.52	4.26 (4.78)	4.49	4.10 (4.62)	4.10
40	0.62	4.12 (4.78)	4.41	4.07 (4.69)	4.07
50	0.72	4.00 (4.72)	4.35	4.03 (4.75)	4.02
60	0.82	3.88 (4.70)	4.28	4.01 (4.83)	3.97

**Fig. 6** (a) A graph and (b) its minimum spanning tree.

shows a weighted graph with vertex set $V(G) = \{a, b, c, d, e\}$ and edge set $E(G) = \{ab, ae, bc, be, ce, de\}$.

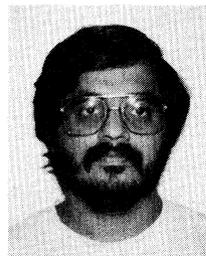
A graph H is a subgraph of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If $V(H) = V(G)$ then H is a spanning subgraph. A path is a sequence of vertices v_0, v_1, \dots, v_n such that $\{v_{i-1}, v_i\} \in E(G)$ for $i = 1, 2, \dots, n$, and $v_i \neq v_j$ for $i \neq j$, $i, j = 1, 2, \dots, n$. If $v_0 = v_n$, then the path is a cycle. A graph G is connected if there is a path between any pair of vertices in G . A graph is acyclic if it has no cycles. A tree is an acyclic connected graph. A subgraph T of G is said to be a spanning tree if T is a tree and a spanning subgraph. If G is weighted, then the sum of the weights on the edges of T is said to be the weight $W(T)$ of T . A spanning tree with minimum weight is the minimum spanning tree of G . Figure 6(b) shows a spanning tree of the graph in Fig. 6(a). The spanning tree shown is a minimum spanning tree.

Acknowledgments

This work was supported in part by the NASA Lewis Research Center under grant NAG 3-806 and by the Center for Communication and Information Science, University of Nebraska—Lincoln.

References

1. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques, Tutorial Texts Series TT07*, SPIE Optical Engineering Press, Bellingham, WA (1991).
2. J. C. Tilton, D. Han, and M. Manohar, "Compression experiments with AVHRR data," in *Proceedings of the Data Compression Conference*, J. A. Storer and J. H. Reif, Eds., pp. 411–420, IEEE Computer Society Press, New York (1991).
3. 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).
4. D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE* **40**, 1098–1101 (1951).
5. J. J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM J. Res. Dev.* **23**(2), 149–162 (1976).
6. A. N. Netravili and B. G. Haskell, *Digital Pictures—Representation and Compression, Applications of Communications Theory Series*, Plenum Press, New York (1988).
7. H. G. Mussman, "Predictive image coding," in *Advances in Electronics and Electron Physics, Suppl. 12*, pp. 73–112, Academic Press Inc., New York (1979).
8. J. R. Cox, S. M. Moore, G. J. Blaine, J. B. Zimmerman, and G. K. Wallace, "Optimization of trade-offs in error-free image transmission," in *Medical Imaging III, Proc. SPIE* **1091**, 1285–1292 (1989).
9. P. Roos, A. Viergever, M. C. A. Van Dijke, and J. H. Peters, "Reversible intraframe compression of medical images," *IEEE Trans. Med. Imag.* **7**(4) 328–336 (1988).
10. M. Manohar and J. C. Tilton, "Progressive vector quantization of multispectral image data using a massively parallel SIMD machine," in *Proceedings of the Data Compression Conference*, J. A. Storer and M. C. Cohn, Eds., pp. 181–190, IEEE Computer Society Press, New York (1992).
11. N. D. Memon, "Image compression using efficient scan patterns," PhD Thesis, University of Nebraska—Lincoln (1992).
12. N. D. Memon, S. S. Magliveras, and K. Sayood, "Prediction trees and lossless image compression—an extended abstract," in *Proceedings of the Data Compression Conference*, J. A. Storer and J. H. Reif, Eds., pp. 83–92, IEEE Computer Society Press, New York (1991).
13. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless image compression using a codebook of prediction trees," in *Proceedings of the Data Compression Conference*, J. A. Storer and M. C. Cohn, Eds., p. 414, IEEE Computer Society Press, New York (1992).
14. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless image compression with efficient scan patterns," presented at the Twenty-Sixth Annual Conference on Information Sciences and Systems, Princeton, New Jersey (March 1992).
15. D. Cheriton and R. E. Tarjan, "Finding minimum spanning trees," *SIAM J. Comput.* **5**(4), 724–741 (1976).
16. M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM* **34**(3), 596–615 (1987).
17. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless compression of multispectral image data," accepted for publication in *IEEE Trans. Geosci. Remote Sensing* (1993).
18. G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM* **34**(4), 31–44 (1991).
19. G. Chartrand and L. Lesniak, *Graphs and Digraphs, Mathematics Series*, 2nd ed., Wadsworth and Brooks/Cole, Belmont, CA (1986).



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 joined the Department of Computer Science, mathematics and physics, at Arkansas State University in 1992, 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.



Spyros S. Magliveras received his BEE and MA degrees at the University of Florida and his PhD in mathematics from the University of Birmingham, England. He has been on the faculty at the University of Nebraska since 1978 and is presently the Henson Professor of Communication and Information Science in the Department of Computer Science and Engineering. Dr. Magliveras has received the AMOCO Award of Distinguished Teaching and is an

honorary senior research fellow at the University of Birmingham. His current research interests include data encryption, combinatorial designs and their applications, and data compression.