Embedded Wavelet-Based Image Compression: State of the Art

Eingebettete Wavelet-basierte Bildkompression: Stand der Technik

James E. Fowler, Mississippi State University, U.S.A.

Summary The architecture of modern image-compression algorithms built upon the embedded coding of wavelet coefficients is reviewed. The production of coefficients from wavelet filter banks is described along with the subsequent partitioning into significant and insigificant coefficient sets via bitplane coding. An overview of the zerotree and context-conditioning mechanisms for coding binary maps of significant coefficients is presented, and other less prominent approaches to significance-map coding are surveyed. Additionally, common approaches to refinement- and sign-bit coding are considered. Finally, the rate-distortion performance is empirically evaluated for several coders representative of each class, including the prominent Set Partitioning in Hierarchical Trees (SPIHT) algorithm and the recent JPEG-2000 standard. ►►► **Zusammenfassung** Der vorliegende Beitrag geht auf eingebettete Wavelet-basierte Bildkompressionsverfahren ein, wie sie zum Beispiel im neuen JPEG-2000 Standard zum Einsatz kommen. Diskrete Wavelet Transformationen im Zusammenspiel mit der Aufteilung der Koeffizienten in signifikante und nichtsignifikante Koeffizienten, die während der Kodierung eine herausragende Rolle spielt, werden beschrieben. Der Beitrag schließt mit einer empirischen Auswertung verschiedener Kodierer einschließlich des bekannten Set Partitioning in Hierarchical Trees (SPIHT) Algorithmus.

KEYWORDS I.4.2 [Image Processing] Compression

1 Introduction

In many applications involving communication of images, progressive transmission is desired in that successive reconstructions of the image are possible. In such a scenario, the receiver can produce a low-quality, or "thumbnail" representation of the image after having received only a small portion of the transmitted bitstream, and this "preview" of the image can be successively refined in quality or resolution (size) as more and more of the bitstream is received. Modern image-compression techniques support such progressive transmission through the use of embedded coding.

An embedded coding of an image is any coding such that 1) any prefix of length N bits of an M-bit coding is also a valid coding of the entire image, $0 < N \leq M$; and 2) if $N' > N$, then the distortion upon reconstructing from the length-N' prefix is less than or equal to that associated with the length-N prefix. Figures 1 and 2 illustrate the difference between transmission of typical nonembedded and embedded codings.

The general philosophy behind embedded coding lies in the recognition that each successive bit of the bitstream that is received reduces the distortion of the reconstructed image by a certain amount. Consequently, in order to achieve an embedded coding, we must organize information in the bitstream in decreasing order of importance,

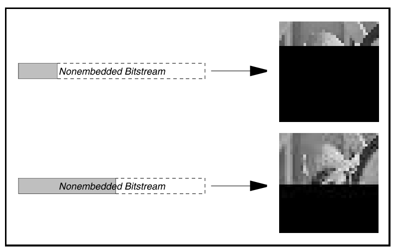


Figure 1 Transmission of a nonembedded coding.

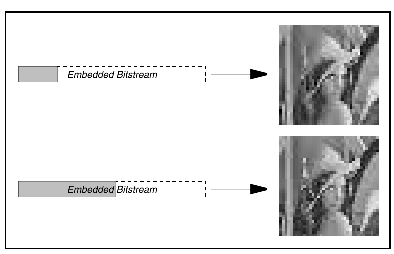


Figure 2 Transmission of an embedded coding.

 © Oldenbourg Verlag

where the most important information is defined to be that which produces the greatest reduction in distortion upon reconstruction. Although it is usually not possible to exactly achieve this ordering in practice, modern embedded image-compression algorithms do come close to approximating this optimal embedded ordering.

Modern embedded image coders are essentially built upon three major components: a wavelet transform, successive-approximation quantization, and significance-map encoding. Below, we overview these components and describe how each are implemented within several prominent algorithms, including the recent JPEG-2000 standard [1; 2].

2 The Discrete Wavelet Transform

Transforms aid the establishment of an embedded coding in that, for naturally occurring images, low-frequency components contain the majority of signal energy and are thus more important than high-frequency components to the image reconstruction. Wavelet transforms are currently the transform of choice for modern image coders since they not only provide this partitioning of information in terms of frequency but also retain much of the spatial structure of the original image. We will see below that many coders exploit this spatial structure in order to obtain substantial coding efficiency.

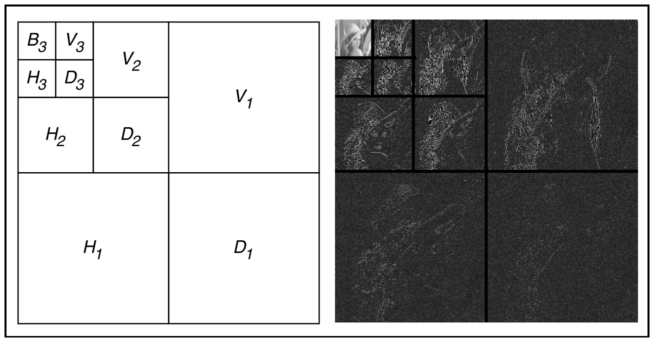


Figure 4 A 3-scale DWT. (**a**) Pyramid arrangement of subbands, (**b**) DWT coefficients of an image.

A discrete wavelet transform (DWT) can be implemented as a filter bank as illustrated in Figure 3. This filter bank decomposes the original image into horizontal (H), vertical (V), diagonal (D), and baseband (B) subbands, each being one-fourth the size of the original image. Wavelet theory provides filter-design methods such that the filter bank is perfectly reconstructing (i. e., there exists a reconstruction filter bank that will generate exactly the original image from the decomposed subbands H, V, D, and B), and such that the lowpass and highpass filters have finite impulse responses (which aids practical implementation). Multiple stages of decomposition can be cascaded together by recursively decomposing the baseband; the subbands in this case are usually arranged in a pyramidal form as illustrated in Figure 4, where subband S_j is subband S at decomposition stage j, $S \in \{H, V, D\}$.

Regarding Figure 4, we observe the following facts about the DWT of an image:

(1) since most images are lowpass in nature, most signal energy is compacted into the lower-frequency subbands (i.e., the baseband and the S_j subbands where j is large);
(2) most coefficients in S_j are zero for small j;
(3) small- or zero-valued coefficients tend to be clustered together within subband S_j; and
(4) clusters of small- or zero-valued coefficients in subband S_j tend to be located in the same relative spatial position as similar clusters in subband S_{j+1}.

3 Bitplane Coding

The partitioning of information into DWT subbands somewhat inherently supports embedded coding in that transmitting coefficients by ordering the subbands as B_J, H_J, V_J, D_J, H_{J-1}, V_{J-1}, D_{J-1}, …, implements a decreasing order of importance. However, more is needed to produce a truly embedded bitstream—even if coefficient $c_i \in S_j$ is more important than coefficient $c_k \in S_j$, not every bit of c_i is necessarily more important than every bit of c_k. That is, not only should the coefficients be transmitted in decreasing order of importance, but also the individual bits that constitute the coefficients should be ordered as well.

Specifically, to effectuate an embedded coding of a set of coefficients, we represent the coefficients in sign-magnitude form as illustrated in Figure 5 and code the sign and magnitude of the coefficients separately. For coefficient-magnitude coding, we transmit the

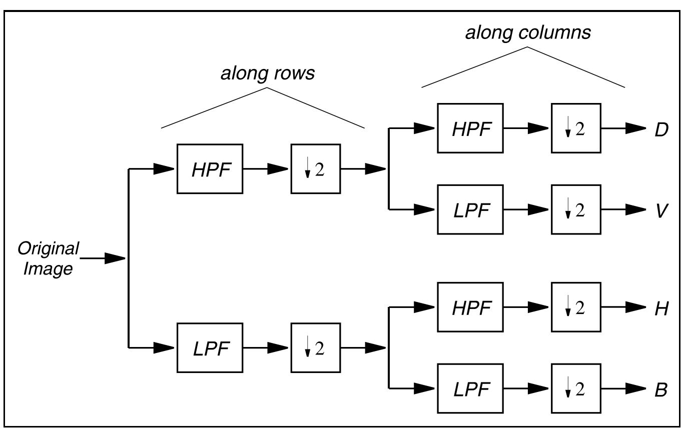


Figure 3 One stage of DWT decomposition composed of lowpass (LPF) and highpass (HPF) filters applied to the columns and rows independently.

Bitplane		Coefficients 11	2	−3	6
	Sign	0	0	1	0
MSB	3	1	0	0	0
	2	0	0	0	1
	1	1	1	1	1
LSB	0	1	0	1	0

Figure 5 Bitplanes of the sign-magnitude representation of coefficients for bitplane coding.

most significant bit (MSB) of *all* coefficient magnitudes, then the next-most significant bit of all coefficient magnitudes, etc., such that each coefficient is successively approximated. This "bitplane-coding" scheme is contrary to the usual binary representation which would output all bits of $|c_0|$, then all bits of $|c_1|$, etc. The net effect of the bitplane coding is that each coefficient magnitude is successively quantized by dividing the interval in which it is known to reside in half and outputting a bit to designate the appropriate subinterval, as illustrated in Figure 6.

In practice, bitplane coding is usually implemented by performing two passes through the set of coefficients for each bitplane—the significance pass and the refinement pass. We define the significance state x_i with respect to threshold t of coefficient c_i as $x_i = 1$ if $|c_i| \geq t$ (i. e., c_i is a *significant* coefficient), and $x_i = 0$ otherwise (i. e., c_i is *insignificant*). The significance pass describes x_i for all the coefficients in the DWT that are currently known to be insignificant but may become significant for the current threshold. On the other hand, the refinement pass produces a successive approximation to those coefficients that are already known to be significant by coding the current coefficient-magnitude bitplane for those significant coefficients. After each iteration of the significance and refinement passes, the significance threshold is divided in half, and the process is repeated for the next bitplane.

4 Significance-Map Coding

The collection of x_i values for all the coefficients in the DWT of an image is called the significance map for a particular threshold value. Given our observations of the nature of DWT coefficients in Section 2, we see that for most of the bitplanes (particularly for large t), the significance map will be only sparsely populated with nonzero values. Consequently, the task of the significance pass is to create an efficient coding of this sparse significance map at each bitplane; the efficiency of this coding will be crucial to the overall compression efficiency of the image coder. Below, we review several approaches that prominent algorithms have taken for the efficient coding of significance-map information.

4.1 Zerotrees

As illustrated in Figure 7, except in subbands S_1 and in the baseband, every "parent" coefficient in subband S_j can be related to *four* "children" coefficients in the same relative spatial location in subband S_{j-1}. In the baseband B_J, each parent has three children, one in each of H_J, V_J, and D_J. A *zerotree* is formed when a coefficient and all of its descendants are insignificant with respect to the current threshold, while a *zerotree root* is defined to be a coefficient that is part of a zerotree yet is not the descendant of another zerotree root.

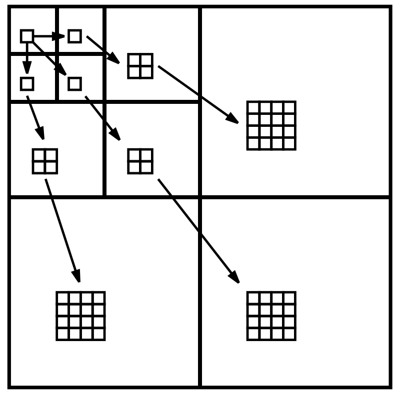


Figure 7 Parent-child relationships between subbands of a DWT.

The Embedded Zerotree Wavelet (EZW) algorithm [3] was the first image coder to make use of zerotrees for the coding of significance-map information. This coder is based on the observation that if a coefficient is found to be insignificant, it is likely that its descendants are also insignificant. Consequently, the occurrence of a zerotree root in the baseband or in the lower-frequency subbands can lead to substantial coding efficiency since we can denote the zerotree root as a special "Z" symbol in the significance map, and not code all of the descendants which are known then to be insignificant by definition. The EZW algorithm then proceeds to code the significance map in a raster scan within each subband, starting with B_J and progressing to the high-frequency subbands. In this raster scan a significant coefficient is denoted by either a "+" or "–" symbol depending on whether the coefficient value is positive or negative, while zerotree roots are denoted by the "Z" symbol and isolated insignificant coefficients (i. e., insignificant coefficients not forming

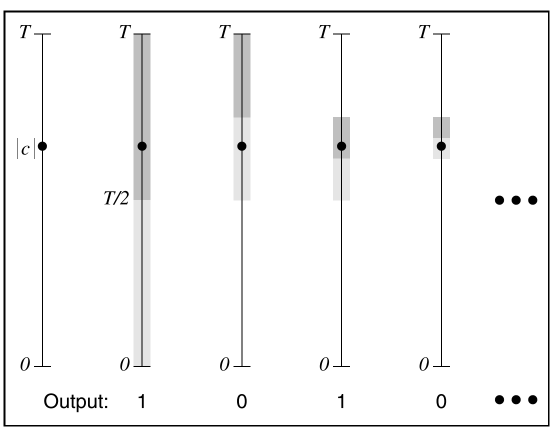


Figure 6 Successive-approximation quantization of a coefficient magnitude $|c|$ in interval $[0, T]$ where T is an integer power of 2.

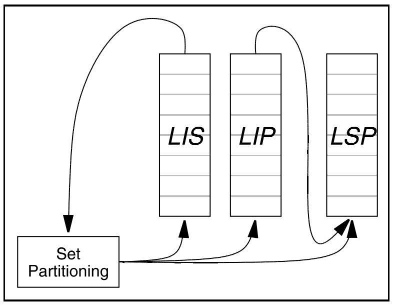


Figure 8 Processing of sorted lists in SPIHT.

a zerotree root) are denoted by the "I" symbol. A lossless entropy coding of this symbol stream then produces a compact representation of the significance map. The significance threshold is halved, and the zerotree coding process is repeated for each successive bitplane. Note that, once a coefficient becomes significant and is coded with a "+" or "–," no further information concerning that coefficient need be coded in the significance pass for subsequent bitplanes.

The Set Partitioning in Hierarchical Trees (SPIHT) algorithm [4] improves upon the zerotree concept by replacing the raster scan with a number of sorted lists that contain sets of coefficients (i.e., zerotrees) and individual coefficients. These lists are illustrated in Figure 8. In the significance pass of the SPIHT algorithm, the list of insignificant sets (LIS) is examined in regard to the current threshold; any set in the list that is no longer a zerotree with respect to the current threshold is then partitioned into one or more smaller zerotree sets, isolated insignificant coefficients, or significant coefficients. Isolated insignificant coefficients are appended to the list of insignificant pixels (LIP), while significant coefficients are appended to the list of significant pixels (LSP). The LIP is also examined, and, as coefficients become significant with respect to the current threshold, they are appended to the LSP. Binary symbols are encoded to describe motion of sets and coefficients between the three lists. Since the lists remain implicitly sorted in an importance ordering, SPIHT achieves a high degree of embedding and compression efficiency.

4.2 Conditional Coding

Recent work [5] has indicated that typically the ability to predict the insignificance of a coefficient through parent-child relationships such as those employed by zerotree algorithms is somewhat limited compared to the predictive ability of neighboring coefficients within the same subband. Consequently, recent algorithms have focused on coding significance-map information using only within-subband information. The typical approach is to employ multiple-context adaptive arithmetic coding.

Adaptive arithmetic coding (AAC) [6] is a lossless coding technique that codes a stream of symbols into a bitstream with length very close to its theoretical minimum limit. Suppose source X produces symbol i with probability p_i. The *entropy* of source X is defined to be

$$H(X) = -\sum_i p_i \log_2 p_i \,, \qquad (1)$$

where $H(X)$ has units of bits per symbol (bps). One of the fundamental tenets of information theory is that the average bit rate in bps of the most efficient lossless (i. e., invertible) compression of source X cannot be less than $H(X)$. In practice, AAC often produces compression quite close to $H(X)$ by estimating the probabilities of the source symbols with frequencies of occurrence as it codes the symbol stream. Essentially, the better able AAC can estimate p_i, the closer it will come to the $H(X)$ lower bound on compression efficiency. Oftentimes, the efficiency of AAC can be improved by *conditioning* the coder with known *context* information and maintaining separate symbol-probability estimates for each context. That is, limiting attention of AAC to a specific context usually reduces the variety of symbols, thus permitting better estimation of the probabilities within that context and producing greater compression efficiency. From a mathematical standpoint, the *conditional entropy* of source X with known information Y is $H(X|Y)$. Since it is well known from information theory that

$$H(X|Y) \leq H(X) \,, \qquad (2)$$

conditioning AAC with Y as the context will (usually) produce a bitstream with a smaller bit rate.

The usual approach to employing AAC with context conditioning for the significance-map coding of an image is to use the known significance states of neighboring coefficients to provide the context for the coding of the significance state of the current coefficient. Specifically, the eight neighboring significance states to x_i are shown in Figure 9. Given that each neighbor takes on a binary value, there are $2^8 = 256$ possible contexts.

JPEG-2000 [1], the most prominent conditional-coding technique, uses contexts derived from the neighbors depicted in Figure 9, but reduces the number of distinct contexts to nine, since not all possible contexts were found to be useful. To further improve the context conditioning, as well as to increase the degree of embedding, JPEG-2000 splits the coding of the significance map into two separate passes rather than employ one significance pass as do most other algorithms. Specifically, JPEG-2000 uses a significance-propagation pass that codes those coefficients that are currently insignificant but have at least one neighbor that is already significant. This pass accounts for all coefficients that are likely to become significant in the current bitplane. The remaining insignificant coefficients are coded in the cleanup pass; these coefficients, which are surrounded by insignificant coeffi-

d_0	v_0	d_1
h_0	x_i	h_1
d_2	v_1	d_3

Figure 9 Significance-state neighbors to x_i.

cients, are likely to remain insignificant. Both passes use the same nine contexts. In addition, the cleanup pass includes one additional context used to encode four successive insignificant coefficients together with a single "insignificant run" symbol.

4.3 Other Significance-Map Techniques

A number of alternatives to zerotrees and conditional coding have been proposed for significance-map coding. Since, for a given significance threshold, the significance map is essentially a binary image, techniques that have long been employed for the coding of bilevel images are applicable. Specifically, runlength coding is the fundamental compression algorithm behind the Group 3 fax standard; the Wavelet Difference Reduction (WDR) [7] combines runlength coding of the significance map with an efficient lossless representation of the runlength symbols to produce an embedded image coder. Another approach to binary-image coding is to use quadtrees; the Set Partitioned Embedded Block Coder (SPECK) [8; 9] employs quadtrees for significance-map coding. Finally, an all-together different approach to significance-map coding was proposed recently—the tarp coder of [10] uses a nonadaptive arithmetic coder coupled with an explicit probability estimate of the significance map generated by Parzen windows, a well known method of nonparametric probability-density estimation. In [10], this probability estimate is efficiently computed by a novel series of 1D filtering operations.

5 The Refinement Pass and Sign Coding

In most embedded image coders, after the significance map is coded for a particular bitplane, a refinement pass proceeds through the coefficients, coding the current bitplane value of each coefficient that is already known to be significant but did not become significant in the immediately preceding significance pass. These "refinement bits" permit the reconstruction of the significant coefficients with progressively greater accuracy. It is usually assumed that the occurrence of a 0 or 1 is equally likely in bitplanes other than the MSB for a particular coefficient; consequently, most algorithms take little effort to code the refinement bits and may simply output them unencoded into the bitstream. Recently, it has been recognized that the refinement bits possess some correlation to their neighboring coefficients [11], particularly for the more significant bitplanes; consequently, the JPEG-2000 standard calls for the conditional coding of refinement bits with three contexts. JPEG-2000 also departs somewhat from the typical significance-pass/refinement-pass processing order since the significance pass is split in two separate passes—in JPEG-2000, the refinement pass occurs between the significance-propagation pass and the cleanup pass.

The significance and refinement passes encode the coefficient magnitudes; to reconstruct the wavelet coefficients, the coefficient signs must also be encoded. As with the refinement bits, most algorithms assume that any given coefficient is equally likely to be positive or negative; however, recent work [11–13] has shown that there is some structure to the sign information that can be exploited to improve coding efficiency. Consequently, JPEG-2000 classifies the signs of neighboring coefficients into one of five contexts for coding of the sign of the current coefficient. As in other embedded algorithms, in JPEG-2000, the sign of a coefficient is encoded into the bitstream immediately following the encoding of the significance-map information that indicates that a coefficient is transitioning from insignificant to significant.

6 Performance Comparison

Two quantities must be considered when comparing the performance of image-compression algorithms—the *distortion* introduced by the compression measured as the difference between the original image and its reconstruction, and the number of bits in the compressed bitstream, usually measured as a bit *rate* in number of bits per pixel (bpp). Most often, distortion is measured as a peak signal-to-noise ratio (PSNR), which is defined for an 8-bit grayscale image as

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{D}, \qquad (3)$$

where D is the mean square error (MSE) between the original image and the reconstructed image. The PSNR has units of decibels (dB).

Figure 10 plots the rate-distortion performance for a variety of

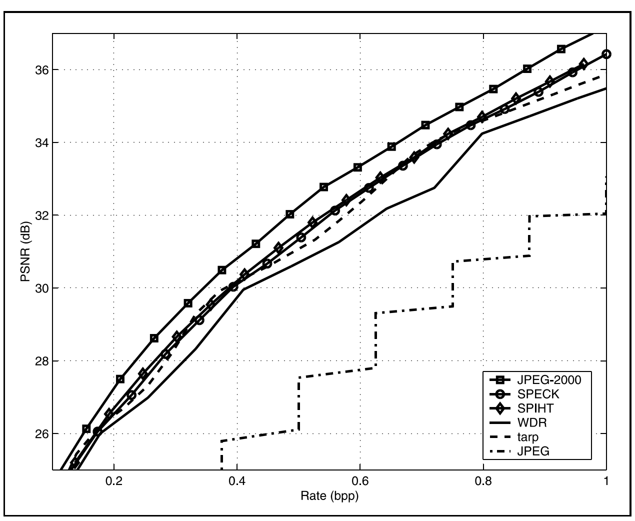


Figure 10 Rate-distortion performance for a variety of image coders for the "barbara" image.

Table 1 PSNR performance at 0.5 bpp.

Algorithm	PSNR (dB) lenna	barbara	goldhill
JPEG-2000	37.3	32.2	33.2
SPECK	37.1	31.5	33.0
SPIHT	37.1	31.3	33.0
tarp	36.7	31.1	33.0
WDR	36.5	30.7	32.8
JPEG	34.6	27.8	31.4

the algorithms we have considered in this paper, including both a zerotree-based technique (SPIHT [4]) and a conditional-coding technique (JPEG-2000 [1]), as well as three other techniques based on other forms of significance-map coding (WDR [7], SPECK [8; 9], and tarp [10]). All coders use a 5-stage wavelet decomposition with the popular 9-7 wavelet filters from [14]. Also shown is the performance for the original JPEG standard [15; 16] which is a nonembedded coder that represented the state of the art in image coder before the rise of embedded wavelet-based coding. The QccPack [17] (http://qccpack.sourceforge.net) implementations for SPIHT, WDR, SPECK, and tarp are used, while JPEG-2000 is Kakadu Ver. 3.4 (http://www.kakadusoftware.com) and JPEG is the Independent JPEG Group implementation (http://www.ijg.org). Table 1 shows the PSNR for several images when all coders produce bitstreams at a rate of 0.5 bpp.

7 Conclusions

In the experimental comparisons of the previous section, we observe that the embedded wavelet-based coders examined perform fairly close to one another, with JPEG-2000 exhibiting a slight advantage in rate-distortion performance. However, it should be noted that JPEG-2000 offers a variety of features (e. g., lossless compression, random bitstream access, large-image coding) not available in the other implementations we have employed. Nonetheless, we see that the modern paradigm of embedded wavelet-based image compression substantially outperforms the prior state of the art as represented by the original JPEG standard. Additionally, this increased compression efficiency is achieved alongside an inherent capacity for progressive transmission, a characteristic in high demand in modern applications yet absent from prior nonembedded techniques.

References

[1] ISO/IEC 15444-1. *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, 2000.

[2] D. Taubman and M. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, MA, 2002.

[3] J. Shapiro. "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[4] A. Said and W. Pearlman. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[5] M. Marcellin and A. Bilgin. "Quantifying the parent-child coding gain in zero-tree-based coders," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 67–69, Mar. 2001.

[6] I. Witten, R. Neal, and J. Cleary. "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[7] J. Tian and R. Wells. "Embedded image coding using wavelet difference reduction," in *Wavelet Image and Video Compression*, P. Topiwala, Ed., chapter 17, pp. 289–301. Kluwer Academic Publishers, Boston, MA, 1998.

[8] A. Islam and W. Pearlman. "An embedded and efficient low-complexity hierarchical image coder," in *Visual Communications and Image Processing*, K. Aizawa, R. Stevenson, and Y. Zhang, Eds., San Jose, CA, Jan. 1999, Proc. SPIE 3653, pp. 294–305.

[9] W. Pearlman, A. Islam, N. Nagaraj, and A. Said. "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. on Circuits and Systems for Video Technology*, 2003.

[10] P. Simard, D. Steinkraus, and H. Malva. "On-line adaptation in image coding with a 2-D tarp filter," in *Proc. of the IEEE Data Compression Conf.*, J. Storer and M. Cohn, Eds., Snowbird, UT, Apr. 2002, pp. 23–32.

[11] D. Taubman. "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[12] A. Deever and S. Hemami. "What's your sign?: Efficient sign coding for embedded wavelet image coding," in *Proc. of the IEEE Data Compression Conf.*, James A. Storer and M. Cohn, Eds., Snowbird, UT, Mar. 2000, pp. 273–282.

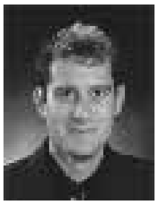
[13] A. Deever and S. Hemami. "Efficient sign coding and estimation of zero-quantized coefficients in embedded wavelet image codecs," *IEEE Trans. on Image Processing*, vol. 12, no. 4, pp. 420–430, Apr. 2003.

[14] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[15] ISO/IEC 10918-1. *Digital Compression and Coding of Continuous-tone Still Image—Part 1: Requirements and guidelines*, 1991.

[16] W. Pennebaker and J. Mitchell, *JPEG Still Image Compression Standard*, Kluwer Academic Publishers, Boston, MA, 1993.

[17] James E. Fowler. "QccPack: An open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, A. Tescher, Ed., San Diego, CA, Aug. 2000, Proc. SPIE 4115, pp. 294–301.



Dr. James E. Fowler received the B.S. degree in computer and information science engineering and the M.S. and Ph.D. degrees in electrical engineering in 1990, 1992, and 1996, respectively, all from the Ohio State University. In 1995, he was an intern researcher at AT&T Labs in Holmdel, NJ, and, from January to July of 1997, he held an NSF-sponsored postdoctoral assignment at the Université de Nice-Sophia Antipolis, France. He is currently an associate professor in the Department of Electrical & Computer Engineering at Mississippi State University in Starkville, MS. He is also a researcher in the Visualization, Analysis, and Imaging Laboratory (VAIL) within the GeoResources Institute (GRI) at the Mississippi State ERC. His research interests include image and video coding, data compression, and wavelets.
Address: Box 9627, Mississippi State, MS 39762 USA. E-Mail: fowler@ece.msstate.edu

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

In der Fakultät für Elektrotechnik, Informatik und Mathematik ist nachfolgende Professur zu besetzen:

Im **Institut für Elektrotechnik und Informationstechnik**

Universitätsprofessur (C 4) für Datentechnik
(Kennziffer 528)

Bevorzugte Schwerpunkte der wissenschaftlichen Arbeit sind:
- Realzeit-Datenverarbeitung (Prozessrechentechnik, eingebettete Systeme)
- Moderne Rechnerarchitekturen
- Verteilte Systeme (ambient intelligence, pervasive/mobile computing)
- Fehlertoleranz, Sicherheit, Integrität und Verlässlichkeit von digitalen Systemen

Es wird Wert darauf gelegt, dass praxisnahe Fragestellungen in Kooperation mit angrenzenden Fachgebieten, insbesondere der Technischen Informatik, des Maschinenbaus und der Naturwissenschaften bearbeitet werden.

Eine aktive Beteiligung an den Lehrveranstaltungen der Studiengänge der Elektrotechnik/Informationstechnik und der Ingenieurinformatik im Grund- und Hauptstudium wird erwartet.

Wir erwarten von den Bewerberinnen/ den Bewerbern Bereitschaft und Interesse, an interdisziplinären Einrichtungen in Paderborn, insbesondere an Arbeiten des C-LAB, des Heinz Nixdorf Instituts, des PC^2 – Paderborn Center for Parallel Computing, des SFB 614 „Selbstoptimierende Systeme des Maschinenbaus“ und der NRW International Graduate School „Dynamic Intelligent Systems“ kooperativ mitzuwirken. Die Bereitschaft, Lehrveranstaltungen in englischer Sprache anzubieten, wird vorausgesetzt.

Einstellungsvoraussetzungen: § 46 Abs. 1 Ziff. 4a HG NW (Habilitation oder habilitationsadäquate Leistungen) und Ziff. 2 HG NW (pädagogische Eignung).

Die Universität Paderborn strebt eine Erhöhung des Anteils an Frauen in Hochschullehrerfunktion an und fordert daher Frauen nachdrücklich auf, sich zu bewerben. Frauen werden nach § 7 LGG bei gleicher Eignung, Befähigung und fachlicher Leistung bevorzugt berücksichtigt. Ebenso ist die Bewerbung geeigneter Schwerbehinderter und Gleichgestellter im Sinne des Sozialgesetzbuches Neuntes Buch (SGB IX) erwünscht.

Bewerbungen mit den üblichen Unterlagen werden innerhalb von 4 Wochen nach Veröffentlichung unter Angabe der jeweiligen Kennziffer erbeten an den Dekan der Fakultät für Elektrotechnik, Informatik und Mathematik der Universität Paderborn, Warburger Str. 100, 33098 Paderborn.