Microsoft®
微软中国研究院

Microsoft

Research

Paper

Microsoft Research China Paper Collection

微软中国研究院论文选集

1998年11月5日，微软中国研究院在北京正式成立。成立仪式上，我们提出了微软中国研究院的长远发展目标：成为世界一流、亚洲最高水准的计算机基础研究机构。

为此，在过去的一年中，我们一直在非常努力地工作着。我认为，成为一个世界一流、亚洲最高水准的学术研究机构，有以下几个重要的参考因素。其一，要有世界一流的人才；其二：要有大量高水平的学术论文在国际知名的学术刊物或会议上发表；其三：要有相当数量的发明创造，包括专利；其四：能够最终转化为产品，造福世界上千千万万的计算机用户。当然，一个良好的研究环境也是不可或缺的。

在这一年里，我们从世界各地聘用了一批国际一流的计算机基础研究人员。在研究院内部，一个良好的学术研究环境和气氛已经形成。在这本论文集中，选编了微软中国研究院张亚勤博士、沈向洋博士、张宏江博士和刘文印博士的部分论文，共28篇。它们都已在国际一流的学术刊物或会议上发表，有很深的学术造诣。在此，我向他们表示祝贺！

此次论文选集的出版也是为了答谢在过去一年里给予我们关心和支持的各界朋友。在今后的发展过程中，我们将一如既往地与政府部门、高等院校和科研机构密切合作，共同促进中国基础研究水平的提高。

我们相信此次论文选集的出版是一个很好的开始。我们会一步一个脚印，向着我们的既定目标，不懈地努力！

微软中国研究院院长
李开复
一九九九年十一月

CONTENTS 目录

by Ya-Qin Zhang et al. 张亚勤 **01–173**

(Continued)

CONTENTS 目录

(Continued)

CONTENTS 目录

Scalable Wavelet Coding for Synthetic/Natural Hybrid Images

Iraj Sodagar, Hung-Ju Lee, Paul Hatrack, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*— This paper describes the texture representation scheme adopted for MPEG-4 synthetic/natural hybrid coding (SNHC) of texture maps and images. The scheme is based on the concept of multiscale zerotree wavelet entropy (MZTE) coding technique, which provides many levels of scalability layers in terms of either spatial resolutions or picture quality. MZTE, with three different modes (single-Q, multi-Q, and bilevel), provides much improved compression efficiency and fine-gradual scalabilities, which are ideal for hybrid coding of texture maps and natural images. The MZTE scheme is adopted as the baseline technique for the visual texture coding profile in both the MPEG-4 video group and SNHC group. The test results are presented in comparison with those coded by the baseline JPEG scheme for different types of input images. MZTE was also rated as one of the top five schemes in terms of compression efficiency in the JPEG2000 November 1997 evaluation, among 27 submitted proposals.**

***Index Terms*— Compression, image and video coding, JPEG-2000, MPEG-4, texture coding, wavelet.**

I. Introduction

SCALABLE picture coding has received considerable attention lately in academia and industry in terms of both coding algorithms and standards activities. In many applications, enhanced features such as content-based scalability, content-based access, and manipulations are required in addition to increased compression efficiency, as exemplified by the effort undertaken in the emerging MPEG-4 international standard [1]–[3]. In contrast to the MPEG-1 and MPEG-2 standards, MPEG-4 includes increased flexibility and user interaction at many different levels for both natural and synthetic image/video contents. This paper describes the scalable texture coding scheme using zerotree wavelet techniques. Zerotree wavelet coding provides very high compression efficiency as well as spatial and quality scalability features compared with discrete cosine transform (DCT)-based approaches [16], [17]. With the advantages of its scalability and high compression, the zerotree wavelet coding technique was adopted in the MPEG-4 standard as the visual texture coding tool [3], which allows the hybrid coding of natural images and video (e.g., acquired with cameras) together with synthetic scenes (e.g., generated by computers).

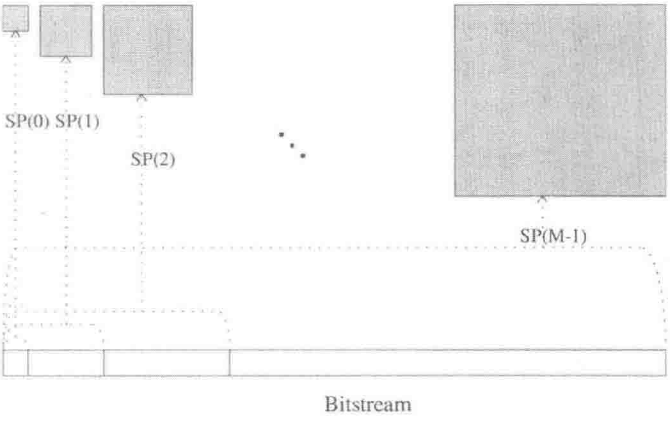


Fig. 1. N layers of spatial scalability.

The organization of this paper is as follows. We begin with a general overview of the scalable image coding and its features. Then, we describe the embedded zerotree wavelet (EZW) that provides fine scalability and zerotree entropy coding (ZTE) that provides spatial scalability with much higher compression efficiency than EZW. Finally, the most general case, known as the multiscale zerotree wavelet entropy (MZTE) coding, is presented to provide an arbitrary number of spatial and quality scalability as well as good compression efficiency. The MPEG-4 still-picture visual texture coding consists of three modes: single-Quant, multiple-Quant, and bilevel Quant. This paper mainly describes the first two modes, developed by Sarnoff Corp. The bilevel mode is presented in detail in [18], and the extension of MZTE to include arbitrary shape wavelet coding, another important feature of MPEG-4, is covered in [10].

II. Scalable Texture Coding Using Wavelets

In scalable compression, the bitstream can be progressively decoded to provide different versions of the image in terms of spatial resolutions (spatial scalability), quality levels [signal-to-noise ratio (SNR) scalability], or combinations of spatial and quality scalabilities.

Figs. 1 and 2 show two examples of such scalabilities. In Fig. 1, the bitstream has M layers of spatial scalability. In this case, the bitstream consists of M different segments. By decoding the first segment, the user can see a preview version of a decoded image at a lower resolution. Decoding the second segment results in a larger reconstructed image. Furthermore,

Manuscript received October 1, 1997; revised November 1, 1998. This paper was recommended by Guest Editors H. H. Chen, T. Ebrahimi, G. Rajan, C. Horne, P. K. Doenges, and L. Chiariglione.

I. Sodagar, H.-J. Lee, and P. Hatrack are with the Multimedia Technology Laboratory, Sarnoff Corp., Princeton, NJ 08512 USA.

Y.-Q. Zhang was with the Multimedia Technology Laboratory, Sarnoff Corp., Princeton, NJ 08512 USA. He is now with Microsoft Research, Beijing 100080 China (e-mail: yzhang@microsoft.com).

Publisher Item Identifier S 1051-8215(99)02338-1.

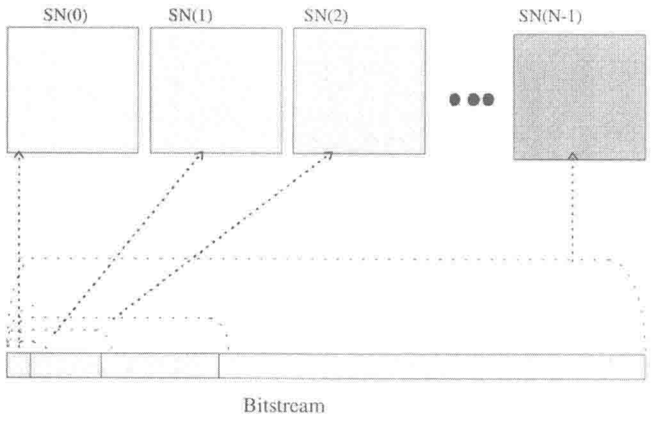


Fig. 2. M layers of quality scalability.

by progressively decoding the additional segments, the viewer can increase the spatial resolution of the image. Fig. 2 shows an example in which the bitstream includes N layers of quality scalability. In this figure, the bitstream consists of N different segments. Decoding the first segment provides an early view of the reconstructed image. Further decoding of the next segments results in increase of the quality of the reconstructed image at N steps.

Fig. 3 shows a more complex case of combined spatial-quality scalabilities. In this example, the bitstream consists of M spatial layers, and each spatial layer includes N layers of quality scalability. In this case, both the spatial resolution and the quality of the decoded image can be improved by progressively decoding the bitstream.

Zerotree wavelet coding is a proven technique for coding wavelet transform coefficients [4]–[12], [16]. Besides superior compression performance, the advantages of zerotree wavelet coding include simplicity, embedded bitstream structure, scalability, and precise bit-rate control. Zerotree wavelet coding is based on three key ideas: 1) using wavelet transform for decorrelation, 2) exploiting the self-similarity inherent in the wavelet transform to predict the location of significant information across scales, and 3) universal lossless data compression using adaptive arithmetic coding. In this section, first we give a brief description of Shapiro's EZW [4]. The EZW technique always generates a bitstream with the maximum possible number of quality scalability layers. Then, we describe the ZTE technique [13], [14], which only provides spatial scalability but with much higher compression efficiency. Last, the most general technique, MZTE [3], is described, which provides a flexible framework to encode the images with an arbitrary number of spatial or quality scalabilities.

A. Embedded Zerotree Wavelet (EZW) Coding [4]

EZW coding is applied to coefficients resulting from a discrete wavelet transform (DWT). The DWT decomposes the input image into a set of subbands of varying resolutions. The coarsest subband is a low-pass approximation of the original image, and the other subbands are finer-scale refinements. In the hierarchical subband system such as that of the wavelet transform, with the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients of similar orientation at the next finer scale. The coefficient at the coarse scale is called the parent, and all coefficients at the same spatial location, and of similar orientation at the next finer scale, are called children. As an example, Fig. 4 shows a wavelet tree descending from a coefficient in the subband HH3. For the lowest frequency subband, LL3 in the example, the parent–child relationship is defined such that each parent node has three children, one in each subband at the same scale and spatial location but different orientation.

EZW introduced a data structure called a *zerotree,* built on the parent–child relationship. The zerotree structure takes advantage of the principle that if a wavelet coefficient at a coarse scale is *insignificant* (quantized to zero) with respect to a given threshold T, then all wavelet coefficients of the same orientation at the same spatial location at finer wavelet scales are also likely to be insignificant with respect to that T. The zerotree structure is similar to the zigzag scanning and end-of-block symbol commonly used in coding DCT coefficients.

EZW scans wavelet coefficients subband by subband. Parents are scanned before any of their children, but only after all neighboring parents have been scanned. Each coefficient is compared against the current threshold T. A coefficient is significant if its amplitude is greater than T; such a coefficient is then encoded using one of the symbols *negative significant* (NS) or *positive significant* (PS). The *zerotree root* (ZTR) symbol is used to signify a coefficient below T, with all its children in the zerotree data structure also below T. The *isolated zero* (IZ) symbol signifies a coefficient below T but with at least one child not below T. For significant coefficients, EZW further encodes coefficient values using a successive approximation quantization (SAQ) scheme. Coding is done bit-plane by bit-plane. The successive approximation approach to quantization of the wavelet coefficient leads to the embedded nature of an EZW coded bitstream.

B. Zerotree Entropy (ZTE) Coding

ZTE coding is an efficient technique for coding wavelet transform coefficients. It is based on, but differs significantly from, the EZW algorithm. Like EZW, this new ZTE algorithm exploits the self-similarity inherent in the wavelet transform of images and video residuals to predict the location of information across wavelet scales. ZTE coding organizes quantized wavelet coefficients into wavelet trees and then uses zerotrees to reduce the number of bits required representing those trees. ZTE differs from EZW in four major ways.

1) Quantization is explicit instead of implicit and can be performed distinctly from the zerotree growing process or can be incorporated into the process, thereby making it possible to adjust the quantization according to where the transform coefficient lies and what it represents in the frame.
2) Coefficient scanning, tree growing, and coding are performed bit-plane by bit-plane.
3) Coefficient scanning can be changed from subband by subband to a depth-first traversal of each tree.

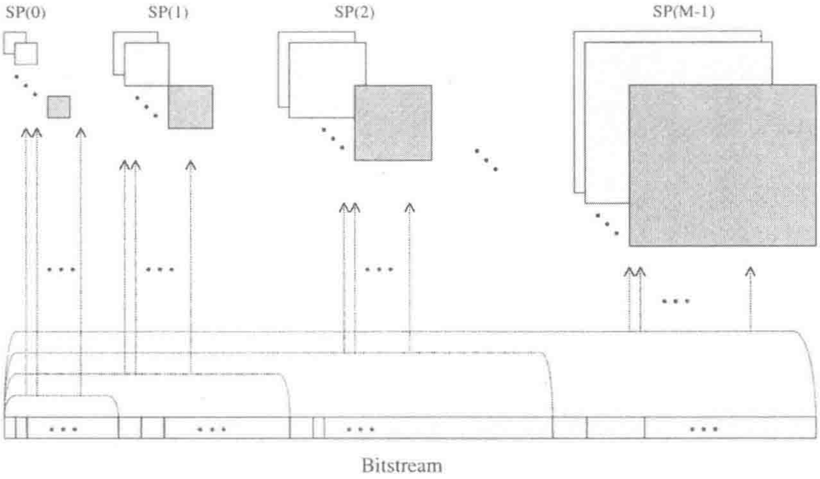


Fig. 3. $N \times M$ layers of spatial/quality scalabilities.

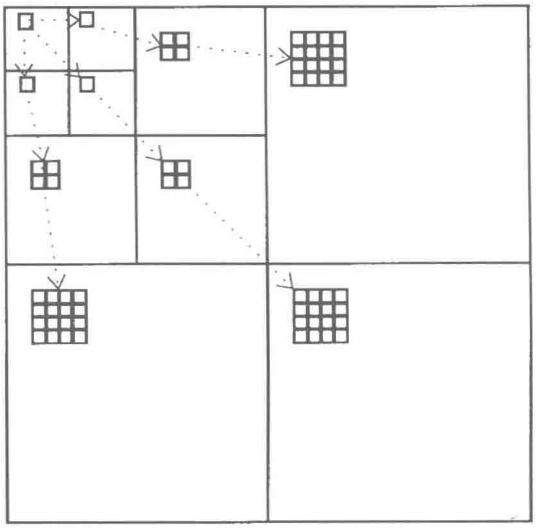


Fig. 4. The parent–child relationship of wavelet coefficients.

4) The alphabet of symbols for classifying the tree nodes is changed to one that performs significantly better for very low bit-rate encoding of video.

Although ZTE does not produce a fully embedded bitstream like EZW, it gains flexibility and other advantages over EZW coding, including substantial improvement in compression efficiency, simplicity, and spatial quality.

In ZTE coding, the coefficients of each wavelet tree are reorganized to form a wavelet block, as shown in Fig. 5. Each wavelet block comprises those coefficients at all scales and orientations that correspond to the image at the spatial location of that block. The concept of the wavelet block provides an association between wavelet coefficients and what they represent spatially in the image. The ZTE entropy encoding is performed by assigning the zerotree symbol of a coefficient, then encoding the coefficient value with its symbol in one of the two different scanning orders described later. A symbol is assigned to each node in a wavelet tree describing the wavelet coefficient corresponding to that node. Quantization of the wavelet transform coefficients can be done prior to the construction of the wavelet tree or as a separate task, or it can be incorporated into the wavelet tree construction. In the second case, as a wavelet tree is traversed for coding, the wavelet coefficients can be quantized in an adaptive fashion, according to spatial location and/or frequency content.

The extreme quantization required to achieve a very low bit rate produces many zero coefficients. Zerotrees exist at any tree node where the coefficient is zero and all the node's children are zerotrees. The wavelet trees are efficiently represented by scanning each tree depth-first from the root in the low-low band through the children and assigning one of four symbols to each node encountered: *ZTR, valued ZTR* (VZTR), *value* (VAL), or IZ. A zerotree root denotes a coefficient that is the root of a zerotree. Zerotrees do not need to be scanned further because it is known that all coefficients in such a tree have an amplitude of zero. A valued zerotree root is a node where the coefficient has nonzero amplitude and all four children are zerotree roots. The scan of this tree can stop at this symbol. A value symbol identifies a coefficient with amplitude nonzero and also with some nonzero descendant. Last, an isolated symbol identifies a coefficient with an amplitude of zero but with some nonzero descendant. The symbols and quantized coefficients are then losslessly encoded using an adaptive arithmetic coder. The arithmetic encoder adaptively tracks the statistics of the zerotree symbols.

In ZTE coding, the quantized wavelet coefficients are scanned either in the tree-depth or the band-by-band fashion. In the tree-depth scanning order, all coefficients of each tree are encoded before starting encoding of the next tree. In the band-by-band scanning order, all coefficients are encoded from the lowest to the highest frequency subbands. The wavelet coefficients of dc band are encoded independently from the other bands. First, the quantization step size is encoded, then the magnitude of the minimum value of the differential quantization indexes "band_offset" and the maximum value of the differential quantization indexes "band_max_value" are encoded into the bitstream. The parameter band_offset is a negative or a zero integer, and the parameter band_max is a positive integer, so only the magnitudes of these parameters are written into the bitstream. The differential quantization

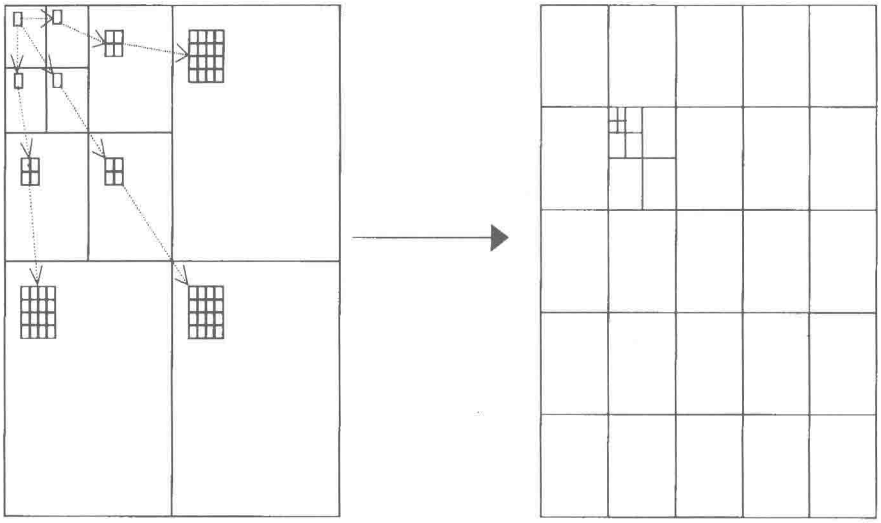


Fig. 5. Building wavelet blocks after taking two-dimensional DWT.

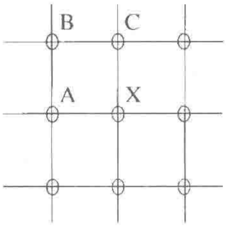


Fig. 6. Differential pulse code modulation encoding of dc band coefficients.

indexes are encoded using the arithmetic encoder in a raster-scan order, starting from the upper left index and ending with the lowest right one. The model is updated with encoding of each bit of the predicted quantization index to adopt the probability model to the statistics of the dc band. The band_offset is subtracted from all the values, and a forward predictive scheme is applied. As shown in Fig. 6, each of the current coefficients w_X is predicted from three other quantized coefficients in its neighborhood, i.e., w_A, w_B, and w_C, and the predicted value is subtracted from the current coefficient

if

$$(|w_{\mathrm{A}} - w_{\mathrm{B}}|) < (|w_{\mathrm{B}} - w_{\mathrm{C}}|)$$
$$w_x = w_{\mathrm{C}}$$

else

$$w_x = w_{\mathrm{A}}$$
$$w_x = w_x - w_x.$$

If any of nodes A, B, or C are not in the image, its value is set to zero for the purpose of the forward prediction.

For wavelet coefficients in all other subbands (i.e., ac subbands), Fig. 7 shows the scanning order for a 16 × 16 image, with three levels of decomposition. The indexes 0–3 represent the dc band coefficients, which are encoded separately. The remaining coefficients are encoded in the order shown in this figure. As an example, indexes 4, 5, $\cdots$, 24 represent one tree. First, coefficients in this tree are encoded starting from index 4 and ending at index 24. Then, the coefficients in the second tree are encoded starting from index 25 and ending at 45. The third tree is encoded starting from index 46 and ending at index 66, and so on.

Fig. 8 shows that the wavelet coefficients are scanned in the subband-by-subband fashion, from the lowest to the highest frequency subbands for a 16 × 16 image with three levels of decomposition. The dc band is located in the upper left corner (with indexes 0–3) and is encoded separately, as described in dc band decoding. The remaining coefficients are encoded in the order shown in the figure, starting from index 4 and ending at index 255.

The zerotree symbols and quantized coefficients are then losslessly encoded using an adaptive arithmetic coder with a given symbol alphabet. The arithmetic encoder adaptively tracks the statistics of the zerotree symbols and encoded values using three models: 1) *type* to encode the zerotree symbols, 2) *magnitude* to encode the values in a bit-plane fashion, and 3) *sign* to encode the sign of the value. For each coefficient, its zerotree symbol is encoded first, and if necessary, then its value is encoded. The value is encoded in two steps. First, its absolute value is encoded in a bit-plane fashion using the appropriate probability model, and then the sign is encoded using a binary probability model, with "0" meaning a positive and "1" meaning a negative sign. The sign model is initialized to the uniform probability distribution.

In EZW, quantization of the wavelet coefficients is done implicitly using successive approximation. When using ZTE, the quantization is explicit and can be made adaptive to scene content. Quantization can be done entirely before ZTE, or it can be integrated into ZTE and performed as the wavelet trees are traversed and the coefficients encoded. If coefficient quantization is performed as the trees are built, then it is possible to dynamically specify a global quantizer step size for each wavelet block, as well as an individual quantizer step size for each coefficient of a block. These quantizers can then be adjusted according to what the coefficients of a particular block represent spatially (scene content), or according to what frequency band the coefficient represents, or both. The

0	1	4	67	5	6	68	69	9	10	13	14	72	73	76	77
2	3	130	193	7	8	70	71	11	12	15	16	74	75	78	79
25	88	46	109	131	132	194	195	17	18	21	22	80	81	84	85
151	214	172	235	133	134	196	197	19	20	23	24	82	83	86	87
26	27	89	90	47	48	110	111	135	136	139	140	198	199	202	203
28	29	91	92	49	50	112	113	137	138	141	142	200	201	204	205
152	153	215	216	173	174	236	237	143	144	147	148	206	207	210	211
154	155	217	218	175	176	238	239	145	146	149	150	208	209	212	213
30	31	34	35	93	94	97	98	51	52	55	56	114	115	118	119
32	33	36	37	95	96	99	100	53	54	57	58	116	117	120	121
38	39	42	43	101	102	105	106	59	60	63	64	122	123	126	127
40	41	44	45	103	104	107	108	61	62	65	66	124	125	128	129
156	157	160	161	219	220	223	224	177	178	181	182	240	241	244	245
158	159	162	163	221	222	225	226	179	180	183	184	242	243	246	247
164	165	168	169	227	228	231	232	185	186	189	190	248	249	252	253
166	167	170	171	229	230	233	234	187	188	191	192	250	251	254	255

Fig. 7. The tree-depth scanning order in ZTE encoding.

advantages of incorporating quantization into ZTE are the following.

1) The status of the encoding process and bit usage are available to the quantizer for adaptation purposes.
2) By quantizing coefficients as the wavelet trees are traversed, information such as spatial location and frequency band is available to the quantizer for it to adapt accordingly and thus provide content-based coding.

C. Multiscale Zerotree Wavelet Entropy (MZTE) Coding

The MZTE coding technique is based on ZTE coding [13], [14], but it utilizes a new framework to improve and extend the ZTE method to achieve a fully scalable yet very efficient coding technique. In this scheme, the low-low band is separately encoded. To achieve a wide range of scalability levels efficiently as needed by the application, the other bands are encoded using the multiscale zerotree entropy coding scheme. This multiscale scheme provides a very flexible approach to support the right tradeoff between layers and types of scalability, complexity, and coding efficiency for any multimedia application. Fig. 9 shows the concept of this technique.

The wavelet coefficients of the first spatial (and/or quality) layer first are quantized with the quantizer Q0. These quantized coefficients are scanned using the zerotree concept, and then the significant maps and quantized coefficients are entropy coded. The output of the entropy coder at this level, BS0, is the first portion of the bitstream. The quantized wavelet coefficients of the first layer are also reconstructed and subtracted from the original wavelet coefficients. These residual wavelet coefficients are fed into the second stage of the coder, in which the wavelet coefficients are quantized with Q1, zerotree scanned, and entropy coded. The output of this stage, BS1, is the second portion of the output bitstream. The quantized coefficients of the second stage are also reconstructed and subtracted from the original coefficients. As shown in Fig. 9, $N+1$ stages of the scheme provide $N+1$ layers of scalability. Each level represents one layer of SNR quality, spatial scalability, or a combination of both.

In MZTE, the wavelet coefficients are quantized by a uniform and midstep quantizer with a dead zone equal to the quantization step size as closely as possible at each scalability layer. Each quality layer and/or spatial layer has a quantization (Q) value associated with it. Each spatial layer has a corresponding sequence of these Q values. The quantization of coefficients is performed in three steps: 1) construction of initial quantization value sequence from input parameters, 2) revision of the quantization sequence, and 3) quantization of the coefficients.

0	1	4	7	16	17	28	29	64	65	68	69	112	113	116	117
2	3	10	13	18	19	30	31	66	67	70	71	114	115	118	119
5	8	6	9	40	41	52	53	72	73	76	77	120	121	124	125
11	14	12	15	42	43	54	55	74	75	78	79	122	123	126	127
20	21	32	33	24	25	36	37	160	161	164	165	208	209	212	213
22	23	34	35	26	27	38	39	162	163	166	167	210	211	214	215
44	45	56	57	48	49	60	61	168	169	172	173	216	217	220	221
46	47	58	59	50	51	62	63	170	171	174	175	218	219	222	223
80	81	84	85	128	129	132	133	96	97	100	101	144	145	148	149
82	83	86	87	130	131	134	135	98	99	102	103	146	147	150	151
88	89	92	93	136	137	140	141	104	105	108	109	152	153	156	157
90	91	94	95	138	139	142	143	106	107	110	111	154	155	158	159
176	177	180	181	224	225	228	229	192	193	196	197	240	241	244	245
178	179	182	183	226	227	230	231	194	195	198	199	242	243	246	247
184	185	188	189	232	233	236	237	200	201	204	205	248	249	252	253
186	187	190	191	234	235	238	239	202	203	206	207	250	251	254	255

Fig. 8. The band-by-band scanning in ZTE encoding.

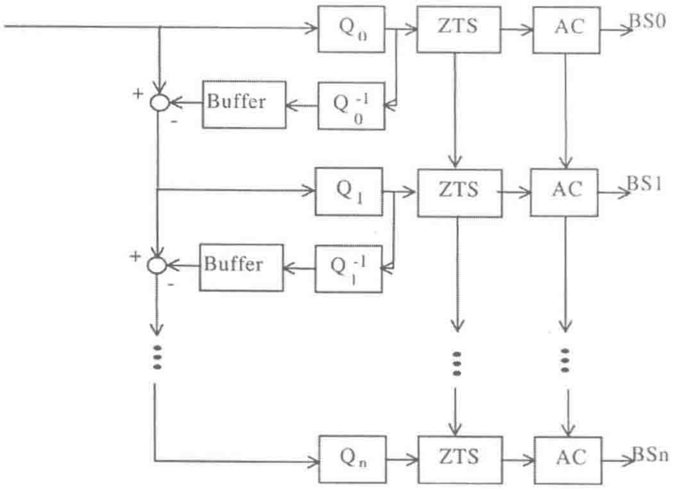


Fig. 9. MZTE encoding structure.

Let n be the total number of spatial layers and $k(i)$ be the number of quality layers associated with spatial layer i. We define the total number of scalability layers associated with spatial layer i, $L(i)$ as the sum of all the quality layers from that spatial layer and all higher spatial layers

$$L(i) = k(i) + k(i+1) + \cdots + k(n).$$

Let $Q(m,\, n)$ be the Q value corresponding to spatial layer m and quality layer n. The quantization sequence (or Q sequence) associated with spatial layer i is defined as the sequence of the Q values from all the quality layers from the ith spatial layer and all higher spatial layers ordered by increasing quality layer and then increasing spatial layer

$$\begin{aligned} Q\_i &= [Q\_i(0),\, Q\_i(1),\, \cdots,\, Q\_i(m)] \\ &= [Q(i,\, 1),\, Q(i,\, 2),\, \cdots,\, Q(i,\, k(i)),\, Q(i+1,\, 1) \\ &\quad \cdot Q(i+1,\, 2),\, \cdots,\, Q(i+1,\, k(i+1)),\, \cdots, \\ &\quad Q(n,\, 1),\, Q(n,\, 2),\, \cdots,\, Q(n,\, k(n))]. \end{aligned}$$

The sequence Q_i represents the procedure for successive refinement of the wavelet coefficients, which are first quantized in the spatial layer i. To make this successive refinement efficient, the sequence Q_i is revised before starting the quantization. Let $Q\_i(j)$ denote the jth value of the quantization sequence Q_i. Consider the case when $Q\_i(j) = pQ\_i(j+1)$. If p is an integer number greater than one, each quantized coefficient of layer j is efficiently refined at layer $(j = 1)$ as each quantization step size $Q\_i(j)$ is further divided into p equal partitions in layer $(j+1)$. If p is greater than one, but not an integer, the partitioning of the $j+1$ layer will not be uniform. This is due to the fact that $Q\_i(j)$ corresponds to quantization levels, which cover $Q\_i(j)$ possible coefficient

TABLE I
REVISION OF THE QUANTIZATION SEQUENCE

Condition on p = **Q_i**(j)/**Q_i**(j+1)	**Revision Procedure**
p < 1.5	**QR_i**(j+1) = **Q_i** (j) (no quantization at layer j+1)
p >= 1.5 p is integer	**QR_i**(j+1) = **Q_i**(j+1) (no revision)

p >= 1.5
p is non-integer
	q = round (**Q_i**(j)/**Q_i**(j+1);
	QR_i(j+1)= ceil (**Q_i**(j)/ q);

values that cannot be evenly divided into $Q\_i(j+1)$ partitions. In this case, $Q\_i(j+1)$ is revised such as to be as close to an integer factor of $Q\_i(j)$ as possible. The last case is when $Q\_i(j+1) >= Q\_i(j)$. In this case, no further refinement can be obtained at the $(j+1)$th scalability layer over the jth layer, so we simply revised $Q\_i(j+1)$ to be $Q\_i(j)$. The revised quantization sequence is referred to as QR_i. Table I summarizes the revision procedure.

Next, we categorize the coefficients in the image in terms of the order of spatial layers. We define these categories as

$$S(i) = \{\text{all coefficients that } \textit{first} \text{ appear in spatial layer } i\}$$

and

$$T(i) = \{\text{all coefficients that appear in spatial layer } i\}.$$

Since once a coefficient appears in a spatial layer it appears in all higher spatial layers, we have the relationship

$$T(1) \subset T(2) \subset \cdots \subset T(n-1) \subset T(n). \tag{1}$$

To quantize each coefficient in $S(i)$, we use the Q values in the revised quantization sequence QR_i. These Q values are positive integers, and they represent the range of values that a quantization level spans at that scalability layer. For the initial quantization we simply divide the value by the Q value for the first scalability layer. This gives us our initial quantization level (note that it also gives us a double-sized dead zone). For successive scalability layers, we need only send the information that represents the refinement of the quantizer. The refinement information values are called residuals and are the index of the new quantization level within the old level where the original coefficient value lies.

We then partition the inverse range of the quantized value from the previous scalability layer in such a way that makes the partitions *as uniform as possible* based on the previously calculated number of refinement levels m. This partitioning always leaves a discrepancy of zero between the partition sizes if the previous Q value is evenly divisible by the current Q value (e.g., previous $Q = 25$ and current $Q = 5$). If the previous Q value is not evenly divisible by the current Q value (e.g., previous $Q = 25$ and current $Q = 10$), then we have a maximum discrepancy of one between partitions. The larger partitions are always the ones closer to zero.

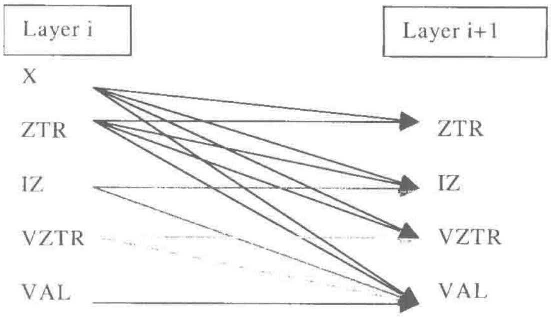


Fig. 10. Zerotree mapping from one scalability layer to the next.

We then number the partitions. The residual index is simply the number of the partition where the original (which is not quantized) value actually lies. We have the following two cases for this numbering.

Case I: If the previous quality level quantized to zero (that is, the value was in the dead zone), then the residual has to be one of the $2m-1$ values in $\{-m, \cdots, 0, \cdots, +m\}$.

Case II: If the previous quality level quantized to a nonzero value, then (since the sign is already known at the inverse quantizer) the residual has to be one of the m values in $\{0, \cdots, m-1\}$.

The restriction of the possible values of the residuals is based solely on the relationship between successive quantization values. Whether the value was quantized to zero in the last scalability pass (both of these facts are known at the decoder) is one reason why using two probability models (one for the first and one for the second case) increases coding efficiency.

For the inverse quantization, we map the quantization level (at the current quality layer) to the midpoint of its inverse range. Thus, we get a maximum quantization error of one-half the inverse range of the quantization level to which we dequantize. One can reconstruct the quantization levels given the list of Q values (associated with each quality layer), the initial quantization value, and the residuals. At the first scalability layer, the zerotree symbols and the corresponding values are encoded for the wavelet coefficients of that scalability layer. The zerotree symbols are generated in the same way as in the ZTE method. For the next scalability layers, the zerotree map is updated along with the corresponding value refinements. In each scalability layer, a new zerotree symbol is encoded for a coefficient only if it was encoded as ZTR, VZTR, or IZ in the previous scalability layer. If the coefficient was decoded as VAL in the previous layer, a VAL symbol is also assigned to it at the current layer, and only its refinement value is encoded from bitstream.

In MZTE, the wavelet coefficients are scanned in either the tree-depth scanning for each scalability layer or in the subband-by-subband fashion, from the lowest to the highest frequency subbands (as shown in Fig. 8). At the first scalability layer, the zerotree symbols and the corresponding values are encoded for the wavelet coefficients of that scalability layer. For the next scalability layers, the zerotree map is updated along with the corresponding value refinements. In each scalability layer, a new zerotree symbol is encoded for a coefficient only if it was decoded as ZTR or IZ in the previous

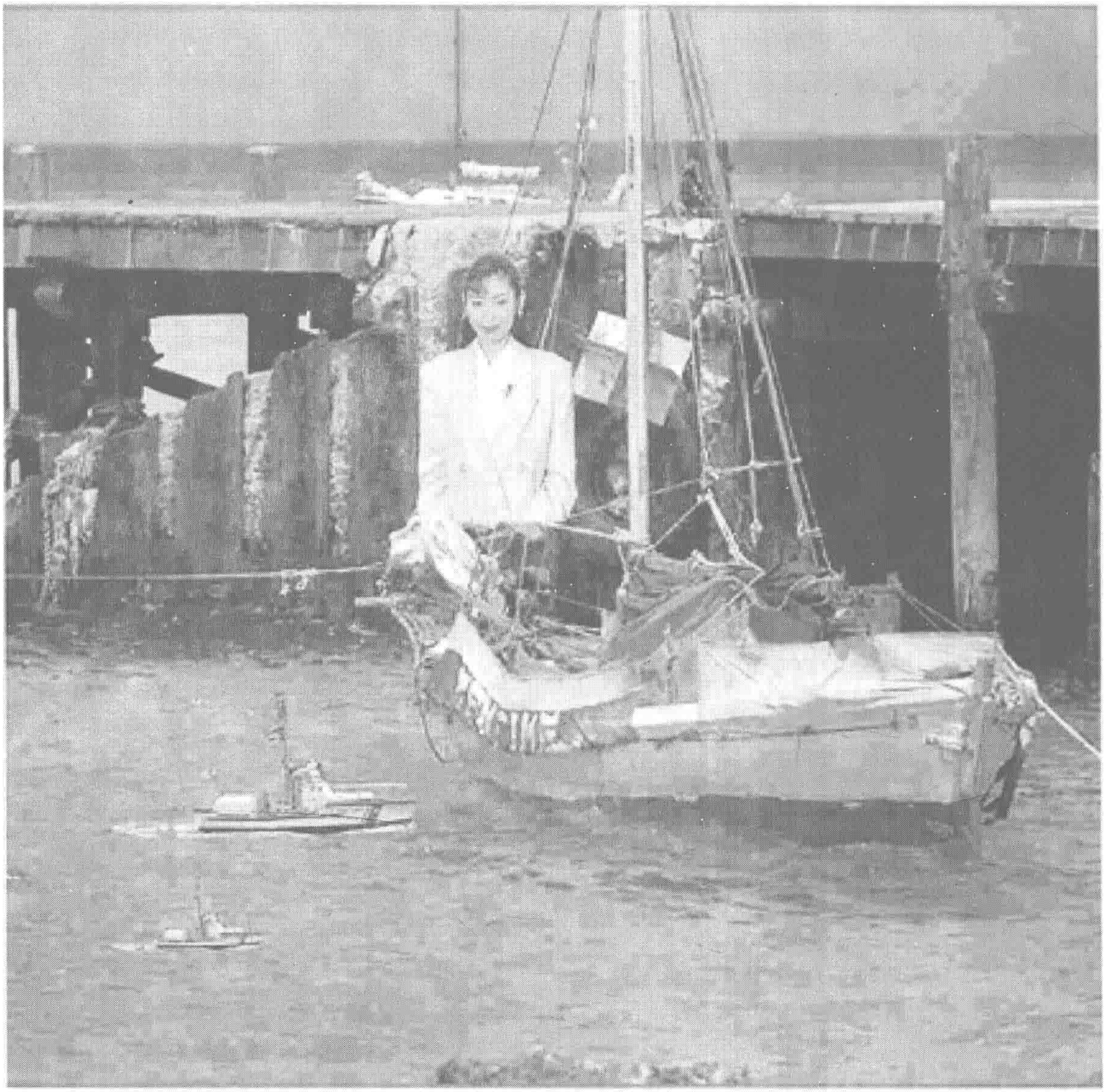


Fig. 11. A synthetic image is compressed and decompressed by JPEG.

scalability layer. If the coefficient was decoded as VAL in the previous layer, a VAL symbol is also assigned to it at the current layer, and only its refinement value is decoded from bitstream.

Fig. 10 shows the relationship between the symbols of one layer and the next layer. If the node is not coded before (shown as "x" in the figure), it can be encoded using any of four symbols. If it is ZTR in one layer, it can remain ZTR or be any of the other three symbols in the next layer. If it is detected as IZ, it can remain IZ or only become VAL. If it is VZTR, it can remain VZTR or become VAL. Last, once it is assigned VAL, it always stays VAL, and no symbol is transmitted in this care.

D. Entropy Coding in EZW, ZTE, and MZTE

Symbols and quantized coefficient values generated by the zerotree stage are all encoded using an adaptive arithmetic coder, such as presented [15]. The arithmetic coder is run over several data sets simultaneously. A separate model with an associated alphabet is used for each. The arithmetic coder uses adaptive models to track the statistics of each set of input data, then encodes each set close to its entropy. The symbols encoded differ based upon whether EZW, ZTE, or MZTE coding is used. For EZW, a four-symbol alphabet is used for the significance map, and a different two-symbol alphabet is used for the SAQ information. The arithmetic coder is restarted every time a new significance map is encoded or a new bit plane is encoded by SAQ. For ZTE, symbols describing node type (zerotree root, valued zerotree root, value, or isolated zero) are encoded. The list of nonzero quantized coefficients that correspond one-to-one with the valued zerotree root or value symbols are encoded using an alphabet that does not include zero. The remaining coefficients, which correspond one-to-one to the value symbols, are encoded using an alphabet that does include zero. For any node reached in a scan that is a leaf with no children, neither root symbol can apply. Therefore,

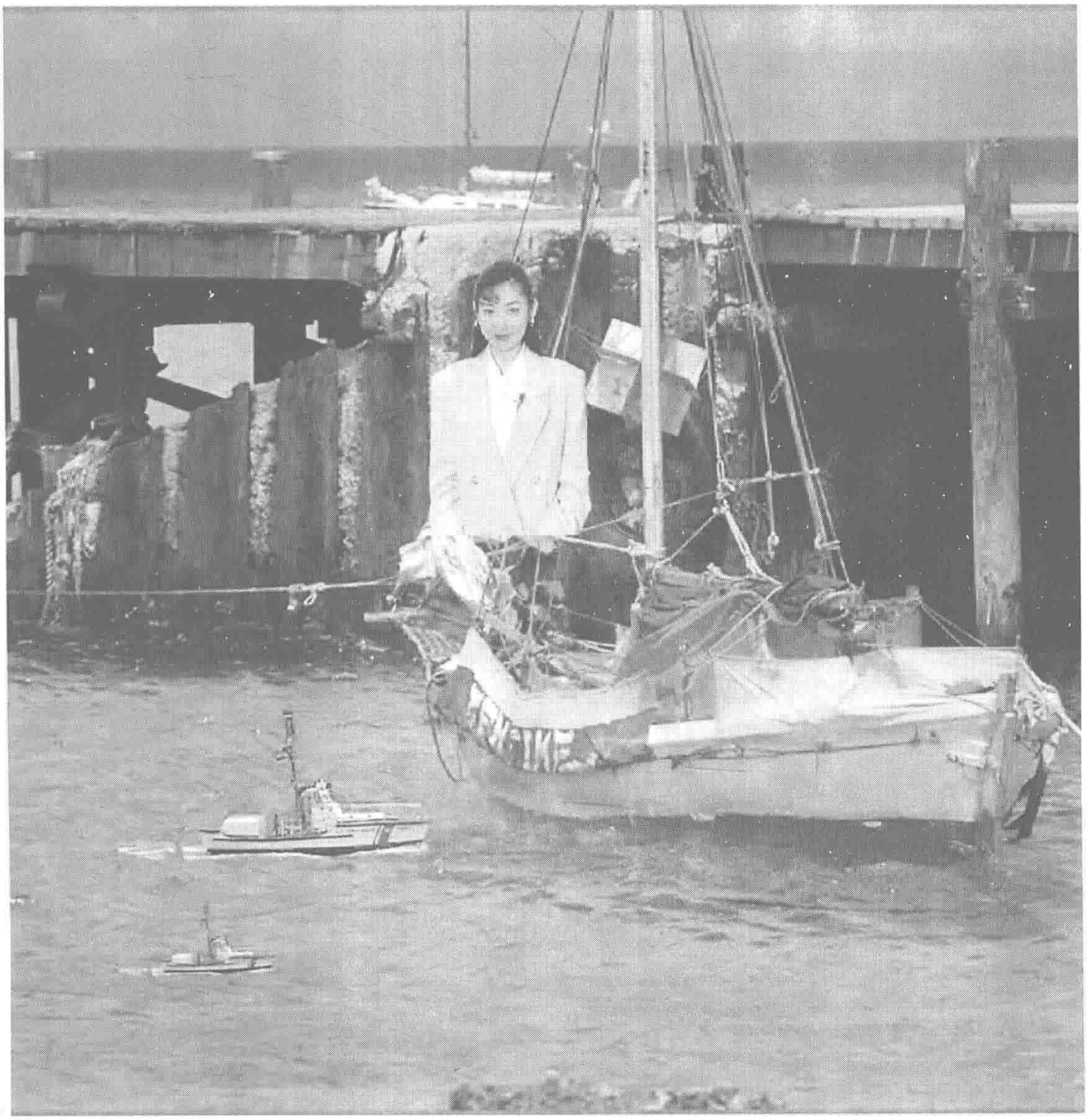


Fig. 12. A synthetic image is compressed and decompressed by MZTE.

bits are saved by not encoding any symbol for this node and encoding the coefficient along with those corresponding to the value symbol using the alphabet that includes zero.

In MZTE, one additional probability model, *residual,* is used for encoding the refinements of the coefficients that were encoded with the VAL or VZTR symbol in any previous scalability layers. If in the previous layer a VAL symbol was assigned to a node, the same symbol is kept for the current pass and no zerotree symbol is encoded. The residual model, same as the other probability models, is also initialized to the uniform probability distribution at the beginning of each scalability layer. The numbers of bins for the *residual* model is calculated based on the ratio of the quantization step sizes of the current and previous scalability. When a residual model is used, only the magnitude of the refinements are encoded, as these values are always zero or positive integers. Furthermore, to utilize the high correlation of zerotree symbol between scalability layers, a context modeling, based on the zerotree symbol of the coefficient in the previous scalability layer in MZTE, is used to better estimate the distribution of zerotree symbols. In MZTE, only INIT and LEAF_INIT are used for the first scalability layer for the nonleaf subbands and leaf subbands, respectively. Subsequent scalability layers in the MZTE use the context associated with the symbols. The different zerotree symbol models and their possible values are summarized in Table II.

If a spatial layer is added, then the contexts of all previous leaf subband coefficients are switched into the corresponding nonleaf contexts. The coefficients in the newly added subbands use the LEAF_INIT context initially.

III. Test Results

Sarnoff's MZTE algorithm has been tested and verified throughout the MPEG-4 core experiment process, with many iterative refinements and support from other partners, such as Sharp, TI, Vector Vision, Rockville, Lehigh, Oki, and Sony.

TABLE II
ZEROTREE SYMBOL MODELS

Context for Nonleaf subbands	Possible values
INIT	ZTR(2), IZ(0), VZTR(3), VAL(1)
ZTR	ZTR(2), IZ(0), VZTR(3), VAL(1)
ZTR DESCENDENT	ZTR(2)
IZ	IZ(0), VAL(1)

Context for Leaf subbands	Possible values
LEAF_INIT	ZTR(0), VZTR(1)
LEAF_ZTR	ZTR(0), VZTR(1)
LEAF_ZTR_DESCENDENT	ZTR(0), VZTR(1)

TABLE III
PSNR VALUES

Compression scheme	PSNR-Y	PSNR-U	PSNR-Y
DCT-based JPEG	28.36	34.74	34.98
Wavelet-based MZTE	30.98	41.68	40.14

This section presents a small subset of the representative results of MZTE in comparison with the JPEG compression, using a hybrid natural and synthetic image. The images in Figs. 11 and 12 are generated by JPEG and MZTE compression schemes, respectively, at the same compression ratio of 45 : 1. The resultant images show that the MZTE scheme generates much better image quality with good preservation of fine texture regions and absence of the blocking effect compared with JPEG. The PSNR values for both reconstructed images are tabulated in Table III.

Fig. 13 demonstrates the spatial and quality scalabilities at different resolutions and bit rates using the MZTE compression scheme. The images in (a) are of size of 128 × 128 are reconstructed by decoding the MZTE bitstream at a bit rate of 80 and 144 kbits, respectively. The two reconstructed images in (b) are of size of 256 × 256 at a bit rate of 192 and 320 kbits, respectively, and the final resolution of 512 × 512 at 750 kbits is shown in (c).

A slight variation of the MZTE algorithm was also submitted in the January 1998 JPEG-2000 meeting. The test group reported its review of the test results and showed that MZTE is one of the top five algorithms that demonstrate the best visual quality with the statistically same compression efficiency among 27 submitted proposals [19]. The JPEG2000 standardization effort is still at early stage, and is scheduled to be defined by 2001.

IV. CONCLUSIONS

In this paper, scalable texture coding is discussed for synthetic and natural hybrid coding applications in the MPEG-4 framework. Spatial and quality scalabilities are two important features desired in many multimedia applications. We have presented three zerotree wavelet algorithms, which provide high compression efficiency as well as scalability of the compressed bitstreams. EZW is a zerotree wavelet algorithm that provides high granularity quality scalability. Zerotree entropy coding was demonstrated with high compression efficiency and spatial scalability. In the ZTE algorithm, quantization is explicit, coefficient scanning is performed in one pass, and tree symbol representation is optimized. The multiscale zerotree entropy coding technique combines the advantages of EZW and ZTE and provides both high compression efficiency and fine-graduality scalabilities in both spatial and SNR domains. MZTE is adopted as the baseline visual texture coding scheme in the MPEG-4 synthetic natural hybrid coding of textural maps and still images.

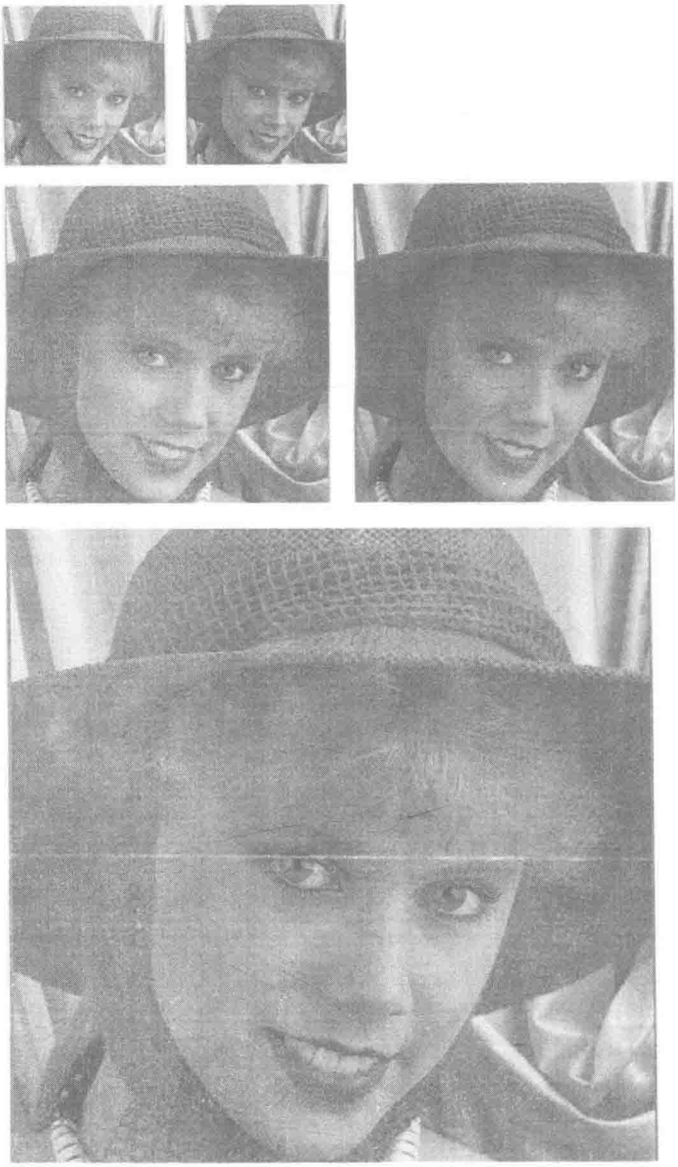


Fig. 13. The spatial and quality scalabilities at different resolutions and bit rates using MZTE.

REFERENCES

[1] Y.-Q. Zhang, F. Pereira, T. Sikora, and C. Reader, Eds., "Special issue on MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Feb. 1997.

[2] "MPEG-4 requirements document v.5," in *Proc. Fribourg MPEG Meeting,* Oct. 1997, Doc. ISO/IEC JTC1/SC29/WG11 W1886.

[3] "Text of ISO/IEC FDIS 14496-2," in *Proc. Atlantic City MPEG Meeting,* Oct. 1998, Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/W2502.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing,* vol. 41, pp. 3445–3462, Dec. 1993.

[5] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[6] A. Arkansi and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets.* New York: Academic, 1996.

[7] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 6, pp. 243–250, June 1996.

[8] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Process.,* vol. 3, pp. 572–588, Sept. 1994.

[9] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint optimization of scalar and tree-structured quantization of wavelet image decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals, Systems, and Computers,* Pacific Grove, CA, Nov. 1993, pp. 891–895.

[10] S. Li and W. Li, "Shape-adaptive wavelet coding," in *Proc. SPIE Visual Communications and Image Processing,* Jan. 1998.

[11] A. Zandi *et al.,* "CREW: Compression with reversible embedded wavelets," in *Proc. IEEE Data Compression Conf.,* Snowbird, UT, Mar. 1995.

[12] S. Servetto, K. Ramchandran, and M. Orchard, "Wavelet based image coding via morphological prediction of significance," in *Proc. Int. Conf. Image Processing,* Washington, DC, 1995.

[13] S. A. Martucci, I. Sodagar, T. Chiang, and Y.-Q. Zhang, "A zerotree wavelet video coder," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 7, pp. 109–118, Feb. 1997.

[14] Sarnoff Corp., "Zero-Tree Wavelet Entropy (ZTE) Coding for MPEG4," ISO/IEC/JTC/SC29/WG11/MPEG95/W512, Nov. 1995

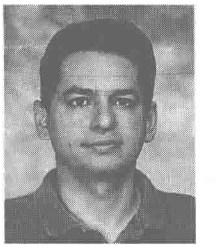
[15] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Commun. ACM,* vol. 30, pp. 520–540, June 1987.

[16] Y.-Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color compression," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 2, Sept. 1992.

[17] Z. Xiong, K. Ramachandran, M. Orchard, and Y.-Q. Zhang, "A comparison study of DCT and wavelet-based coding," in *Proc. ISCAS'97;* see also *IEEE Trans. Circuits Syst. Video Technol.,* to be published.

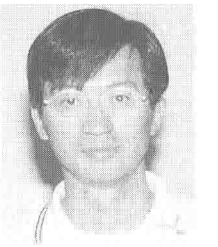
[18] J. Liang, "Highly scalable image coding for multimedia applications," in *Proc. ACM Multimedia,* Seattle, WA, Oct. 1997.

[19] "Requirements and profiles of JPEG-2000," in *Sydney JPEG Meeting,* Nov. 1997, Doc. ISO/IEC JTC1/SC29/WG1 N715.



Iraj Sodagar received the B.S. degree from Tehran University, Tehran, Iran, in 1987 and the M.S. and Ph.D. degrees from the Georgia Institute of Technology (Georgia Tech.), Atlanta, in 1993 and 1994, respectively, all in electrical engineering.

He is Head of the Interactive Media Group at the Multimedia Technology Laboratory, Sarnoff Corp., Princeton, NJ. He was a Research Assistant at the School of Electrical Engineering, Georgia Tech., from 1991 to 1994. At Georgia Tech., his research was focused on multirate filter banks and wavelets and their applications in image coding. In January 1995, he joined Sarnoff Corp., where he has developed low bit-rate image and video compression algorithms for multimedia applications. He has been representing Sarnoff at ANSI NCITS, ISO MPEG-4, and JPEG-2000 standards bodies since 1995. He was the Chair of the MPEG-4 visual texture coding group and Cochair of the MPEG-4 Version 2 core experiments and JPEG-2000 scalability and progressive-to-lossless coding ad hoc groups. His current interests include low bit-rate image and video coding, multimedia content-based representation and indexing, interactive media authoring, MPEG-4, MPEG-7, and JPEG-2000.



Hung-Ju Lee received the B.S. degree from Tatung Institute of Technology, Taipei, Taiwan, in 1987 and the M.S. and Ph.D. degrees from Texas A&M University in 1993 and 1996, respectively, all in computer science.

In 1996, he joined Sarnoff Corp., Princeton, NJ, as a Member of Technical Staff. He actively participates in ISO's MPEG digital video standardization process, with a particular focus on wavelet-based visual texture coding and scalable rate control. His research interests include image and video coding and network resource management for multimedia applications.

Paul Hatrack received the B.S. degree from George Washington University, Washington, DC, in 1987 and the M.S. degree from Rutgers—The State University, New Brunswick, NJ, in 1998, both in electrical engineering. He currently is pursuing the Ph.D. degree in electrical engineering at Rutgers.

He currently is a Member of the Wireless Information Network Laboratory (Winlab) at Rutgers. He is an Associate Member of Technical Staff in the Interactive Media Group, Sarnoff Corp., Princeton, NJ, which he joined in summer 1996. His previous research interests included multiuser detection and power control for CDMA systems and video and image coding. His current research interests include video and image coding, joint source-channel coding, and signal detection and estimation.

Ya-Qin Zhang (S'87–M'90–SM'93–F'97) was born in Taiyuan, China, in 1966. He received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China (USTC) in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from George Washington University, Washington DC, in 1989.

He joined Microsoft Research, Beijing, China, in January 1999. He was Director of the Multimedia Technology Laboratory, Sarnoff Corp., Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). His laboratory is actively engaged in R&D and commercialization of digital television, MPEG, multimedia, and Internet technologies. He is a Cofounder of E-Vue.com, a startup venture that develops image and video compression and communications technologies and products on the Internet. He was with GTE Laboratories, Inc., Waltham, MA, and Contel Technology Center, VA, from 1989 to 1994. He is the author or coauthor of more than 150 refereed papers and 30 U.S. patents granted or pending in image/video compression and communications, multimedia, wireless networking, satellite communications, and medical imaging. Many of the technologies he and his team developed have become the basis for startup ventures, products, and international standards. He is a member of the editorial boards of seven professional journals and more than a dozen conference committees. He has been an active contributor to and participant in ISO/MPEG and ITU standardization efforts.

Dr. Zhang is Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is Chairman of the Visual Signal Processing and Communications Committee of the IEEE Circuits and Systems Society. He has received many awards, including several industry technical achievement awards, the "1997 Research Engineer of the Year" award by the Central Jersey Engineering Council, and "The 1998 Outstanding Young Electrical Engineer" award from Eta Kappa Nu.

Multiresolution Watermarking for Images and Video

Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang

Abstract—**This paper proposes a unified approach to digital watermarking of images and video based on the two- and three-dimensional discrete wavelet transforms. The hierarchical nature of the wavelet representation allows multiresolutional detection of the digital watermark, which is a Gaussian distributed random vector added to all the high-pass bands in the wavelet domain. We show that when subjected to distortion from compression or image halftoning, the corresponding watermark can still be correctly identified at each resolution (excluding the lowest one) in the wavelet domain. Computational savings from such a multiresolution watermarking framework is obvious, especially for the video case.**

***Index Terms*—Copyright protection, multimedia, watermarking, wavelet transforms.**

I. Introduction

With the rapid growth of network distributions of images and video, there is an urgent need for copyright protection against pirating. Different digital watermarking schemes have been proposed to address this issue of ownership identification. Early work on digital watermarking focused on information hiding in the spatial domain. For example, Schyndel *et al.* proposed to insert a watermark by changing the least significant bit of some pixels in an image [1]. Bender *et al.* described a watermarking approach by modifying a statistical property of an image [2]. Recent efforts are mostly based on frequency-domain techniques for still images. In particular, Cox *et al.* described a method where the watermark is embedded in large discrete cosine transform (DCT) coefficients using ideas borrowed from spread spectrum in communications [3], [4]. For digital watermarking of video sequences, Hartung and Girod [6] proposed a watermarking technique for MPEG-2 encoded video in the bitstream domain. Swanson *et al.* also considered MPEG-2 compressed domain video watermarking [7] and a wavelet-based multiresolution video watermarking method [8], in which the multiresolutional wavelet transform is performed in the temporal domain only. Although different transforms (e.g., discrete Fourier transform, discrete cosine transform, and discrete wavelet transform) have been used in digital watermarking schemes reported in the literature, there is no common framework for multiresolutional digital watermarking of both images and video.

In this paper, we propose a unified approach to digital watermarking of images and video based on the two-dimensional (2-D) and three-dimensional (3-D) discrete wavelet transforms [9]. Our wavelet-based watermarking framework is motivated by the fact that most network-based images and video are in compressed form and that wavelets are playing an important role in upcoming compression standards such as JPEG2000 and MPEG-4. We first experimentally show that a watermark signal [e.g., an independently identically distributed (i.i.d.) Gaussian random vector] can be embedded in *every* high-pass wavelet coefficient without any impact on the visual fidelity. This is different from the approach in [3], where the watermark is only placed into a small number of the perceptually most important coefficients (e.g., 1000 largest coefficients). Our results indicate that the capacity or the amount of information in an invisible watermark can be quite large.

We then describe the proposed framework where an i.i.d. Gaussian random vector is added to all the high-pass bands in the wavelet domain as a multiresolutional digital watermark. The watermark added to a lower resolution can be thought of as a nested version of the one corresponding to a higher resolution. The hierarchical nature of the wavelet representation allows detection of watermarks at all resolutions except the lowest one. Detection of lower resolution watermarks reduces computational complexity, as fewer frequency bands are involved. This computational savings can be quite significant for the video case.

The multiresolutional property makes our watermarking scheme robust to image/video downsampling operation by a power of two in either space or time. We also test our proposed watermarking scheme against common distortions introduced by compression and image halftoning. We use state-of-the-art wavelet image and video coders [10], [11] for compression and error diffusion for halftoning [12]. Experiments show that for both cases, the corresponding watermark can still be correctly identified at each possible resolution in the wavelet domain.

II. Capacity Issues in Digital Image Watermarking

Digital watermarking is a process of hiding a watermark (or signature) signal in image or video media by making small changes in the media content. Properties of watermarks include unobstructiveness and robustness. The former indicates that a watermark should be perceptually invisible; the later means that the watermark should be difficult to remove or destroy before resulting in severe degradation in visual fidelity. To make the watermark invisible, one would intuitively pick a watermark signal with small energy and hide it in the perceptually insignificant regions. However, the main thrust of [4] is the placement of the watermark in the perceptually significant regions of an image for robustness. It is argued that visual fidelity is only preserved if the perceptually significant regions

Manuscript received March 25, 1998; revised December 30, 1998. This paper was recommended by Associate Editor C. W. Chen.

W. Zhu is with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA.

Z. Xiong is with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA.

Y.-Q. Zhang is with Microsoft Research, Beijing 100080 China.

Publisher Item Identifier S 1051-8215(99)02961-4.