

# Edge-Directed Prediction for Lossless Compression of Natural Images

Xin Li, *Member, IEEE*, and Michael T. Orchard, *Fellow, IEEE*

**Abstract**—This paper sheds light on the recent least-square (LS)-based adaptive prediction schemes for lossless compression of natural images. Our analysis shows that the superiority of the LS-based adaptation is due to its edge-directed property, which enables the predictor to adapt reasonably well from smooth regions to edge areas. Recognizing that LS-based adaptation improves the prediction mainly around the edge areas, we propose a novel approach to reduce its computational complexity with negligible performance sacrifice. The lossless image coder built upon the new prediction scheme has achieved noticeably better performance than the state-of-the-art coder CALIC with moderately increased computational complexity.

**Index Terms**—Edge-directed prediction, least-square optimization, lossless image compression, orientation adaptation.

## I. INTRODUCTION

LINEAR prediction is an effective tool for decorrelating stationary Gaussian processes, providing the basis for efficient lossless coding of such sources. However, natural images are characterized by abrupt changes in local statistics and adaptive approaches to linear prediction are needed to fully exploit the dependencies within images. Context-based adaptive prediction schemes [1] have shown significant improvements over fixed prediction schemes such as lossless JPEG [4]. Their representatives include gradient adaptive prediction (GAP) used by the state-of-the-art coder CALIC [3] and median edge detector (MED) adopted by the new lossless image compression standard JPEG-LS [2]. Context-based adaptive prediction can be viewed as the switched prediction that adapts to changing statistics with an experimentally tuned switching function. Recently, a new class of least-square (LS)-based adaptive prediction schemes [6]–[11] have demonstrated impressive improvements over former context-based adaptive prediction schemes. In this paper, we want to shed light on the superiority of the LS-based adaptation and present a novel approach to reduce its computational complexity.

The idea of LS-based adaptation dates back to [5], which improves the performance of lossless JPEG by locally optimizing the prediction coefficients. The computational complexity of optimizing the predictor through the LS optimization process on a pixel-by-pixel basis was regarded as prohibitive, which

impeded its application into practice. Recently, with more powerful computing facilities, the method of LS-based adaptation has been rediscovered independently by several research groups [6], [7], [11]. However, little insight into the superiority of LS-based adaptation has been yielded so far. Here, we provide an interpretation from the viewpoint of its edge-directed property. The dominant role of the pixels around the edge in the LS optimization process makes the predictor adapt reasonably well from smooth regions to edge areas. Illustrative example is used to explain how the support of the predictor is tuned by the LS method to match the edge orientation.

Recognizing that the LS-based adaptation method improves the prediction performance mainly around the edge areas, we propose a novel way of reducing the overall computational complexity. Instead of performing the LS optimization on a pixel-by-pixel basis, we update the predictor coefficients only when the magnitude of the prediction error is beyond a pre-selected threshold. Since the set of the optimal predictors for an edge belongs to the set of optimal predictors for the smooth region, updating the prediction coefficients on an edge-by-edge basis is enough to achieve the gain offered by LS-based adaptation. Therefore we propose to store the predictor coefficients optimized for an edge and repeat using them until the scanning reaches the next edge event. By modestly increasing the memory requirement, we can achieve significant reduction on the computational complexity. Simulation results have shown that a typical gray-scale image with the size of  $512 \times 512$  can be compressed within seconds on a common computing machine.

The rest of this paper is organized as follows. Section II-A starts from the overview of LS-based adaptive prediction schemes. We explain the edge-directed property of the LS optimization and provide some quantitative analysis using an illustrative example in Section II-B. In Section II-C, we present the new approach of reducing the computational complexity. Experiment results are included in Section III to support the efficiency of the new prediction scheme. We make some conclusions in Section IV.

## II. EDGE DIRECTED PREDICTION

### A. Least-Square-Based Adaptation

Throughout this paper, we use  $n$  to denote the spatial coordinate in an image. Suppose the image is scanned in a raster-scanning order; then the prediction of  $X(n)$  is always based on its past causal neighbors (so-called “context”). The motivation behind prediction-based lossless image coders is that if the predictive model can accurately characterize the spatial correlation

Manuscript received December 28, 1999; revised March 2, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon.

X. Li is with Sharp Laboratories of America, Camas WA 98607 USA (e-mail: xli@sharplabs.com).

M. T. Orchard is with the Department of Electrical Engineering, Princeton University, Princeton NJ 08544 USA (e-mail: orchard@ee.princeton.edu).

Publisher Item Identifier S 1057-7149(01)04472-4.

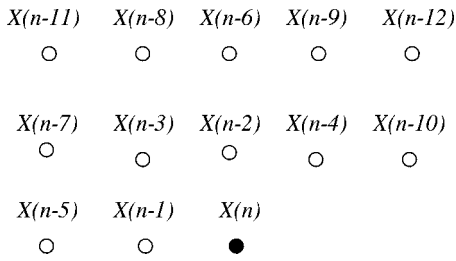


Fig. 1. Ordering of the causal neighbors.

among the pixels, prediction residues would be mostly decorrelated and become easier to model. The major challenge is the development of an accurate predictive model, i.e., how to fully exploit the information contained in the context to resolve the uncertainty of  $X(n)$ .

A reasonable assumption made with the natural-image source is the  $N$ th order Markovian property. That is, we only need to consider the  $N$  nearest causal neighbors in the prediction (as shown in Fig. 1 for the case of  $N = 12$ )

$$\hat{X}(n) = \sum_{k=1}^N a(k)X(n-k). \quad (1)$$

Former context-based adaptive prediction schemes such as GAP [3] and MED [2] employ the edge-detection based switching mechanism to achieve reasonable adaptation from smooth regions to edge areas. However, ad-hoc switching strategies can not effectively handle nontrivially oriented edges and produce large prediction errors around them. Intuitively, an optimal prediction scheme would detect the edge event first and then predict along the edge *orientation*. However, robust edge detection and orientation estimation in an explicit fashion are difficult problems themselves, let alone how to exploit the orientation information during the prediction.

LS-based adaptive prediction schemes [6]–[11] provide an attractive alternative to achieve orientation adaptation and approximate the optimal prediction. Instead of detecting the edge or estimating the orientation explicitly, LS-based approaches locally optimize the prediction coefficients inside a causal window (called “training window”). A convenient choice of the training window is the double-rectangular window, which contains  $M = 2T(T+1)$  causal neighbors, as shown in Fig. 2. Let us denote the training window by a  $M \times 1$  column vector  $\vec{y} = [X(n-1) \cdots X(n-M)]^T$ . Then the prediction neighbors of  $\vec{y}$  would form an  $M \times N$  matrix

$$C = \begin{bmatrix} X(n-1-1) & \cdots & X(n-1-N) \\ \vdots & & \vdots \\ X(n-M-1) & \cdots & X(n-M-N) \end{bmatrix}$$

where  $X(n-j-k)$  is the  $k$ th prediction neighbor of  $X(n-j)$ . The prediction coefficients are solved through LS optimization inside the training window:  $\min \|\vec{y} - C\vec{a}\|_2$ . It is well-known that the LS optimization has a closed-form solution

$$\vec{a} = (C^T C)^{-1} (C^T \vec{y}) \quad (2)$$

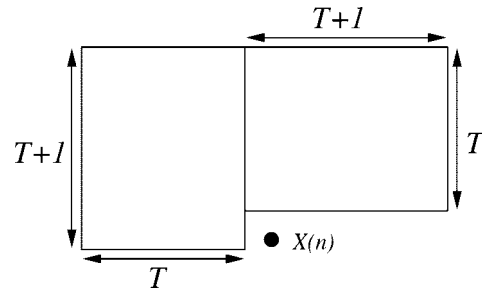


Fig. 2. Training window used to optimize the prediction coefficients.

where  $\vec{a} = [a(1) \cdots a(N)]^T$  is the optimized prediction coefficients.

We can also derive the above result by following the traditional linear prediction theory [14]. It is well-known that the MMSE prediction for a stationary Gaussian process is determined by the second-order statistics (covariance) only

$$\vec{a} = R_{XX}^{-1} \vec{r}_X \quad (3)$$

where

$$\vec{r}_X = [r_1 \cdots r_k \cdots r_N], \quad r_k = \text{Cov}\{X(n)X(n-k)\} \\ (1 \leq k \leq N)$$

and

$$R_{XX} = [R_{kl}], \quad R_{kl} = \text{Cov}\{X(n-k)X(n-l)\} \\ (1 \leq k, l \leq N).$$

Geometrically, (3) is the projection of  $X(n)$  onto the subspace spanned by  $X(n-1) \cdots X(n-N)$  in the LS sense. However, the image source often violates the assumption of stationary Gaussian process. A practical approach to handle such nonstationary source is to instantaneously estimate the local statistics. If we keep the above definition of  $C$  and  $\vec{y}$ , the instantaneously estimated second-order statistics ( $\hat{r}_X$ ,  $\hat{R}_{XX}$ ) can be written as [14]

$$\hat{R}_{XX} = \frac{1}{M} C^T C, \quad \hat{r}_X = \frac{1}{M} C^T \vec{y}. \quad (4)$$

Plugging (4) into (3), we obtain (2) again. To have more insight into the method of LS-based adaptation, we provide the interpretation from the point of view of its edge-directed property, as we shall detail next.

### B. Edge-Directed Property

The effectiveness of any adaptive prediction scheme depends on its capability of adapting from smooth regions to edge areas. The difficulty of achieving ideal adaptation mainly arises from the edge areas because the orientation of an edge could be arbitrary. Though it is easy to imagine that an optimal prediction should always go along the edge orientation, the implementation of such idea is not trivial. As aforementioned, explicit approaches of doing the edge detection and the orientation estimation often have their own limitations (e.g., robustness). In contrast, the approach of LS-based adaptation provides an elegant way of approximating the optimal orientation adaptive prediction due to its edge-directed property.

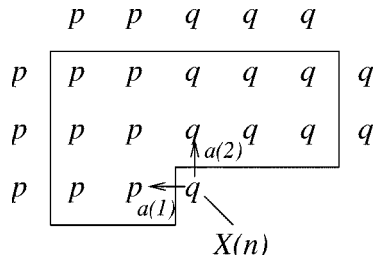


Fig. 3. Example of a vertical edge.

Edge-directed property refers to the dominant role played by the pixels around an edge in the LS optimization process, which gives the name “edge-directed prediction” (EDP). Intuitively, the causal neighbors in the training window can be classified into two classes: the *edge* neighbors (around the edge) and the *nonedge* neighbors (away from the edge). For the nonedge neighbors, the matrix  $C$  is often not full-ranked and the LS optimization does not have a unique solution. In fact, the set of optimal predictors for the nonedge neighbors lies in a hyperplane  $\sum_{k=1}^N a(k) = 1$  in the  $N$ -dimensional space. While for the edge neighbors, the matrix  $C$  is usually full-ranked and the LS optimization does have a unique solution. It is easy to see that the set of optimal predictors for the edge neighbors is a subset of the hyperplane  $\sum_{k=1}^N a(k) = 1$ . Consequently, the edge neighbors would dominate the LS optimization process.

The LS optimization over the whole training window offers a convenient way of finding the optimal prediction coefficients for the edge neighbors without the necessity of edge detection. Moreover, the predictor coefficients optimized for the edge neighbors of  $X(n)$  are also suitable for  $X(n)$  because it belongs to the same edge. Therefore, no explicit estimation of the edge orientation is necessary. In order to incorporate enough edge neighbors into the training window, the window size  $T$  is chosen to be  $\min(N, 7)$ . Empirical studies show that the window size larger than seven does not further improve the prediction performance.

To strengthen our arguments on the edge-directed property, we use a simple illustrative example to quantitatively analyze the relationship between the predictor support and the edge orientation. Fig. 3 shows the example in which the current pixel  $X(n)$  is along a sharp vertical edge ( $|p - q| \gg 0$ ). For simplicity, we only consider the second-order predictor  $\hat{X}(n) = a(1)X(n-1) + a(2)X(n-2) = a(1)p + a(2)q$ , in which case the training window has 12 elements. By some simple derivation, we obtain the estimated covariances from (2)

$$\hat{R}_{XX} = \begin{bmatrix} 8p^2 + 4q^2 & 6p^2 + 2pq + 4q^2 \\ 6p^2 + 2pq + 4q^2 & 6p^2 + 6q^2 \end{bmatrix} \quad (5)$$

and

$$\hat{r}_X = \begin{bmatrix} 6p^2 + 2pq + 4q^2 \\ 6p^2 + 6q^2 \end{bmatrix}. \quad (6)$$

Then it is straightforward to verify that the optimized prediction coefficients are given by

$$\begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (7)$$

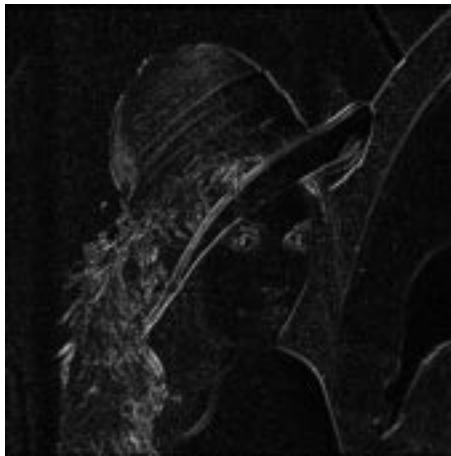
Equation (7) shows that the predictor support is ideally aligned along the edge orientation (the vertical direction). The case of a horizontal edge can be justified in a similar fashion. When the edge is nontrivially oriented and the predictor order is more than two, mathematical derivation becomes complicated. However, our experiments with synthetic images [11] have verified that the LS optimization is capable of tuning the predictor support to match an arbitrarily oriented edge. We have also compared the prediction residue images given by the different adaptive prediction schemes for the real-world images. Fig. 4 shows the amplitude images of prediction residues given by MED, GAP and a tenth-order EDP for the “Lennagrey” image. It can be clearly seen that EDP produces much smaller errors around the edge areas than both MED and GAP.

### C. Computational Complexity

The principle drawback with the edge-directed prediction is its prohibitive computational complexity. The bottleneck of the LS optimization is the computation of the covariance matrix  $C^T C$  in (4). In [5] and [6], conventional “inclusion-and-exclusion” techniques are employed to more efficiently update the estimated covariance matrix. They are based on the observation that the overlapping of the adjacent training windows can be used to speed up the implementation. In this section, we present a novel approach of reducing the computational complexity of the edge-directed prediction while maintaining its performance. The computation savings are achieved by performing the LS optimization only for a fraction of the pixels in the image.

The motivation behind our approach is based on the following two observations with the edge-directed prediction: one is that the prediction coefficients optimized for a pixel around an edge are often also suitable for its neighbors along the same edge; the other is that the set of optimal predictors for an edge is the subset of the set of optimal predictors for the smooth regions. Therefore, the prediction coefficients optimized for an edge can be stored and repeatedly used until the scanning reaches the next edge event. In other words, we want to perform the LS optimization on an edge-by-edge basis rather than on a pixel-by-pixel basis.

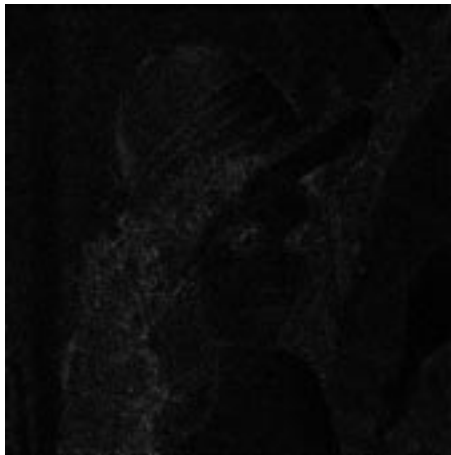
To implement the above idea, we propose the following switching strategy: only if the amplitude of the prediction residue  $e(n) = X(n) - \hat{X}(n)$  is beyond a pre-selected threshold  $th$ , the LS optimization is activated to update the prediction coefficients; otherwise we employ the stored coefficients to predict the next pixel. An implementation in the Pseudo-C code in the one-dimensional (1-D) case is shown in Fig. 5. For two-dimensional (2-D) images, we can use the stored prediction coefficients of the nearest four causal neighbors to generate four prediction values and take their average as the final prediction result. As an example, Fig. 6 shows the locations where the LS optimization is performed in the “Lennagrey” image when  $th$  is set to be eight. Overall, less than 10% of pixels activate the LS optimization, which means the computation savings by an order of the magnitude. Meanwhile, the performance sacrifice brought by the above switching strategy is negligible.



(a)



(b)



(c)

Fig. 4. Residue images of Lennagrey after different prediction schemes. (a) MED ( $H = 4.56$  bpp), (b) GAP ( $H = 4.39$  bpp), and (c) EDP ( $H = 4.22$  bpp).

### III. EXPERIMENTAL RESULTS

In this section, we shall use extensive experiment results to demonstrate 1) the performance of EDP with the different prediction order  $N$  and 2) the tradeoff between the complexity and

```

% initialize
flag=0; a_save(i)=1/N(i=1~N)
% main loop
while(scanning unfinished)
{
  if(flag==0)
    P=Σa_save(i)X(n-i);
  else
    {solve a by LS optimization;
    P=Σa(i)X(n-i);
    a_save(i)=a(i) (i=1~N);
    }
  e(n)=X(n)-P;
  if(|e(n)|<T)
    flag=0;
  else
    flag=1;
}
  
```

Fig. 5. Pseudo-C implementation of switching strategy in the 1-D scenario.

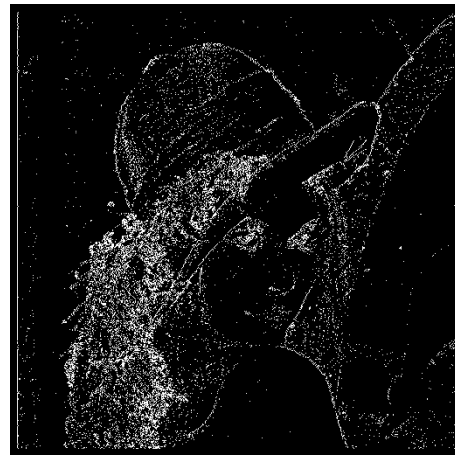


Fig. 6. Pixels in Lennagrey image where LS optimization is performed (the total number is 25 870).

the performance achieved by EDP for lossless image compression. To have a fair comparison, we download eight gray-scale (8-bit) images from Bernie's TMW0.51 home page [12] as our test set. Their entropy values are relatively large because they all contain plenty of edges. Such choice of the test set makes it easier to observe the gain brought by LS-based adaptation around edges.

To compare the performance of the different adaptive prediction schemes, we use the first-order entropy of the prediction residue image as the objective measure. EDP is compared with two of the very best context-based adaptive prediction schemes: MED [2] and GAP [3]. Since the performance of EDP depends on the prediction order  $N$ , we change it from low (4) to high (10) to observe the evolution of the entropy values. For any given  $N$ , the size of the training window  $T$  is chosen to be  $\min[N, 7]$ . From Table I, we can make the following conclusions: 1) the gain of EDP over MED and GAP quickly saturates as the prediction order  $N$  increases. For most images, a sixth-order predictor almost exploits all the gain offered by LS-based adaptation and 2) the gain is image-dependent. For the images containing abundant strong edges such as "barb," EDP significantly outperforms MED and GAP.

We have also developed a complete lossless image coder and compared it with the current state-of-the-art coders such as

TABLE I  
FIRST-ORDER ENTROPY (BPP) COMPARISON AMONG MED, GAP AND EDP

image	MED	GAP	EDP			
			N=4	N=6	N=8	N=10
baboon	6.28	6.22	6.04	6.01	6.00	5.99
lena	4.90	4.75	4.64	4.60	4.59	4.58
lennagrey	4.56	4.40	4.32	4.26	4.24	4.22
peppers	4.95	4.78	4.55	4.52	4.51	4.50
barb	5.21	5.15	4.67	4.44	4.40	4.35
barb2	5.19	5.06	4.93	4.80	4.79	4.78
boats	4.31	4.29	4.20	4.14	4.12	4.10
goldhill	4.72	4.70	4.64	4.60	4.59	4.58
average	5.02	4.92	4.75	4.67	4.66	4.64

TABLE II  
COMPARISON PERFORMANCE (BPP) COMPARISON AMONG CALIC, TMW, AND EDP. THE LAST COLUMN INCLUDES THE RUNNING TIME OF EDP ON A CELERON 500 M MACHINE

image	CALIC	TMW	EDP	seconds
baboon	5.88	5.73	5.81	14.11
lena	4.48	4.30	4.40	4.17
lennagrey	4.11	3.91	4.02	3.02
peppers	4.42	4.25	4.35	3.62
barb	4.32	4.08	4.11	6.21
barb2	4.53	4.38	4.52	9.83
boats	3.83	3.61	3.80	4.56
goldhill	4.39	4.27	4.39	7.14
average	4.50	4.32	4.42	-

CALIC [3] and TMW [12]. Our coder is built on a 6th order EDP and employs ad-hoc context modeling techniques for the prediction residues. The coding context is the quantized local variance in the prediction residue domain and we borrow similar bias cancellation techniques from CALIC [1]. As shown in Table II, we have achieved noticeably better performance than CALIC with moderately increased complexity. Though TMW still outperforms EDP, the complexity of our EDP coder is much lower than that of the TMW coder (seconds versus hours). Other researchers have shown that the gap between EDP and TMW can be further reduced by more sophisticated modeling and coding techniques [9].

#### IV. CONCLUSIONS

In this paper, we provide an interpretation of the LS-based adaptive prediction from the edge-orientation point of view. Its superior performance is attributed to the edge-directed property of the LS optimization. Based on a better understanding of LS-based adaptation, we propose a novel approach of reducing its computational complexity. The LS optimization is performed only for a fraction of pixels in the image. The performance and the complexity of our lossless image coder built upon the edge-directed prediction lie somewhere between those of CALIC and TMW. The edge-directed property of the LS optimization has also found other important applications in other image processing tasks such as error concealment [13].

#### REFERENCES

[1] N. Memon and X. Wu, "Recent developments in context-based predictive techniques for lossless image compression," *Comput. J.*, vol. 40, no. 2/3, pp. 127–136, 1997.

[2] M. Weinberger, G. Seroussi, and G. Shapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, pp. 1309–1324, Aug. 2000.

[3] X. Wu and N. Memon, "Context-based adaptive lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

[4] W. Pennebaker and J. Mitchell, *JPEG: Still Image Data Compression Standard*. London, U.K.: Chapman & Hall, 1992.

[5] N. Kuroki *et al.*, "Lossless image compression by two-dimensional linear prediction with variable coefficients," *IEICE Trans. Fund.*, vol. E75-A, no. 7, pp. 882–889, 1992.

[6] X. Wu and K. Barthel, "Pieewise 2D autoregression for predictive image coding," in *Proc. Int. Conf. Image Processing*, vol. 3, Oct. 1998, pp. 901–905.

[7] M. Kwon *et al.*, "Lossless image coder with context-based minimizing MSE prediction and entropy coding," in *Proc. Int. Symp. Circuits Systems*, vol. 4, 1999, pp. 479–482.

[8] G. Motta *et al.*, "Adaptive linear prediction lossless image coding," in *Proc. Data Compression Conf.*, Mar. 1999, pp. 491–500.

[9] H. Ye, G. Deng, and J. Devlin, "Least squares approach for lossless image coding," in *Proc. 5th Int. Symp. Signal Processing Applications*, vol. 1, 1999, pp. 63–66.

[10] B. Aiazzi *et al.*, "Lossless image compression based on an enhanced fuzzy regression prediction," in *Proc. Int. Conf. Image Processing*, vol. 1, Oct. 1999, pp. 435–439.

[11] X. Li and M. Orchard, "Edge directed prediction for lossless compression of natural images," in *Proc. Int. Conf. Image Processing*, vol. 4, Oct. 1999, pp. 58–62.

[12] B. Meyer and P. Tischer, "TMW—A new method for lossless image compression," in *Proc. Picture Coding Symp.*, Oct. 1997.

[13] X. Li and M. Orchard, "Novel sequential error concealment techniques using orientation adaptive interpolation," in *Proc. Visual Communications Image Processing*, Jan. 2001, pp. 30–40.

[14] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.



**Xin Li** (S'97–M'00) received the B.S. degree with highest honors in electronic engineering and information science from University of Science and Technology of China, Hefei, in 1996, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 2000.

He has been a Member of Technical Staff with Sharp Laboratories of America, Camas, WA, since August 2000. His research interests include image/video coding and processing.

Dr. Li received the Best Student Paper Award at the Conference of Visual Communications and Image Processing, San Jose, CA, in January 2001.



**Michael T. Orchard** (F'00) was born in Shanghai, China. He received the B.S. and M.S. degrees in electrical engineering from San Diego State University, San Diego, CA, in 1980 and 1986, respectively, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1988 and 1990, respectively.

He was with the Government Products Division, Scientific Atlanta, from 1982 to 1986, developing passive sonar DSP applications, and has consulted with the Visual Communication Department, AT&T Bell Laboratories, since 1988. From 1990 to 1995, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, where he served as Associate Director of the Image Laboratory, Beckman Institute. Since 1995, he has been an Associate Professor with the Department of Electrical Engineering, Princeton University. During the Spring of 2000, he served as Texas Instruments Visiting Professor at Rice University, Houston, TX.

Dr. Orchard received the National Science Foundation Young Investigator Award in 1993, the Army Research Office Young Investigator Award in 1996, and was elected IEEE Fellow in 2000 for "contribution to the theory and development of image and video compression algorithms."