Wavelet-Domain Reconstruction of Lost Blocks in Wireless Image Transmission and Packet-Switched Networks

Shantanu D. Rane, Jeremiah Remus, Guillermo Sapiro
Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455, USA
guille@ece.umn.edu.

Abstract—

A fast scheme for wavelet-domain interpolation of lost image blocks in wireless image transmission is presented in this paper. In the transmission of block-coded images, fading in wireless channels and congestion in packet-switched networks can cause entire blocks to be lost. Instead of using retransmission query protocols, we reconstruct the lost block in the wavelet-domain using the correlation between the lost block and its neighbors. The algorithm first uses simple thresholding to determine the presence or absence of edges in the lost block. This is followed by an interpolation scheme, designed to minimize the blockiness effect, while preserving the edges or texture in the interior of the block. The interpolation scheme minimizes the square of the error between the border coefficients of the lost block and those of its neighbors, at each transform scale. The performance of the algorithm on standard test images, its low computational overhead at the decoder, and its performance vis-a-vis other reconstruction schemes, is discussed.

I. INTRODUCTION

In common operation of still image compression standards like JPEG and JPEG2000 [1], the encoder tiles the image into blocks of $n \times n$ (n being a power of 2) pixels, calculates a 2-D transform, quantizes the transform coefficients and encodes them using Huffman or arithmetic coding. In common wireless scenarios, the image is transmitted over the wireless channel block by block. Due to severe fading, entire image blocks can be lost. In [2] the authors report that average packet loss rate in a wireless environment is 3.6% and occurs in a bursty fashion.

Error resilient channel coding schemes (e.g., Forward Error Correction) use Reed Solomon codes or convolutional codes to reconstruct the lost portion of the bitstream, sacrificing some useful bandwidth in the process. This method, which is designed for a fixed bit error rate (BER), cannot completely prevent loss of data when the BER is unknown, as in most practical cases.

The common techniques to recover the lost block are grouped under Automatic Retransmission Query Protocols (ARQ). As noted in [3], ARQ lowers data transmission rates and can further increase the network congestion which can aggravate the packet loss. Instead, we show that it is possible to satisfactorily reconstruct the lost blocks by using the available information surrounding them. [1] The basic idea is to first automatically classify the block with respect to the presence or absence of an edge, and then to interpolate the missing block from its 8-neighborhood. We test the proposed scheme with a variety of images and simulated block losses. We show that a reconstruction of acceptable visual quality and high PSNR is obtained at a considerably low computational cost.

II. PREVIOUS RELATED WORK

Purely decoder based error concealment in baseline JPEG coded images has been demonstrated in the image domain and in the DCT domain. Though wavelet-domain methods to reconstruct entire lost blocks have not yet been reported, various studies have successfully used the wavelet framework for texture synthesis [4], reconstruction of edges which are distorted during compression [5], and enhancement of edges which are blurred during interpolation [6].[2]

In [7] the authors provide a survey of commonly used error control and concealment methods in image transmission. Image domain methods use interpolation as in [8], or separate reconstruction methods for structure and texture as in [9]. Most transform based methods, notably those described for MPEG-2 video in [10] and earlier for DCT-JPEG images in [8], assume a smoothness constraint on the image intensity. These methods define an object function which measures the variation at the border between the lost block and its neighbors, and then proceed to minimize this object function. The work in [11] describes a different DCT based interpolation scheme which uses only 8 border pixels to reconstruct the 64 lost DCT coefficients. The idea of [8], which exploits interblock correlation and minimizes the squared error between the lost block and its neighbors, forms the starting point of the present work. The novelty of our method is that it first classifies the lost blocks as "edgy" and "non-edgy", and then selectively applies the linear least squares interpolation on various scales of the DWT. No side information about the image is assumed in this study.[3]

III. THE PROPOSED ALGORITHM

Once the missing block has been detected, the reconstruction of lost blocks includes the following steps: 1. Classify lost

[1] The location of lost data, i.e. lost image blocks, is assumed to be known at the stage of reconstruction.

[2] D. Donoho and R. DeVore are currently working on an interpolation scheme based on wavelets as well (C. Schwartz, personal communication).

[3] We assume that the DC value is available to us, when a lost block is to be reconstructed. If this assumption is dropped, the DC estimation method in [13] could be used.

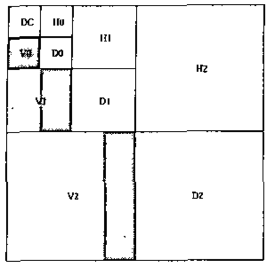


Fig. 1. Pyramid Ordering of Wavelet Coefficients. H = Horizontal, V = Vertical, D = Diagonal. Level 0 = singleton value, Level 1 = 2 × 2 array, Level 2 = 4 × 4 array. Shaded areas indicate high coefficients, caused by a vertical edge through the right half of the block

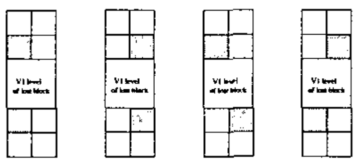


Fig. 2. Possible cases where vertical edge is detected. Shaded areas indicate coefficients greater than an empirically determined threshold T1.

blocks into "edgy" and "non-edgy," 2. Reconstruct edgy blocks from selected edgy blocks in the 8-neighborhood and non-edgy blocks from selected non-edgy blocks in the 8-neighborhood. We now proceed to describe each one of these components. In the following explanation and the results, we have used 8 × 8 tiles, though the method is easily extendable for 16 × 16 and larger tiles.

A. Block classification

The magnitude of a wavelet coefficient specifies both the amount of change, as well as the spatial location at which the said change occurs. For the wavelet decomposition of a 8 × 8 image block shown in Fig. 1, level 0 is too coarse, while level 2 coefficients represent very localized details and may even appear noisy. Hence we choose level 1 coefficients to determine the presence or absence of an edge. Upon comparing the magnitude of the level 1 coefficients with a threshold T1 (determined empirically after testing with a number of images), we can ascertain whether the edge, if present, is horizontal or vertical. A vertical edge can occur in one of the 4 different ways shown in Fig.2. Similar cases occur for horizontal edges. If the lost block fails to pass any of these 4 tests, the algorithm decides that it does not contain an edge. This is a simple and computationally efficient technique.

B. The reconstruction method

The main assumptions of the reconstruction algorithm are: 1. The type of detail (horizontal, or vertical) being reconstructed determines which of the neighboring blocks are used for reconstruction; 2. The above propagation of details does not cross an edge.

Differing from [8], where the authors always interpolate the lost DCT coefficients from the *entire* 4-neighborhood, our algorithm selects which blocks among the 8-neighborhood will be used for interpolation based on the classification of Section III-A. This reduces the computational cost and in general improves the reconstruction quality (see below).

B.1 Reconstruction of edgy blocks

To clarify the explanation, let us consider an example where a perfectly vertical edge is detected. This means that the vertical details in the top and bottom neighbors need to be used for interpolation. The horizontal and diagonal details are, by our assumption, negligible, and need not be interpolated. On each level, the matrix X of lost coefficients, is obtained by interpolation from the matrices T and B containing appropriately chosen coefficients from the top and bottom blocks respectively. The smoothness constraint is that we must minimize the square of the error at the top and bottom borders. For level $V0$, X,T,B consist of only one coefficient, and hence X is the average of T and B. For level $V1$, X,T,B are 2 × 2 matrices so that
$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, T = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$.
We find x_{11} and x_{12} while minimizing the squared error at the top border, i.e we solve $\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = m \begin{bmatrix} t_{11} \\ t_{12} \end{bmatrix} + n \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}$ while minimizing $\epsilon_t = \| X_t - T_b \|^2$ where $X_t = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$, and $T_b = \begin{bmatrix} t_{21} \\ t_{22} \end{bmatrix}$.
Then we find x_{21} and x_{22} while minimizing the squared error at the bottom border, i.e we solve $\begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} = m \begin{bmatrix} t_{21} \\ t_{22} \end{bmatrix} + n \begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix}$ while minimizing $\epsilon_b = \| X_b - B_t \|^2$ where $X_b = \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix}$, and $B_t = \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}$.
For 8 × 8 tiling, the finest level is $V2$, which is a 4 × 4 matrix. To reuse the above equations, consider the $V2$ levels of the missing block, and top and bottom blocks. Now split the rows of 4 coefficients into two groups of two coefficients each, as shown in Fig.3, and form the 2 × 2 matrices X,T,and B as above. To reconstruct the top and bottom row of $V2$ level, we will need to form X,T,and B matrices twice (i.e once for each group). These reconstructed outer rows, are used in the reconstruction of the middle two rows, by repeating the procedure as shown in the right half of Fig.3. Note that, in forming these groups of two coefficients, we choose to pair alternate coefficients in the left half of Fig.3, but successive coefficients in the right half. This is expected to spread out the interpolation error. An actual reconstruction example using this method is shown in Fig. 4.

B.2 Reconstruction of non-edgy blocks

In the simplest possible case we need to interpolate all types of details from all neighbors of the lost block. The method for interpolating horizontal and vertical details is exactly as explained above for edgy blocks, i.e horizontal (resp. vertical) details are interpolated from left and right (resp. top and bottom) neighbors.

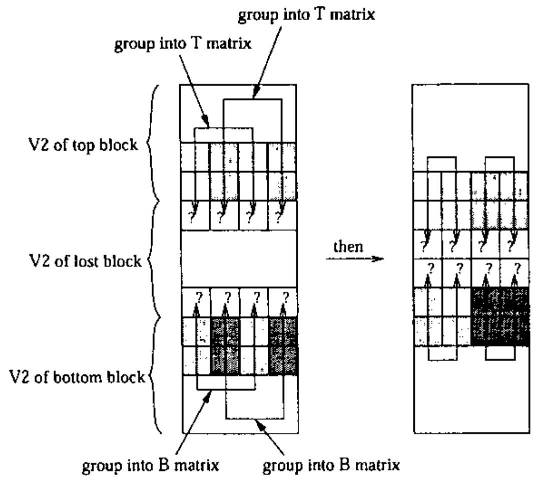
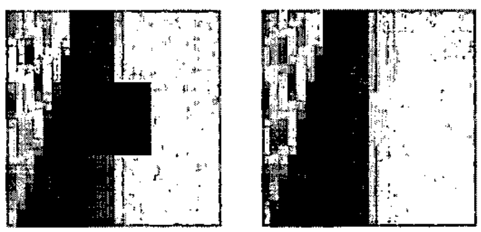


Fig. 3. Reconstruction of Vertical Level 2. Similarly shaded areas indicate coefficients grouped together to form matrices T and B. The question marks and arrow-ends indicate the corresponding missing elements in X. The equations of SectionIII-B.1 are solved 4 times to fill up the missing $V2$ level.



Received Reconstructed

Fig. 4. Reconstruction of a perfect vertical edge. This is a 24 × 24 tile of the Barbara image magnified 4 times.

In case an edge occurs in the neighborhood of the lost non-edgy block, then the block containing that edge is removed from the interpolation process to prevent the unpleasant edge migration effect.

If higher computational levels are accepted, non-edgy blocks could also be addressed using wavelet-based texture synthesis algorithms.

IV. Experimental results

Since we have no control over the fading channel, there is no prior information about the relative locations and number of blocks that can be lost in the process. We note that before transmission of the 8 × 8 blocks, a packetization scheme is applied so that a bursty packet loss during transmission is scattered into a pseudorandom loss in the image domain. Therefore, consecutive image blocks are rarely lost and the reconstruction scheme can use the neighborhood of the lost block for reconstruction. A sample packetization scheme may be found in [2]. Figs. 5 and 6 show the results obtained after reconstruction of 272 × 272 sub-images. The default irreversible Daubechies biorthogonal 9/7 filter, as specified in the JPEG2000 standard, has been used in all examples. In Fig. 5 we also see details of the image and the comparison of our algorithm with that of [8]. Our method shows a better continuation of the structures across the missing block.

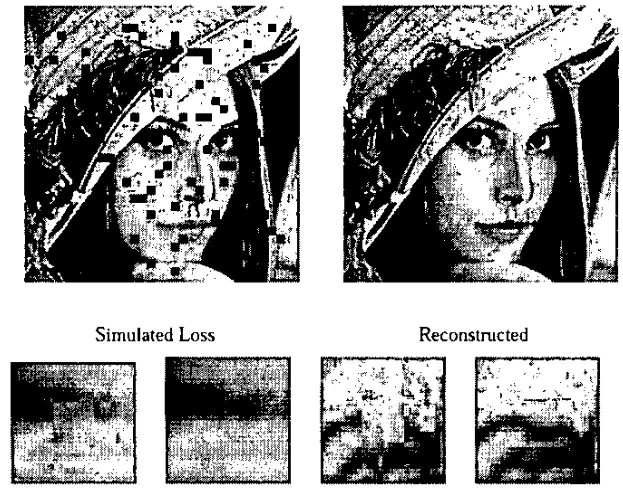
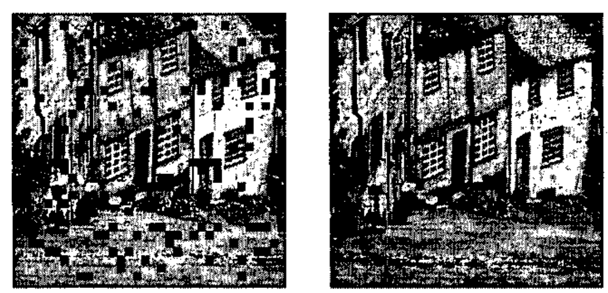


Fig. 5. Reconstruction of Lena with PSNR = 32.422 db. The bottom row shows details (hair and shoulder) of this image and the comparison of our technique (2nd and 4th) with the one by Hemami et al. (1st and 3rd).



Simulated Loss Reconstructed

Fig. 6. Reconstruction of Goldhill with PSNR = 31.684 db

Fig. 7 demonstrates the behavior of the PSNR with worsening losses and also shows how the execution time increases. We note that the PSNR remains satisfactory (above 30 db) even for losses upto 15% and the CPU time required to repair this loss is only about 80 ms.

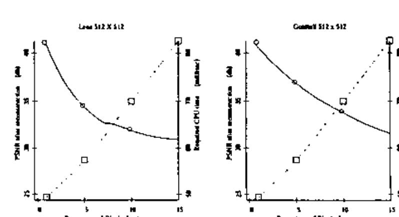


Fig. 7. PSNR (round dots) behavior for random block loss. Execution times (square dots) were measured on a Sun Ultra 10 Machine running SunOS 5.

V. Extension to Larger Tiles and Advanced Reconstruction

Our results here are restricted to 8×8 tiles where *all* the information has been lost. The new still-image compression standard, JPEG2000, successively applies the wavelet transform to decompose each $n \times n$ image tile into a pyramid of horizontal, vertical and diagonal details. (See [14], [15], [16] for implementation details). For improved compression and reduced blockiness, the wavelet transform is calculated for larger tile sizes (e.g 64×64 or more). The wavelet subbands are further divided into codeblocks, which are encoded independently. To reconstruct code-blocks lost in transmission, we can exploit correlation between adjacent codeblocks, instead of correlation between adjacent tiles. Hence, the lost codeblocks will be reconstructed, by the above method, from available neighboring codeblocks in the same subband. We are currently investigating the performance of the algorithm for large code-block sizes (e.g 32×32), which are typically used in JPEG2000.

VI. Conclusion

An algorithm for wavelet-domain reconstruction of lost blocks was presented. While preserving smoothness constraints as in [8] and [10], the algorithm uses the wavelet framework to provide a better reconstruction of edges in the interior of the lost block. The scheme selects the edge orientation, and applies a linear least squares problem to that orientation on all transform scales.

The interpolation is obtained fast, at a very low computational complexity. Indeed, the most complex part of the algorithm is the solution of $PY = Q$, where P is at most a 2×2 matrix.

The method fails to reconstruct image features which are completely obliterated during transmission, and, at present, is not uniformly satisfactory for all diagonal edges. Slanting edges which deviate slightly from the horizontal and vertical directions are reconstructed properly. But, the algorithm has no specific solution for perfect $\pm 45°$ diagonal edges, and such edges are not satisfactorily reconstructed at this low computational complexity. The reconstruction capability for larger block sizes used in JPEG2000 remains to be seen, and is a part of ongoing work.

Acknowledgments

WaveLab routines from Stanford University's Department of Statistics website were used to obtain the 2d-DWT using the biorthogonal 9-7 filter with symmetric extension, as specified by the JPEG2000 standard. J. Remus was supported by R.E.U (Research Experience for Undergraduates), a program funded by the National Science Foundation. He is with the University of Idaho, Moscow, ID 83843, USA. This work was partially supported by a grant from the Office of Naval Research ONR-N00014-97-1-0509, the Office of Naval Research Young Investigator Award, the Presidential Early Career Awards for Scientists and Engineers (PECASE), a National Science Foundation CAREER Award, and by the National Science Foundation Learning and Intelligent Systems Program (LIS).

References

[1] A.N.Skodras, C. Christopoulos, and T. Ebrahimi, "Jpeg2000: The upcoming still image compression standard," *Proceedings of the 11th Portuguese Conference on Pattern Recognition, Porto, Portugal*, pp. 359-366, May 2000.

[2] E. Chang, "An image coding and reconstruction scheme for mobile computing," in *Proceedings of the 5th IDMS (Springer-Verlag LNCS 1483), p.137-148, Oslo, Norway, September 1998.*, 1998.

[3] S. S. Hemami, "Digital image coding for robust multimedia transmission," in *Symposium on Multimedia Communications and Video Coding*, (New York, NY), 1995.

[4] D. J. Heeger and J. R. Bergen, "Pyramid-Based texture analysis/synthesis," in *SIGGRAPH 95 Conference Proceedings* (R. Cook, ed.), pp. 229-238, Addison Wesley, 1995.

[5] G. Fan and W. Cham, "Model-based edge reconstruction for low-bit-rate wavelet transform compressed images," Feb. 2000.

[6] S.-H. G. Chang, Z. Cvetković, and M. Vetterli, "Resolution enhancement of images using wavelet transform extrema extrapolation," in *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, vol. 4, (Detroit, MI, USA), pp. 2379-2382, 1995.

[7] V. DeBrunner, L. DeBrunner, L. Wang, and S. Radhakrishnan, "Error control and concealment for image transmission," *IEEE Communications Society Surveys and Tutorials*, vol. 3, no. 1, 2000.

[8] S. S. Hemami and T. H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Transactions on Image Processing*, vol. 4, pp. 1023-1027, July 1995.

[9] S. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *Submitted to IEEE Transactions on Image Processing*, 2001.

[10] J. W. Park, J. W. Kim, and S. U. Lee, "Dct coefficients recovery-based error concealment technique and its application to the mpeg-2 bit stream error," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 845-854, Dec. 1997.

[11] Z. Alkachouch and M. Bellanger, "Fast dct-based spatial domain interpolation of blocks in images," *IEEE Transactions on Image Processing*, vol. 9, pp. 729-732, Apr. 2000.

[12] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Data Compression Conference*, pp. 523-544, 2000.

[13] S. Shirani, F. Kossentini, and R. Ward, "Reconstruction of baseline jpeg coded images in error prone environments," *IEEE Transactions on Image Processing*, vol. 9, pp. 1292-1299, July 2000.

[14] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann Publishers, 2000.

[15] D. Salomon, *Data Compression: The Complete Reference*. Springer-Verlag, second ed., 2000.

[16] D. Taubman and M. Marcellin, *JPEG2000 Image Compression - Fundamentals, Standards, and Practice*. Kluwer Academic Publishers, first ed., 2002.