Edge-Oriented Uniform Intra Prediction

Dong Liu, Xiaoyan Sun, *Member, IEEE*, Feng Wu, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—We propose an intra prediction solution to block-based image compression. In order to adapt to local image features during intra prediction, we consider the distinct image singularities within the model of piece-wise smooth functions. With such singularities, i.e., edges in this paper, intra prediction can be performed by solving Laplace equations. Moreover, since edges exhibit spatial correlations, we design a rate-distortion optimized method for edge extraction and edge coding. Our edge-oriented intra prediction thus consists of the prediction of smooth regions as well as the prediction of edges. We compare our intra prediction with that in H.264 and achieve superior performance. Our intra prediction can also be integrated into a block-based image coding scheme, which is comparable to JPEG2000 in terms of objective quality. An important advantage of our intra prediction is the improvement in visual quality at low bit-rate due to the preservation of edges.**

***Index Terms*—Edge extraction, image compression, inpainting, intra prediction, Laplace equation.**

I. Introduction

INTRA prediction is an important technique in image and video compression to exploit spatial correlation within one picture. The idea of intra prediction can be traced back to the well-known DPCM coding [1] and the DC prediction in JPEG [2]. But it was formally named recently, accompanied with the state-of-the-art video coding standard MPEG-4 AVC/H.264 [3]. To be specific, intra prediction in H.264 is the spatial prediction of pixel values, which can be adopted in both intra and inter picture coding as the counterpart of temporal prediction - motion compensation. Within the block-based architecture, encoder and decoder can both estimate the current block based on spatially neighboring reconstructed blocks. Therefore, only the differences between estimated and real pixel values need to be transmitted, known as residues. Residues are then transformed, quantized, and entropy coded to form the final bit-stream.

Since pictures exhibit locally variant features, it is difficult to predict one block from its spatial neighbors. Intra prediction should adapt to the local features to improve the prediction accuracy. In H.264, several prediction modes are designed

Manuscript received February 1, 2008; revised July 1, 2008. Current version published September 10, 2008. This work was performed during D. Liu's internship at Microsoft Research Asia. Part of this work was presented at the 2008 Data Compression Conference (DCC'08), Snowbird, UT. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Amy R. Reibman.

D. Liu is with the University of Science and Technology of China, Hefei 230027, China (e-mail: liud@mail.ustc.edu.cn).

X. Sun, F. Wu, and Y.-Q. Zhang are with Microsoft Research Asia, Beijing 100190, China (e-mail: xysun@microsoft.com; fengwu@microsoft.com; yzhang@microsoft.com).

Digital Object Identifier 10.1109/TIP.2008.2002835

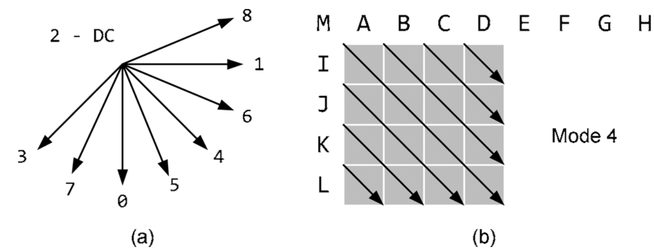


Fig. 1. Intra prediction modes for 4 × 4 blocks in H.264.

for this adaptation. As shown in Fig. 1(a), there are eight directional modes plus the DC mode for one 4 × 4 block. An exemplar directional mode is shown in Fig. 1(b). Boundary pixel values marked by A-M are copied and pasted into the 16 predicted pixels along a specified direction indicated by the mode, followed by a low-pass filtering (not shown in the figure). Intuitively, each directional mode can deal with blocks that display the same dominant direction in their pixel values. Thus, for each block, encoder can choose one mode that matches its directional feature, and signals the choice to decoder. In practice, rate-distortion optimization is widely adopted to select the mode.

It is arguable whether the directional modes in H.264 are efficient enough for accurate prediction. Some other intra prediction methods have been proposed in the literature. For example, sometimes the image blocks feature no direction but rather have repeated patterns. Running motion estimation and compensation within one picture can handle such blocks as proposed in [4]. In that case, the auxiliary information for intra prediction is the motion vectors instead of directional modes. Another example is template matching for intra prediction, which is inspired by the texture synthesis works and thus is able to handle regular textures [5].

This paper proposes an intra prediction method that is uniform since the prediction strategy is unique instead of different modes, and adapts to local image features. Our considered image feature is the singularities, also known as edges or boundaries. The underlying assumption is that images can be described by piece-wise smooth functions, thus, the prediction consists of two parts. The first is how to predict each smooth piece, and the second is how to predict the singularities. We use the well-known Laplace equation to predict the smooth image regions, while we design an edge estimation method to efficiently predict and code the edges.

Compared to the intra prediction in H.264, we utilize edges instead of the directional modes. Since edges represent distinct image structures, they also often correspond to the local directional features within small blocks. But several differences are noticeable. First, directional features do not always correspond to edges, i.e., they do not assure the correct descriptions of image singularities, whereas the latter is of great interest in

1057-7149/$25.00 © 2008 IEEE

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

our work. Second, edges can be in arbitrary shapes, and can be extracted during the compression, which makes edges more flexible compared to the predefined several modes. Third, we further consider the efficient coding of edges by means of edge prediction; in other words, the auxiliary information for prediction is also predictable; though in H.264 the modes are also predictively coded, they lack the immediacy as that of edges.

Indeed, the utilization of edges is widely accepted in image compression. First proposed by Kunt *et al.* [23], decomposing image data into contours and textures has been a common strategy. For example, sketch-based image coding [24] extracts both geometry and intensity information of edges, based on which it interpolates a so-called sketch image, and then codes the difference between the original and sketch images. An extension of this technique is reported in [25] and [26], where a three-component image model is proposed that splits the image into edge, texture, and smooth components. All these works are inspired by the human vision system (HVS) characteristics and target at the improvement of perceptual quality. Recent work that is most similar to ours is presented in [27], which also integrates edges into intra prediction. Note that in our method we adopt Laplace equations as the basic tool of intra prediction, and we allow edges to be in arbitrary shapes instead of only straight lines. Accordingly, the edge coding becomes quite different from that in [27].

We would also like to point out the relations between our proposed intra prediction and the image inpainting problem. Generally speaking, inpainting fills-in a specified image region based on the information provided by its spatial neighbors. Therefore, the existing inpainting methods are ready to be utilized by intra prediction. A great category of inpainting works has been constructed on the piece-wise smooth image functions. As shown by impressive results, high-order partial differential equations (PDEs) can predict smooth regions as well as some special singularities, such as straight line edges or parabolas [6]–[8]. In our work, we consider the singularities independently from the smooth regions. As a result, a simple PDE, the Laplace equation is adopted as the prediction method for pixel values. Since we also consider the prediction of edges, our approach is more general and capable in handling irregular singularities efficiently.

Our presented intra prediction method will be detailed in Sections II and III. Section II shows the prediction of smooth regions given the extracted edges. Section III discusses how the edges can be predicted and coded accordingly. Section IV presents some experimental results. Section V concludes this paper and outlines intended future work.

II. Intra Prediction With Edges

In this section, we consider the intra prediction problem with the assumption that distinct singularities have been extracted by any means. Commonly, we represent such singularities as edges with one-pixel width, which divide the image into smooth pieces.

Some notations are illustrated in Fig. 2. Let I be the image definition domain, which is divided into non-overlapped blocks. When coding the current block, its spatial neighbors above it and to its left have been reconstructed at both encoder and decoder (colored white and light-gray). Our proposed intra prediction

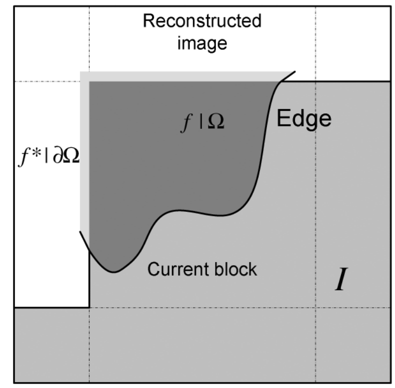


Fig. 2. Notations for intra prediction with edges.

works as follows. The current block is divided by edges into several regions, which are processed one by one, in either sequential or parallel manner. For each region Ω (colored dark-gray), its boundaries are checked, and those located in reconstructed image are marked as $\partial\Omega$ (colored light-gray). If $\partial\Omega$ is empty, the region will be filled-in with a default color (e.g., gray-level 128); otherwise, the pixel values in this region, i.e., f defined on Ω in the figure, will be generated by a Laplace equation according to f^* defined on $\partial\Omega$. At last, the pixels on the edges are generated by a Laplace equation according to the already generated smooth regions.

The continuous form of the Laplace equation reads

$$\frac{\partial f}{\partial t} = \Delta f \tag{1}$$

where $\Delta = (\partial^2)/(\partial x^2) + (\partial^2)/(\partial y^2)$ is known as the Laplacian operator. Given a specific initial state, e.g.,$f^{(0)}$, the Laplace equation evolves in time with generating a series of $f^{(t)}$ for $t > 0$. As shown in [9], in continuous case the solver is essentially convolving the initial state with a Gaussian kernel with variance t

$$f^{(t)} = f^{(0)} \circledast G_t. \tag{2}$$

This relation reveals the isotropic nature of the Laplace equation, which also demonstrates that the Laplace equation is capable in generating smooth regions. In practice, the PDE form (1) is preferable than the filtering form (2) due to the automatic decision of t. Specifically, the discrete form of (1) is

$$f^{(t+1)} = f^{(t)} + c^{(t)}(\Delta f)^{(t)}, t = 0, 1, 2, \ldots \tag{3}$$

Until convergence, i.e., $\|f^{(\tau+1)} - f^{(\tau)}\|$ less than a threshold, the state $f^{(\tau+1)}$ is regarded as the solution. Note that $c^{(t)}$ as the step size can vary in iterations but should only ensure the convergence.

Now turn to the realization. Due to our settings (cf. Fig. 2), a Laplace equation for the region Ω may not have proper boundary conditions. On the one hand, the pixels on the edge are not available yet, indeed, they will be generated after all smooth regions. On the other, if the boundary is located in not-coded blocks, i.e., the blocks to the right or below the current, the boundary is not available either. As mentioned above, only the available

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

boundaries are marked as $\partial\Omega$, then f^* defined on $\partial\Omega$ is posed as Dirichlet boundary condition. Moreover, the Laplacian is redefined to exclude the unavailable boundaries. To that end, we need an indicator function that evaluates 1 for available pixels and 0 for unavailable ones. Such an indicator can be defined as

$$\chi(x,y) = \begin{cases} 1, & \text{if } (x,y) \in \Omega \cup \partial\Omega \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Then, the Laplacian can be estimated by

$$\mathcal{L}f(x,y) = \sum_{(i,j)\in\mu_4(x,y)} \chi(i,j)(f(i,j) - f(x,y)) \quad (5)$$

where $\mu_4(x,y)$ involves the four neighboring pixels of (x,y). In this manner, we solve the pixel values f only based on the available boundaries f^*. Note that the above discussion is for the smooth regions. As for the pixels on the edges, also a Laplace equation is solved, but the already generated smooth regions are regarded as available, and posed as Dirichlet boundary condition.

By setting the step size to

$$c^{(t)}(x,y) = c(x,y) = (\sum_{(i,j)\in\mu_4(x,y)} \chi(i,j))^{-1} \quad (6)$$

we get the final evolution

$$f^{(t+1)}(x,y) = c(x,y) \sum_{(i,j)\in\mu_4(x,y)} \chi(i,j) f^{(t)}(i,j) \quad (7)$$

also known as the Jacobi iteration. If the pixels in one region are processed in scan-line order, the former pixels can be updated, i.e., $f^{(t)}(x-1,y)$ and $f^{(t)}(x,y-1)$ can be replaced by $f^{(t+1)}(x-1,y)$ and $f^{(t+1)}(x,y-1)$, respectively, so as to accelerate the convergence. This is known as the Gauss-Seidel iteration.

The Laplace equation (1) has been investigated for a long time since it relates to the heat conduction or diffusion problem. It also finds physical or geometrical interpretations in many computer vision or image processing scenarios. A lot of anisotropic variants of the Laplace equation can be found in [9]. As for intra prediction, the application is not found yet, to our best knowledge. Since the Laplace equation cannot predict any singularities, in the context of image inpainting, many PDEs have been designed to replace (1), which are often of high-order and time-consuming [6]–[8]. However, in our scenario, the singularities have been extracted and represented independently, thus, the Laplace equation works properly for the prediction of smooth regions.

Another noticeable issue is that neither the Laplace equation (1) nor advanced PDEs can predict the textural oscillations in images. In this sense, our proposed intra prediction can be a complement to the methods proposed in [4] and [5], which mainly target the prediction of textures.

III. Edge Extraction and Coding

With the available prediction method for smooth regions, the problem now is how to extract and record image singularities.

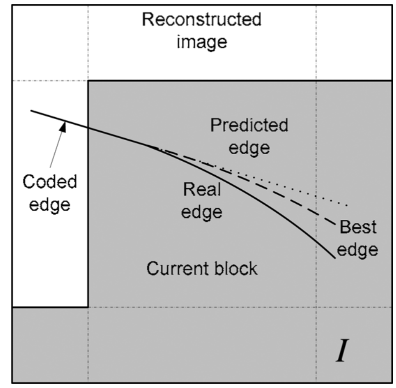


Fig. 3. Example shows that edge extraction and coding can be jointly considered to achieve the rate-distortion optimization.

Within the image model of piece-wise smooth functions, the singularities are generally understood as edges or boundaries [10]. There has been excellent research on the extraction, representation, coding, or other manipulations of edges. For example, many mature tools are provided for edge detection [11]. As for coding, chain codes are classic [12], whereas binary map coding is also widely adopted [13], [14]. It is possible to directly utilize these methods in our proposed intra prediction. However, from the compression point of view, such separate processes, i.e., edge extraction followed by coding, are not efficient in terms of coding.

Since edges act as the auxiliary information for the intra prediction in our approach, the correlation between edges can be exploited from the perspective of prediction accuracy. Fig. 3 depicts our observation. According to edge detection, the real edge is shown as the solid line, part of which has been coded with the previous blocks. Generally, to code this edge requires the analysis of its geometrical properties. Due to the spatial continuity of the edge, encoder and decoder can each predict the current edge piece based on the coded piece. An exemplar predicted edge is shown as the dot line. Coding the difference between the predicted and the real edges often costs fewer bits.

But the story goes further. If we directly use the predicted edge in the intra prediction, it virtually costs no bits for the edge, while the prediction accuracy would likely decrease, residues cost more bits or incur more distortion. In terms of the minimum rate-distortion cost that involves both edges and residues, the *best* edge may turn out to be neither the real nor the predicted, as the dash line depicts in Fig. 3. This example shows that the prediction efficiency judges the quality of edges, which is the difference from previous edge coding work.

Moreover, two other examples are illustrated in Fig. 4. In (a), there is one real edge crossing the corner of the current block. Due to the edge, Laplace equation is not applicable to the bottom-right region (colored dark-gray). Such edge can be erased to let the block be entirely generated by the Laplace equation, which may provide an even better prediction. For another example, even there is no real edge in (b), we can still impose an imaginary edge and divide the block into different regions, so as to generate them independently. Actually, the cases shown in Fig. 4 can be viewed as remedies for the edge detection, since

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

TABLE I
EDGE PREDICTION ALGORITHM

Step 1. **Identify the case.** Find the coded edges that stop at the boundaries of the current block. If no such edge exists, go to Step 3.

Step 2. **Case 1: Elongate a coded edge.** If more than one coded edges were found, choose the one with the maximum average gradients. Elongate the chosen edge in straight line. Halt.

Step 3. **Case 2: Imagine an edge.**
3.1 From the boundaries of the current block, choose one pixel that has the maximum gradient;
3.2 From the eight neighbors of the pixel found in 3.1, choose the one that has the minimum difference from it;
3.3 Draw a straight line across the two pixels found in 3.1 and 3.2, respectively. Halt.

TABLE II
FOUR MODES IN EDGE ESTIMATION

Mode index	Description	Remark
0	Use NONE edge.	
1	Use REAL edge. Code REAL edge.	Edge (a binary map) coded by JBIG. [a]
2	Use PREDICTED edge.	
3	Use REAL edge. Code the difference between REAL and PREDICTED edges.	Difference (also a binary map) coded by JBIG. [b]

[a] For technical details of JBIG, we refer to [13]. Our adopted realization is that in [28].

[b] Both real and predicted edges for one block are regarded as binary maps, thus, their difference is equivalent to their pixel-wise exclusive-OR (XOR), then also a binary map. At decoder, after the generation of predicted edge and decoding of the difference, the real edge is the pixel-wise XOR of predicted edge and difference.

TABLE III
TEST CONDITIONS FOR COMPARISONS WITH H.264 INTRA PREDICTION (RESULTS IN TABLE IV)

Item	Comparison 1	Comparison 2	Comparison 3
Coding methods	16×16	16×16	4×4 and 16×16
Prediction methods	Only ours *vs.* Only H.264	Both ours and H.264 *vs.* Only H.264	Both ours and H.264 *vs.* Only H.264
Mode selection	Rate-distortion optimization (RDO) ON		
Quantization	Fixed quantization parameter (QP) at four levels: 24, 30, 36, and 42		
Entropy coding	Variable length coding (VLC)		

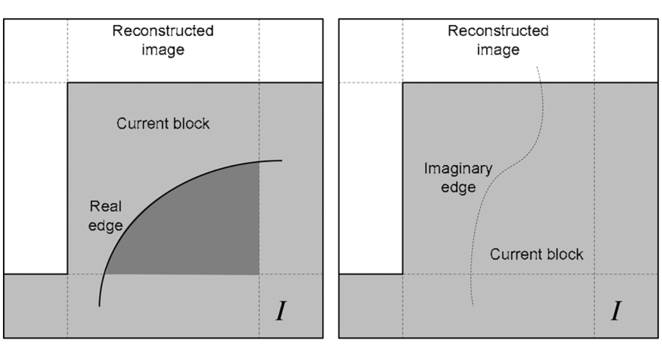


Fig. 4. Two examples show that edges can be erased or added to provide better predictions.

edge detection tools are difficult to tune and not robust for various images.

We can draw an analogy between the edges for our proposed intra prediction and the motion vectors for temporal prediction in video coding. The analogy inspires us to design a rate-distortion optimized algorithm for simultaneous edge extraction and edge coding, termed *edge estimation*. As shown in Fig. 3 by the dash line, the best edge is implicit information that can be different from the real edge or the predicted one. However, in practice, such a best edge is difficult to demonstrate. We can enumerate all possible edge configurations but the computational complexity may not be affordable. Thus, we have restricted the edge for prediction to be either the real one, or the predicted one, or none. Besides, the prediction of edges can also be diverse, whereas we only consider the simplest case—prediction of straight lines.

The edge estimation algorithm works as follows. Before the image is coded block by block, an edge detector is adopted on

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

TABLE IV
COMPARISON RESULTS WITH H.264 INTRA PREDICTION (TEST CONDITIONS IN TABLE III): SELECTED PERCENTAGE OF OUR INTRA PREDICTION (QP IS 42), AND AVERAGE PSNR IMPROVEMENT (QP FROM 24 TO 42)

Image	Size	Comparison 1		Comparison 2		Comparison 3	
		Mode selected	PSNR gain (dB)	Mode selected	PSNR gain (dB)	Mode selected	PSNR gain (dB)
Cameraman	256×256	100% (forced)	0.307	44.9%	0.504	12.9%	-0.007
Foreman	352×288		1.192	57.8%	1.478	34.6%	0.161
Barbara	512×512		0.171	54.6%	0.341	28.8%	0.038
Boat			0.058	49.5%	0.236	27.3%	0.023
Jet			0.011	43.8%	0.451	21.6%	0.036
Lena			0.090	46.1%	0.393	28.0%	0.044
Mandrill			0.098	64.6%	0.188	28.0%	0.020
Milk			-0.307	28.1%	0.458	20.1%	0.013
Peppers			0.193	49.0%	0.420	26.7%	0.048
Monarch	768×512 or 512×768		0.844	52.6%	1.055	30.4%	0.074
Kodim03			0.426	41.8%	0.734	27.0%	0.099
Kodim07			0.068	46.0%	0.536	19.1%	0.043
Kodim10			0.143	41.8%	0.444	25.6%	0.029
Kodim12			-0.085	33.7%	0.293	23.4%	0.065
Kodim15			-0.113	37.6%	0.384	25.7%	0.021
Kodim17			0.227	50.3%	0.435	31.3%	0.055
Kodim20			0.077	29.4%	0.465	14.8%	0.001
All 24 images in the Kodak image library (768×512 or 512×768)			0.036	45.1%	0.321	25.8%	0.033

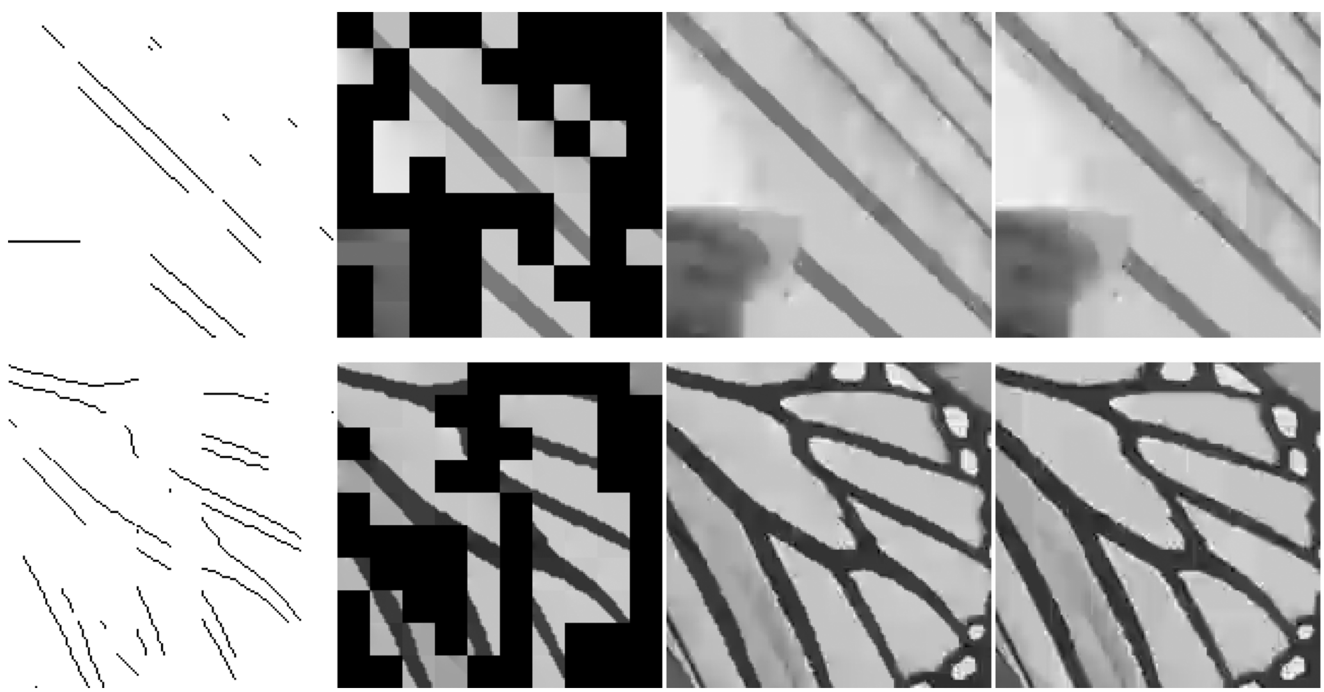


Fig. 5. Visual quality comparisons between our intra prediction and the directional one in H.264. The top row shows Foreman (partial) and the bottom row shows Monarch (partial). Results are gotten at QP equal to 42; test conditions are shown in Table III for Comparison 3. From left to right: The coded edges; our intra prediction results (only selected blocks are shown whereas others are blacked out); reconstructed image with our intra prediction; and reconstructed image without our intra prediction. Note the visual quality improvement of the selected blocks in the reconstructed images.

the entire image [15], followed by an edge thinning process that condenses the edge pixels into simple curves with one-pixel width [16]. For each block, encoder and decoder will predict an edge according to the edge prediction method, which is listed in Table I. After that, encoder will try the four modes listed in Table II; choose the mode with the minimum rate-distortion cost

$$j_i = D_i + \lambda R_i, \quad i = 0, 1, 2, 3 \tag{8}$$

where i indicates the mode. D_i is the coding distortion, measured by sum of squared differences (SSD) between the reconstructed and the original image block, R_i is the coding bits for edges and for residues, and λ is a Lagrange multiplier. In practice, λ is set according to quantization step Q [29]

$$\lambda = cQ^2 \tag{9}$$

where c is a constant. The chosen mode will be signaled to decoder. When mode index equals 1 or 3, the JBIG codes are also included in the bit-stream (cf. Table II). And residues are normally transformed and coded.

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

TABLE V
BIT-RATE PERCENTAGE OF DIFFERENT CODED ELEMENTS IN COMPARISON 1 (TEST CONDITIONS IN TABLE III)

Image	QP = 24 (high bit-rate)			QP = 42 (low bit-rate)		
	H.264	Ours		H.264	Ours	
	Mode + CBP [a]	CBP	Edge	Mode + CBP	CBP	Edge
Cameraman	2.06%	1.56%	2.56%	8.55%	7.58%	9.25%
Foreman	2.81%	2.22%	4.30%	12.67%	11.64%	13.51%
Lena	2.97%	2.18%	1.86%	13.85%	12.13%	10.12%
Milk	4.92%	3.47%	2.76%	21.65%	19.80%	18.06%
Monarch	2.74%	2.24%	3.10%	10.63%	10.08%	12.27%
Kodim03	3.45%	2.73%	3.28%	19.11%	17.47%	17.43%

[a] CBP stands for coded block pattern that indicates the positions of non-zero quantized coefficients. In H.264 16×16 block coding, prediction mode and CBP are combined and coded.

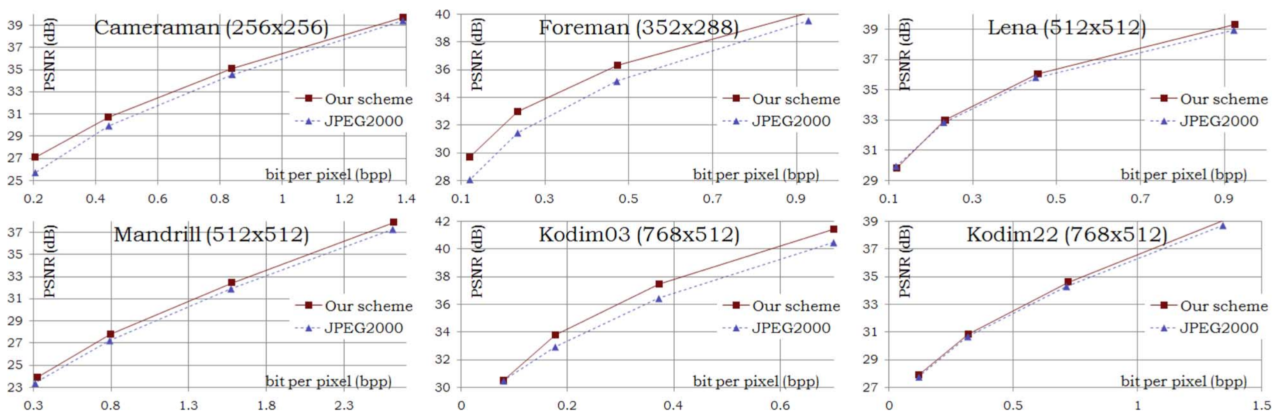


Fig. 6. Objective quality comparisons between our scheme and JPEG2000 in PSNR versus bit-rate curves.

IV. EXPERIMENTAL RESULTS

Our proposed intra prediction can be integrated into any existing block-based image or video compression systems. Its efficiency can be demonstrated by comparison with other intra predictions. Its potential can be further revealed by comparison with non-block-based compression schemes. We conduct both comparisons as follows. Our experiments concentrate on image compression, but the results have informative meanings for video compression, too. The test images are all gray-scale ones, coming from the standard library [17], as well as the Kodak image library [18].

A. Comparison With H.264 Intra Prediction

We integrate our intra prediction into the H.264 reference software [19] so as to compare our method with the directional prediction in H.264. As known, the basic coding unit in H.264 is a 16×16 macroblock in luminance (Y) component. One macroblock can be divided into 16 4×4 blocks, predicted and coded one by one, with nine modes as shown in Fig. 1. It can also be predicted as a whole with four modes, including vertical, horizontal, DC, and plane modes, which are quite similar to that for 4×4 blocks [3]. There are two other methods defined in the extension of H.264, but we omit them for brevity.

We realize our edge-oriented intra prediction only at 16×16 level. And we conduct three comparisons accordingly. The test conditions are summarized in Table III. Table IV lists typical results of the comparisons. Note that we provide average PSNR

Fig. 7. Reconstructed images by JPEG2000 (left) and our scheme (right). The top row shows Cameraman at 0.206 bpp; the bottom row shows Foreman at 0.12 bpp.

gain introduced by our intra prediction. The average gain is calculated over a wide bit-rate range (QP from 24 to 42) by the method proposed in [20]. We also provide the selected percentage of our intra prediction at low bit-rate (QP is 42) since the percentage is often more significant than that at high bit-rate.

It can be observed from Table IV that our intra prediction is comparable to the directional modes in H.264. The comparison

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

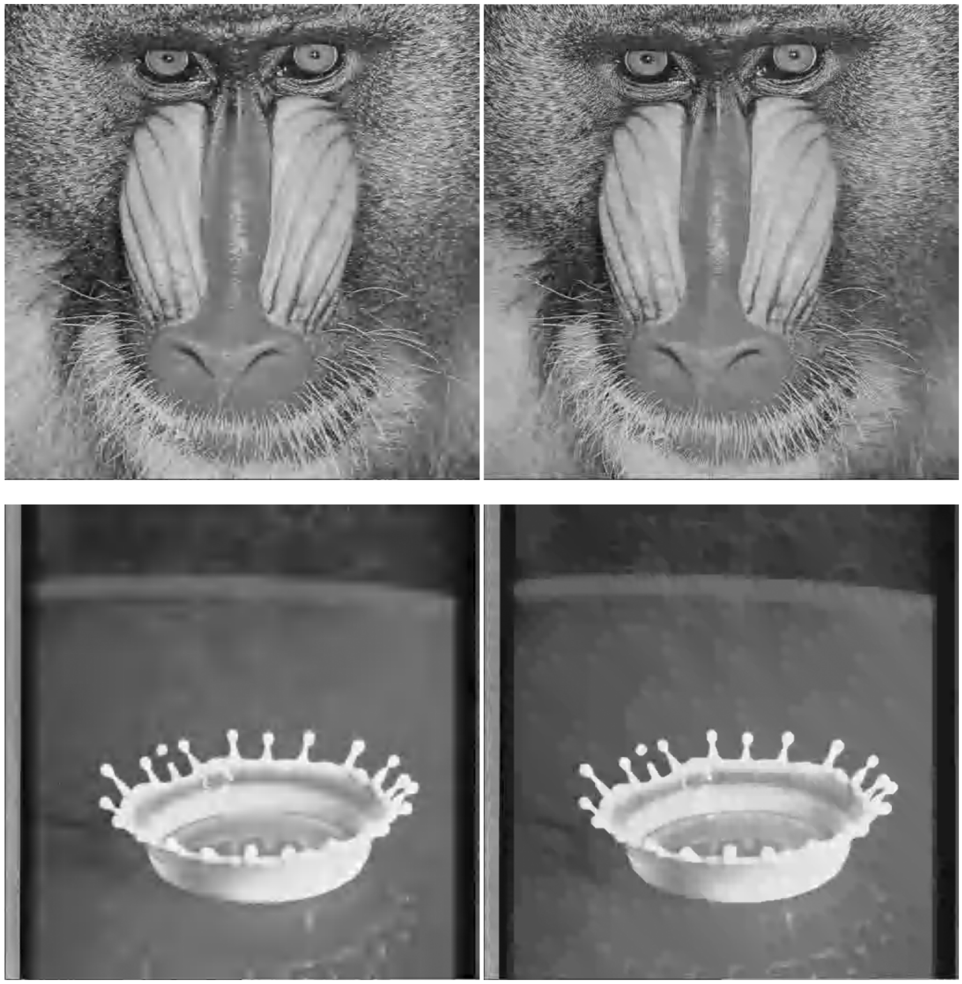


Fig. 8. Reconstructed images by JPEG2000 (left) and our scheme (right). The top row shows Mandrill at 0.32 bpp; the bottom row shows Milk at 0.067 bpp.

results depend on the image characteristics. Generally, for images that feature long and clean edges such as Foreman, our intra prediction has obvious gain, and vice versa. From Comparison 1 in Table IV, our intra prediction can be almost as good as the H.264 one for 16×16 block coding. When integrating both at 16×16 level (Comparison 2), our intra prediction can bring out as high as average 1.478 dB gain (Foreman, Comparison 2, in Table IV). When compared to both 4×4 and 16×16 coding (Comparison 3), the PSNR gain is not so obvious, but our intra prediction can still be selected quite often, more than 20% for most images at low bit-rate. Since we only integrate our intra prediction at 16×16 level, the results are reasonable: One macroblock can be divided into small 4×4 blocks, edges can be represented as a set of directional modes for small blocks. We may also integrate our intra prediction at 4×4 level, but then the bits for prediction modes will be excessive. How to efficiently integrate our method into H.264 remains an open issue.

Despite the PSNR comparisons, one important feature of our proposed intra prediction is the improvement in visual quality. It can be observed from Fig. 5 that the blocks selecting our intra prediction often show clear edges, whereas by the directional prediction in H.264, the edges have been distorted. As a result, when coding at low bit-rate, the blocks with our prediction contain less ringing artifacts or jaggies.

Table V lists the bit-rate percentage cost on different coded elements. We found that edges need more bits compared with the directional modes in H.264, and such overhead is obvious at low bit-rate. Therefore, a specially designed edge coding method instead of JBIG can hopefully improve the coding performance of the entire system.

B. Comparison With Non-block-Based Compression

We also realized a block-based image coding scheme to compare with the state-of-the-art image compression standard JPEG2000. Our scheme has the common architecture with 8×8 block size. Each block is predicted *only* by our proposed intra prediction method, and residues are transformed by 8×8 DCT. The coefficients are uniformly quantized and collected by an arithmetic encoder. We test this scheme in comparison with the JPEG2000 reference software [21]. Typical results are reported hereafter.

First, Fig. 6 presents the objective quality comparisons in PSNR versus bit-rate curves. In general, our scheme is comparable to JPEG2000 in terms of PSNR measurement. As high as 1.6 dB gain is achieved at low bit-rate for the Foreman image. For the Lena image, our scheme has a little loss at low bit-rate. Similar results have also been observed for the images with rich

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

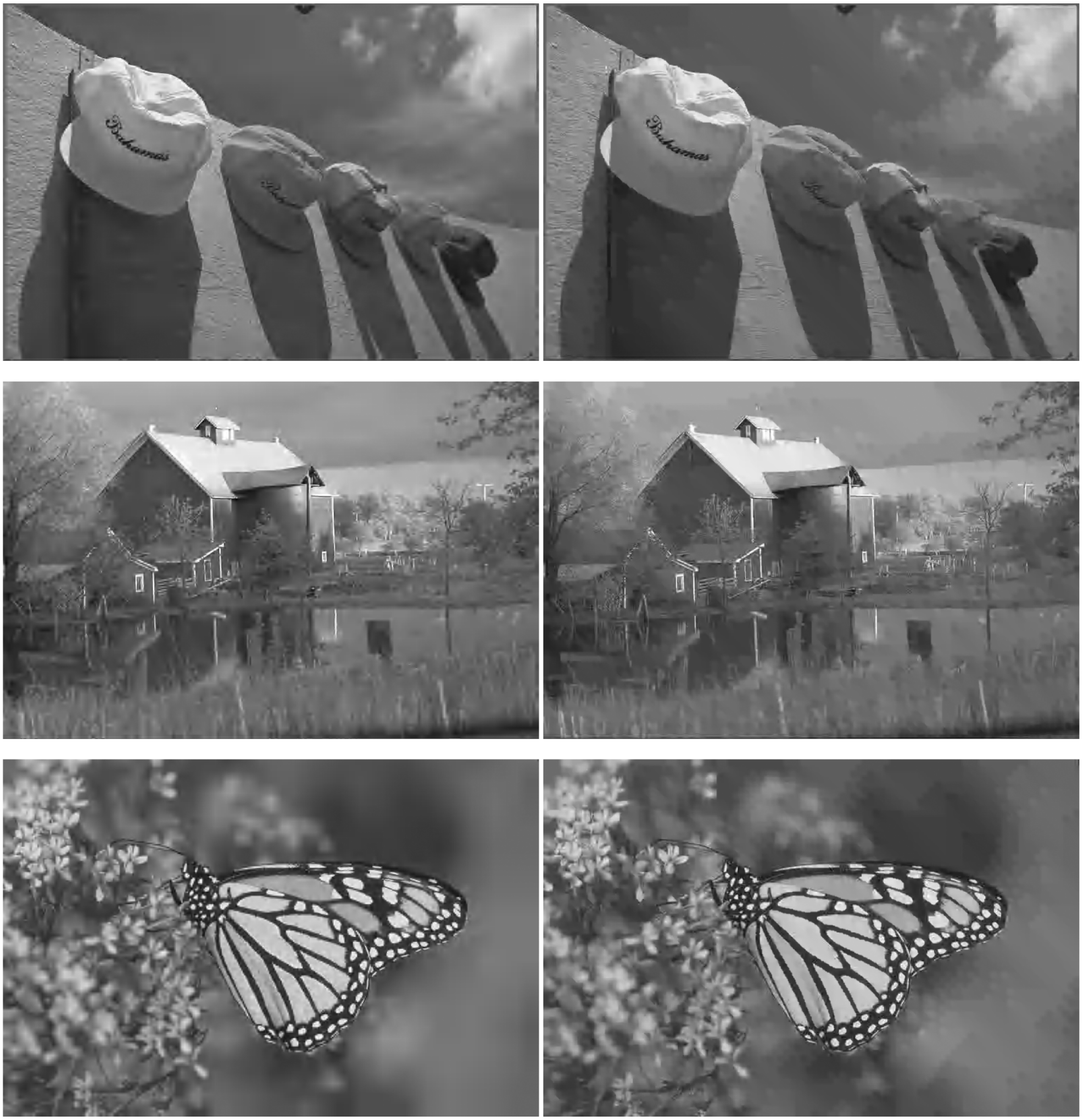


Fig. 9. Reconstructed images by JPEG2000 (left) and our scheme (right). From top to bottom: Kodim03 at 0.08 bpp; Kodim22 at 0.11 bpp; Monarch at 0.145 bpp.

textures. The reason is twofold. On the one hand, our intra prediction only tackles image singularities but is not able to predict textures efficiently. On the other, global wavelet transform in JPEG2000 is more suitable for textures than the block DCT in our scheme.

Second, Figs. 7–9 show some reconstructed images at low bit-rates with our scheme as well as JPEG2000. The JPEG2000 reconstructed images often suffer from ringing artifacts around edges at low bit-rate. As for ours, the ringing has been greatly eliminated and edges look "clean." Such results demonstrate the edge preserving nature of our proposed intra prediction, which can improve the visual quality since human vision tends to be more sensitive to edges. We would like to remind that textures are not well preserved by our scheme, and the blockiness inherent in block-based coding may be visible at low bit-rate.

C. Computational Complexity

Since intra prediction will be performed at decoder, its computational complexity is an important issue in practice. In our proposed method, the edge prediction step requires calculations of gradients as well as geometrical drawings. But we have observed that the most computations are cost on the solving of Laplace equations. Although the Laplace equation is much simpler than most PDEs, it still requires an iteration process until convergence. Note that mathematically, our evolution (7) can be substituted by the so-called successive over relaxation (SOR)

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

method for acceleration. Another possible realization is to solve the Laplace equation on GPU since techniques have been available [22]. Moreover, the Laplace equations for several smooth regions within one block can be solved in parallel manner since they are independent from one another. Such issues are worthy of further investigations.

V. Conclusion

We have proposed an intra prediction method in this paper in the context of block-based image compression. Our prediction assumes piece-wise smooth functions as image model, and identifies the image singularities to adapt to local image features. The prediction consists of two parts. First, smooth regions can be generated by solving Laplace equation, which is simple and computational efficient. Second, image singularities are represented as edges; we design an edge estimation algorithm for joint edge extraction and edge coding, so as to achieve the rate-distortion optimization.

As experimental results demonstrate, our intra prediction can be integrated into H.264 and superior performance has been achieved. Block-based image compression scheme with our intra prediction can be comparable to JPEG2000 in terms of objective quality. The preservation of edges leads to better visual quality at low bit-rate.

There are some possible improvements in the current approach. As mentioned, our edge prediction is simple for only straight lines; some refined methods can be integrated to deal with other edge shapes, such as parabolas. Edge coding is now realized by JBIG, whereas chain coding can be an alternative. The parametric description of edges, such as straight lines in [27], may be more efficient. Moreover, the PDEs for inpainting [6]–[8] may be tested in replacement of the Laplace equation. The key issue then is how to identify the singularities that cannot be predicted by inpainting, as well as how to represent such irregular singularities.

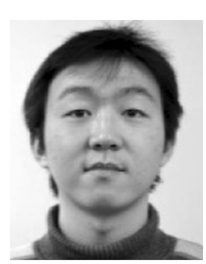
Due to the inherent assumption of piece-wise smooth image model, our intra prediction is not effective at predicting textures. We plan to investigate the texture synthesis methods as a complementary approach. We nonetheless believe that edges will play an important role for the prediction of complex image blocks.

Acknowledgment

The authors would like to thank two anonymous reviewers for their constructive opinions. They would especially like to thank D. Daniels for his editing.

References

[1] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366–406, Mar. 1980.
[2] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, Apr. 1991.
[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[4] S.-L. Yu and C. Chrysafis, "New intra prediction using intra-macroblock motion compensation," presented at the 3rd Meeting of Joint Video Team (JVT), Fairfax, VA, May 2002, JVT-C151, unpublished.
[5] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by template matching," in *Proc. Int. Conf. Image Process.*, Atlanta, GA, Oct. 2006, pp. 1693–1696.
[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, New Orleans, LA, Jul. 2000, pp. 417–424.
[7] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001.
[8] T. F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, Feb. 2002.
[9] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.
[11] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[12] H. Freeman, "Computer processing of line-drawing images," *ACM Comput. Surv.*, vol. 6, no. 1, pp. 57–97, Mar. 1974.
[13] "Information technology—Coded representation of picture and audio information—Progressive bi-level image compression," ISO/IEC 11544/ITU-T Rec. T.82., Mar. 1993.
[14] "Information technology—Lossy/lossless coding of bi-level images," ISO/IEC 14492/ITU-T Rec. T.88, Feb. 2000.
[15] C. A. Rothwell, J. L. Mundy, W. Hoffman, and V.-D. Nguyen, "Driving vision by topology," in *Proc. Int. Symp. Computer Vision*, Coral Gables, FL, Nov. 1995, pp. 395–400.
[16] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1273–1287, Oct. 2007.
[17] [Online]. Available: http://sipi.usc.edu/services/database/
[18] [Online]. Available: http://r0k.us/graphics/kodak/
[19] [Online]. Available: http://iphome.hhi.de/suehring/tml/, JM ver. 12.4.
[20] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," presented at the 13th Meeting of Video Coding Experts Group (VCEG), Austin, TX, Apr. 2001, VCEG-M33, unpublished.
[21] [Online]. Available: http://www.ece.uvic.ca/~mdadams/jasper/, JasPer ver. 1.900.1,
[22] J. Bolz, I. Farmer, E. Grinspun, and P. Schröder, "Sparse matrix solvers on the GPU: Conjugate gradients and multigrid," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 917–924, Jul. 2003.
[23] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE*, vol. 73, no. 4, pp. 549–574, Apr. 1985.
[24] S. Carlsson, "Sketch based coding of grey level images," *Signal Process.*, vol. 15, no. 1, pp. 57–83, Jul. 1988.
[25] X. Ran and N. Farvardin, "A perceptually motivated three-component image model—Part I: Description of the model," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 401–415, Apr. 1995.
[26] X. Ran and N. Farvardin, "A perceptually motivated three-component image model—Part II: Applications to image compression," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 430–447, Apr. 1995.
[27] C. Dai, O. D. Escoda, P. Yin, X. Li, and C. Gomila, "Geometry-adaptive block partitioning for intra prediction in image & video coding," in *Proc. 2007 Int. Conf. Image Processing*, San Antonio, TX, Sep. 2007, vol. VI, pp. 85–88.
[28] [Online]. Available: http://www.cl.cam.ac.uk/~mgk25/jbigkit/, JBIG-KIT ver. 1.6.
[29] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.



Dong Liu received the B.S. degree in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2004. He is currently pursuing the Ph.D. degree at USTC.

He has been a visiting student at Microsoft Research Asia since July 2005, where his research concentrates on image/video compression and compression-oriented vision technologies. He is also interested in image segmentation and image representation.

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.

Xiaoyan Sun (M'04) received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1997, 1999, and 2003, respectively.

She joined Microsoft Research Asia, Beijing, China, as an Associate Researcher in 2003 and has been a Researcher since 2006. She has authored or co-authored over 30 conference and journal papers and submitted several proposals and contributed techniques to MPEG-4 and H.264. Her research interests include video/image compression, video streaming, and multimedia processing.

Feng Wu (M'99–SM'06) received the B.S. degree in electrical engineering from the University of Xi'an Electrical Science and Technology, Xi'an, China, in 1992, and the M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1996 and in 1999, respectively.

He joined Microsoft Research Asia, Beijing, China, as an Associate Researcher in 1999 and was promoted to Lead Researcher in 2006. He has played a major role in Internet Media Group to develop scalable video coding and streaming technologies. He has authored or co-authored over 100 papers in video compression and contributed some technologies to MPEG-4 and H.264. His research interests include video and audio compression, multimedia transmission, and video segmentation.

Ya-Qin Zhang (S'87-M'90-SM'93-F'98) received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 1983 and 1985, and the Ph.D. degree from George Washington University, Washington, DC, in 1989, all in electrical engineering.

He is currently the Corporate Vice President of Microsoft, Beijing, China. He was the Managing Director, Microsoft Research Asia, Beijing, China, in 1999. Previously, he was the Director of the Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). Prior to that, he was with GTE Laboratories, Inc., Waltham, MA, from 1989 to 1994. He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He has authored and coauthored over 200 refereed papers in leading international conferences and journals, and has been granted over 40 U.S. patents in digital video, Internet, multimedia, wireless, and satellite communications. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards. He serves on the Board of Directors of five high-tech IT companies and has been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Zhang served as the Editor-In-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He was the Chairman of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems (CAS) Society. He serves on the Editorial Boards of seven other professional journals and over a dozen conference committees. He has received numerous awards, including several industry technical achievement awards and IEEE awards, such as the CAS Jubilee Golden Medal. He was named "Research Engineer of the Year" in 1998 by the Central Jersey Engineering Council for his "leadership and invention in communications technology, which has enabled dramatic advances in digital video compression and manipulation for broadcast and interactive television and networking applications." He received The Outstanding Young Electrical Engineer of 1998 award.

Authorized licensed use limited to: MICROSOFT. Downloaded on January 31, 2009 at 02:28 from IEEE Xplore. Restrictions apply.