BINARY SHAPE MASK REPRESENTATION FOR ZEROTREE-BASED VISUAL OBJECT CODING

Karl Martin, Rastislav Lukac, and Konstantinos N. Plataniotis
Bell Canada Multimedia Lab, The Edward S. Rogers Sr. Department of ECE, University of Toronto
kmartin@dsp.utoronto.ca, lukacr@ieee.org, kostas@dsp.utoronto.ca

Abstract

A new technique for representing the binary shape mask of an arbitrarily-shaped visual object is presented. By performing the shape-adaptive discrete wavelet transform (SA-DWT) on the object using a globally uniform subsampling policy, the shape mask is naturally decomposed via the "Lazy wavelet transform." The resulting shape mask coefficients are arranged into spatial orientation trees for the purpose of parallel coding with the object texture in a zerotree-based scheme. This arrangement provides an excellent means for the prediction of uncoded shape pixels, thus lending itself to highly efficient coding.

Keywords: *object-based coding, shape coding, spatial orientation tree*

1. INTRODUCTION

Coding of arbitrarily-shaped visual objects has been increasing in importance in recent times. Object-based coding is the basis for many emerging applications, such as very low bit-rate imagery. Several wavelet-based visual object coding schemes have been described in the literature, including ones that involve modifications of the popular Embedded Zerotree Wavelet coder (EZW) and Set Partitioning in Hierarchical Trees coder (SPIHT) [1–5]. However, most object-based coding schemes only address coding of texture and require that the object shape mask, represented as a binary image, be: a) coded separately using a binary shape mask coding scheme; and b) fully decoded *prior* to the decoding of the texture. These restrictions can negatively impact computational overhead and the quality of very low bit-rate reconstructions.

A new way of representing the binary shape mask of a visual object is presented here. The proposed methodology is designed to allow the shape mask to be coded in conjunction with the texture in a wavelet-based, zerotree coding scheme by decomposing it and arranging the coefficients in spatial orientation trees in tandem with the texture coefficients. This has the advantage of having only to implement one algorithm which enables the simultaneous coding of shape and texture, as well as the possibility of very low bit-rate object decoding with lossy shape reconstruction.

2. BINARY SHAPE MASK DECOMPOSITION USING "LAZY WAVELET TRANSFORM"

An arbitrarily-shaped object consists of a shape mask S, and texture T. Given the object has a bounding box of size $M \times N$, the shape mask S is an $M \times N$ matrix with elements $S(i,j) \in \{0,1\}$, for $i = 0,1,\ldots,M-1$ and $j = 0,1,\ldots,N-1$. The case $S(i,j) = 1$ denotes that the location (i,j) is *inside* the object and that there is a texture component (pixel) $T(i,j)$ associated with the location. Otherwise, $S(i,j) = 0$ denoting that the location (i,j) is *outside* the object and $T(i,j) = \{\emptyset\}$.

The shape-adaptive discrete wavelet transform (SA-DWT) [3] provides a compact and efficient method for decomposing the texture of an arbitrarily-shaped visual object. It is well defined for all types of wavelet filters and allows for a choice of even or odd subsampling for each local image segment.

In performing the first level SA-DWT decomposition of an object, the LL1, LH1, HL1 and HH1 subbands are created with associated texture $T_{\mathrm{LL}1}$, $T_{\mathrm{LH}1}$, $T_{\mathrm{HL}1}$ and $T_{\mathrm{HH}1}$ and associated subband mask $S_{\mathrm{LL}1}$, $S_{\mathrm{LH}1}$, $S_{\mathrm{HL}1}$ and $S_{\mathrm{HH}1}$ respectively. Assuming that X is either LL, LH, HL, or HH, each location within the subband X1 is denoted $(i_{\mathrm{X}1}, j_{\mathrm{X}1})$. The subband texture $T_{\mathrm{X}1}$ is similar to a standard rectangular discrete wavelet transform (DWT) subband, except that the value $S_{\mathrm{X}1}(i_{\mathrm{X}1}, j_{\mathrm{X}1})$ is used to denote whether or not $T_{\mathrm{X}1}(i_{\mathrm{X}1}, j_{\mathrm{X}1})$ has a valid value.

To perform the SA-DWT, a subsampling policy should be chosen — either a *local* policy which ensures that each individual image segment in T decomposes to have a low-pass component with length equal to or greater than the high-pass component, or a *global* policy which ensures that the relative phase of all components is maintained.

In [5], the prior decoding of the object shape allows for the determination of the texture decoder execution path by way of the initial discarding of "empty" spatial orientation trees. For the proposed scenario of parallel shape and texture coding, no *a priori* shape information is provided to the decoder and thus a global, shape-independent SA-DWT subsampling policy is required to ensure a well defined execution path. As such, the *in-place lifting* DWT implementa-

CCECE 2004 - CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00 © 2004 IEEE

tion [6] is chosen to transform the 1-D image segments since it has the natural quality of a global subsampling policy, as well as being both computationally and memory efficient.

The output after performing one level of the transform is the high-pass/low-pass interleaved arrangement shown in Fig. 1. Globally, all even positioned coefficients are low-pass components and all odd coefficients are high-pass components (in the vertical or horizontal direction). An important consequence is that subband masks $S_{\mathrm{X}\,1}$ do not need to be generated beforehand to determine the placement of the subband coefficients $T_{\mathrm{X}\,1}$ — i.e., by performing the DWT operation *in-place*, the subband coefficients are placed in a pseudo-spatial domain interleaved arrangement T_1 using the original shape mask S as a reference. To extract each of the subbands $T_{\mathrm{X}\,1}$ from T_1, a simple subsampling rule is used. The extraction procedure corresponding to $T_{\mathrm{LL}1}$ is defined as follows:

$$(i,j) \leftarrow (2i_{\mathrm{LL}1}, 2j_{\mathrm{LL}1}) \tag{1}$$

thus mapping the LL1 subband coordinates $(i_{\mathrm{LL}1}, j_{\mathrm{LL}1})$ to the interleaved subband coordinates (i,j). Similarly, $T_{\mathrm{HL}1}$, $T_{\mathrm{LH}1}$ and $T_{\mathrm{HH}1}$ are extracted by:

$$(i,j) \leftarrow (2i_{\mathrm{HL}1} \quad 1, 2j_{\mathrm{HL}1}), \tag{2}$$
$$(i,j) \leftarrow (2i_{\mathrm{LH}1}, 2j_{\mathrm{LH}1} \quad 1), \tag{3}$$
$$(i,j) \leftarrow (2i_{\mathrm{HH}1} \quad 1, 2j_{\mathrm{HH}1} \quad 1). \tag{4}$$

This coordinate mapping represents a subsampling operation where even or odd subsampling is used in the horizontal and vertical directions to extract a specific first level subband. The actual subband extraction is performed by the assignment

$$T_{\mathrm{X}\,1}(i_{\mathrm{X}\,1}, j_{\mathrm{X}\,1}) \leftarrow T_1(i,j), \tag{5}$$

used in conjunction with the coordinate mappings of (1)–(4). This is the same mapping used to extract the subbands of a standard DWT (i.e., not shape-adaptive) when in-place lifting is used [6]. The extra significance in this scenario is that the mapping also allows for the extraction of the subband masks from the original *spatial domain* shape mask S. Specifically,

$$S_{\mathrm{X}\,1}(i_{\mathrm{X}\,1}, j_{\mathrm{X}\,1}) \leftarrow S(i,j). \tag{6}$$

Level k 1 of the SA-DWT decomposition may be performed by repeating the procedure, treating T_{LL} and S_{LL} as new input. These higher level subbands can be extracted from the interleaved arrangement by iteratively applying the subsampling operations of (1)–(4).

Since (1)–(4) describe a *generic* subsampling rule independent of the data, the pseudo-spatial domain arrangement of T allows for unambiguous positioning of all coefficients relative to S. However, it also provides insight into an alternative representation of the shape mask. By iteratively applying (1)–(4) to extract each S_{X} from S, the "Lazy wavelet transform" (LWT) [6] is being performed on S. The LWT, being the initial subsampling step of the lifting DWT implementation, does not in fact decorrelate textural data, but it does provide a basic multiresolution decomposition. This representation may typically only be viewed as a tool for determining which coordinates $(i_{\mathrm{X}}\ , j_{\mathrm{X}}\ )$ in T_{X} are inside or outside the object, but it can also be used for coding of the shape. We propose that the LWT decomposition of S into S_{X} be used so that the shape may be coded in conjunction with the decomposed textural data T_{X} .

3. SHAPE MASK CODING USING SPATIAL ORIENTATION TREES

To code the object shape mask S, the subband mask coefficients $S_{\mathrm{X}}\ (i_{\mathrm{X}}\ , j_{\mathrm{X}}\ )$ are arranged into spatial orientation trees (as in the SPIHT algorithm [2]). This allows a zerotree-based algorithm to be used to code the shape information concurrently with the texture coefficients. As a result of the one-to-one mapping of $S \rightarrow S_{\mathrm{X}}$, coding of the values in S_{X} corresponds to a direct coding of S. This is in contrast with coding of the texture coefficients, where each pixel in the object texture T requires the contribution of several coefficients from the subbands T_{X} .

The creation of the spatial orientation trees maps the coordinates $(i_{\mathrm{X}}\ , j_{\mathrm{X}}\ )$ to a particular node in a tree. Denoting ζ_m as the set of elements in spatial orientation tree m, the specification of whether $S_{\mathrm{X}}\ (i_{\mathrm{X}}\ , j_{\mathrm{X}}\ ) \in \zeta_m$ directly translates to a specification of whether $S(i,j) \in \zeta_m$. Thus, a spatial domain map of ζ_m can be made.

Given a K level wavelet decomposition, a particular subband mask element $S_{\mathrm{X}}\ (i_{\mathrm{X}}\ , j_{\mathrm{X}}\ ) \in \zeta_m$ will have four children nodes also in ζ_m, if $1 < k \leq K$. Using the spatial orientation tree specification described in [2], all the children of S_{X} will be from the subband mask $S_{\mathrm{X}\,(\ -1)}$ (where $\mathrm{X} \neq \mathrm{LL}$). For example, a tree node derived from subband mask $S_{\mathrm{HL}2}$ will have children only from $S_{\mathrm{HL}1}$ (the directions HL, LH and HH are never mixed within a tree). Specifically, the four children of $S_{\mathrm{X}}\ (i_{\mathrm{X}}\ , j_{\mathrm{X}}\ )$ are

$$S_{\mathrm{X}\,(\ -1)}(2i_{\mathrm{X}}\ , 2j_{\mathrm{X}}\ ),$$
$$S_{\mathrm{X}\,(\ -1)}(2i_{\mathrm{X}} \quad 1, 2j_{\mathrm{X}}\ ),$$
$$S_{\mathrm{X}\,(\ -1)}(2i_{\mathrm{X}}\ , 2j_{\mathrm{X}} \quad 1),$$
$$S_{\mathrm{X}\,(\ -1)}(2i_{\mathrm{X}} \quad 1, 2j_{\mathrm{X}} \quad 1).$$

Accordingly, each element in $S_{\mathrm{X}\,(\ -1)}$ will have its own children if $k > 2$. The top level nodes of all trees are coefficients from $S_{\mathrm{LL}K}$ resulting in the trees having K 1 levels.

Because S_{HL} , S_{LH} and S_{HH} are segregated into separate trees, there are always three trees which map to a square block of pixels, as shown in Fig. 2. With darker pixels representing nodes higher in the trees, the multiresolution struc-

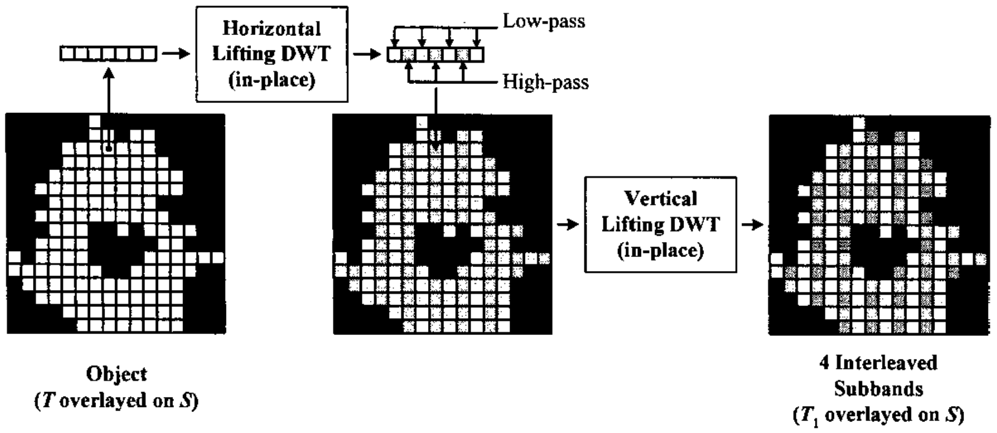
Horizontal Lifting DWT (in-place)
Low-pass
High-pass
Vertical Lifting DWT (in-place)
Object (T overlayed on S)
4 Interleaved Subbands (T_1 overlayed on S)

Fig. 1. One-level, two-dimensional SA-DWT using the in-place lifting DWT implementation. The input is the object with the texture T overlayed on the shape mask S. The output is the interleaved texture subband arrangement T_1 overlayed on S.

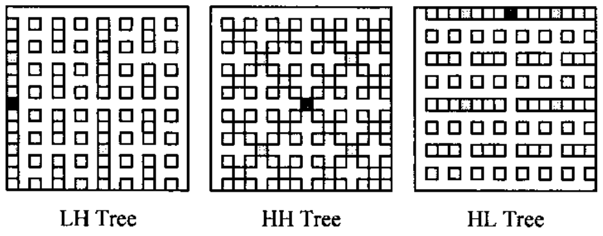

Fig. 2. Three spatial orientation trees map to cover block in spatial domain. Darker pixels represent higher level nodes in the trees.

ture of the trees is evident. The subtrees of a spatial orientation tree map to smaller spatial blocks allowing finer detail to be coded as the tree is traversed from top to bottom.

With the operation $S \rightarrow S_X \rightarrow \zeta_m$, coding of ζ_m results in a multiresolution coding of S. Using the zerotree coding paradigm [1,2], if the situation

$$S_X(i_X, j_X) = 0, \quad \forall S_X(i_X, j_X) \in \zeta_m \qquad (7)$$

or

$$S_X(i_X, j_X) = 1, \quad \forall S_X(i_X, j_X) \in \zeta_m \qquad (8)$$

occurs, the entire spatial orientation tree can be coded with little overhead. It is expected that this will occur frequently since the elements of ζ_m are derived from a compact spatial block. However, this will not occur when ζ_m spans the edge of the object shape. In these cases, ζ_m contains elements with different values and must be partitioned into subtrees to be tested in the same manner as in (7) and (8). This iterative procedure is similar to finding zerotrees in texture coding. Coarse detail in the shape will require few bits to code, whereas finer detail will require the partitioning of trees and subsequently more bits to code. Lossy coding can be achieved by not coding lower level subtrees and predicting their values.

4. EXPERIMENTAL RESULTS

A generalized lossy shape coding scheme using the spatial orientation tree arrangement can be simulated by assuming that lower nodes in the trees are not coded and subsequently having their values predicted. For the experiments, a simple prediction scheme was used in which an uncoded node value is set to be equal to the value of its parent. If its parent is not coded either, then it is set to its grandparent's value, and so on. A predicted node value represents a predicted shape mask pixel and so the prediction can either be correct (resulting in no actual loss) or incorrect (resulting in the inversion of a shape mask pixel).

First, all the nodes derived from S_{X1} (the first level subbands) were discarded and their values predicted. This represents 75% of the total pixels from S being unknown. The amount of uncoded information was then increased by discarding more levels from the bottom of the trees upward (i.e., S_{X2}, then S_{X3}, and so on).

Fig. 3a) shows the original 12×12 binary shape mask and Fig. 3b) depicts the lossy shape mask reconstructed after discarding and predicting the S_{X1} elements from all trees. The reconstruction error manifests itself as "fuzziness" around the shape edges. This is due to the fact that errors occur when the true values of discarded nodes are not equal to their respective parents' values (i.e., for a mixed tree). This will generally occur along detailed shape edges.

Figs. 3c)–e) show the results derived from discarding the bottom 2 to 4 levels of the spatial orientation trees respec-

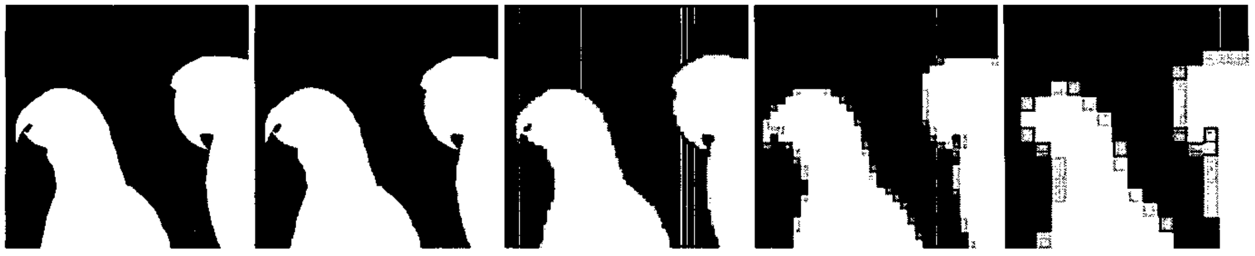

Fig. 3. Image (a) is the original shape mask, whereas (b)–(e) are the reconstructed shape masks based upon predicting the bottom 1 to 4 levels of the spatial orientation trees, respectively.

Table 1. Summary of simulated reconstruction results

Tree nodes predicted	% of nodes predicted	% error in reconstruction
S_{X1}	75%	0.48%
S_{X1}, S_{X2}	93.75%	1.36%
S_{X1}, S_{X2}, S_{X3}	98.44%	2.85%
$S_{X1}, S_{X2}, S_{X3}, S_{X4}$	99.61%	6.42%

tively. Again, the errors are manifested as increased fuzziness and blocking artifacts around the shape edges. Table 1 summarizes the corresponding numerical results. The loss in the reconstruction is described as the percentage of shape mask pixels which are incorrect. Due to the size of the subbands in a wavelet decomposition, each time another level is discarded from the bottom of the spatial orientation trees, the number of *coded* nodes is reduced by 75%. Thus, the percentage of coded and uncoded nodes is independent of the shape mask.

Discarding all the nodes derived from elements of S_{X1}, S_{X2}, S_{X3}, and S_{X4}, results in only 0.39% of the pixels in the shape mask being coded and the remaining 99.61% being predicted. Even in such a drastic situation and with such a simple prediction scheme, the percentage of shape mask pixels that were incorrect is only 6.42%. Clearly, the LWT decomposed shape mask arranged in the spatial orientation trees provides an excellent means for the prediction of unknown values and thus lends itself to highly efficient coding. The benefit of this method of coding is that it can be done in conjunction with zerotree coding of texture coefficients for combined embedded coding of shape and texture.

5. CONCLUSION

This paper introduced a new way of representing the shape mask of a visual object so that it can be coded in conjunction with the object texture using a zerotree-based coding scheme. The experimental results indicated that, with the creation of spatial orientation trees using the LWT decomposed shape mask, uncoded coefficients can be efficiently predicted. This paves the way for advanced object-based coding schemes which may provide excellent very low bit-rate reconstructions by not having to completely code/decode the object shape mask, thus allocating more bandwidth to increase texture quality.

References

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

[2] A. Said and W. A. 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.

[3] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 725–743, Aug. 2000.

[4] O. Egger, P. Fleury, and T. Ebrahimi, "Shape-adaptive wavelet transform for zerotree coding," in *Proc. European Workshop Image Analysis and Coding*, vol. 1, 1996, pp. 201–208.

[5] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1063–1068, Sept. 2001.

[6] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds. Proc. SPIE 2569, 1995, pp. 68–79.