FULL PAPER
International Journal of Recent Trends in Engineering, Vol 2, No. 5, November 2009

Progressive Image Compression Using Contourlet Transform

A. Vasuki[1], P.T. Vanathi[2]
[1] Kumaraguru College of Technology / Department of ECE, Coimbatore, India
Email: avasuki@hotmail.com
[2] PSG College of Technology / Department of ECE, Coimbatore, India
Email: ptvani@yahoo.com

***Abstract*— This paper proposes a novel progressive image compression algorithm using the nonlinear contourlet transform. Contourlets are effective in capturing directional information in images using a flexible set of basis functions that are elongated and directional. Since the contourlet transform is redundant, a wavelet based contourlet transform has been applied on the image. The wavelet based contourlet transform is the wavelet transform followed by the contourlet transform on the subbands. The resulting set of coefficients is sparse and has been effectively coded using three different algorithms – Set Partitioning In Hierarchical Trees, Tag Setting In Hierarchical Trees and Bit Plane Coding. The results have been compared based on the peak signal-to-noise ratio for the *zoneplate* image and it is found to be superior to ordinary wavelet transform coding.**

***Index Terms* — Image Compression, Progressive, Contourlets, SPIHT, TSIHT, Bit Plane Coding**

I. INTRODUCTION

Efficient representation of visual information is the ability to capture significant information about an object of interest using a small description [1]. For image compression or content-based image retrieval, the use of an efficient representation implies the compactness of the compressed file or the index entry for each image in the database. The underlying basis of the reduction process is the removal of redundant information. Redundancy removal in images can be achieved by using a transform that operates on the image. The image is sparsely represented by the coefficients of the transformation, which leads to compression [2].

A computationally efficient image representation should be based on a local, directional, multiresolution expansion. The first three characteristics are successfully provided by separable wavelets, while the last two require new constructions. For one-dimensional piecewise smooth signals, wavelets have been established as the right tool, because they provide an optimal representation for these signals. Natural images are not simply stacks of 1D piecewise smooth scan-lines and edges are typically located along curves owing to smooth boundaries of physical objects [3][4]. Thus, natural images contain intrinsic geometrical structures that are key features in visual information. Image representation using separable orthonormal bases are not optimized since they do not account for geometric regularity of images. As a result of a separable extension from 1D bases, wavelets in 2D are good at isolating the discontinuities at edge points, but will not “see” the smoothness along the contours [5]. Therefore, more powerful representations are needed in higher dimensions.

Nonlinear wavelet transforms are more flexible in representing images. They change the filtering directions according to the image features, thus achieving more energy compaction for sharp features. The nonlinear wavelet transforms that are used in compressing images in this paper are the contourlets. The difference between the separable wavelet transform and nonlinear wavelet transform in capturing contours is illustrated in Fig.1 [6].

The linear wavelet transform has been combined with the contourlet transform to achieve compaction that is higher than either wavelet transform or contourlet transform. This new method is called Wavelet Based Contourlet Transform (WBCT). The coefficients of the transformation are encoded using three different techniques – Set Partitioning in HierarchicalTrees (SPIHT), Tag Setting in Hierarchical Trees (TSIHT) and Bit Plane Coding (BPC).

The performance of the algorithms has been compared based on certain performance measures. Rate is the average number of bits per pixel (bpp). The important parameter that indicates the quality of the reconstruction is the peak signal-to-noise ratio (PSNR). PSNR is defined as the ratio of square of the peak value of the signal to the mean squared error, expressed in decibels [7][8].

Section II deals with Wavelet Based Contourlet Transform and Section III discusses the proposed coding algorithm. Section IV gives the simulation results and Section V concludes the paper.

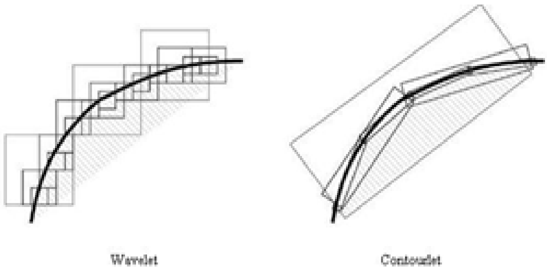

Figure 1. Wavelets vs. Contourlets

II. WAVELET BASED CONTOURLET TRANSFORM

The need for evolving a better coding technique by combining the simplicity of the wavelet transform and efficient coding of geometrical surfaces by the contourlets gave birth to the WBCT. In the contourlet transform, redundancy occurs at the Laplacian pyramid decomposition stage [9]. In order to take advantage of the directionality offered by contourlet transform and to avoid the redundancy, the Laplacian pyramid can be replaced with Mallat decomposition. This is the main idea behind a new transform called WBCT which is a non-redundant transform.

The WBCT is computed in two stages. The first stage provides subband decomposition, which is a wavelet transform. The second stage of the WBCT is a Directional Filter Bank (DFB) which provides angular decomposition [10]. The first stage is realized by separable filter banks, while the second stage is implemented using non-separable filter banks [11]. At each level in the wavelet transform, three high pass bands corresponding to the LH, HL and HH bands are obtained. The DFB with the same number of directions is applied in each band at a given level. Starting from the desired maximum number of directions $N = 2^i$ on the finest level J of the wavelet transform, the number of directions is decreased at every other dyadic scale (coarser levels, $i<J$). Hence, the anisotropy scaling law, i.e., *width* = *length*^{2} is achieved [12].

Fig. 2 illustrates the WBCT using three wavelet decomposition levels and three directional levels. Since there are mostly vertical directions in the HL image and horizontal directions in the LH image, partially decomposed DFBs with vertical and horizontal directions can be used on the HL and LH bands, respectively. However, since the wavelet filters are not perfect in splitting the frequency space to the lowpass and highpass components, fully decomposed DFBs are used on each band.

The process to compute the WBCT is as follows:

- Compute the DWT of an image, up to a maximum of log2(N) levels, where N x N is the size of the image.
- Design the directional filters.
- Perform the directional decomposition using the LH, HL and HH subbands.
- Repeat the above step with the next higher level of LH, HL and HH images.

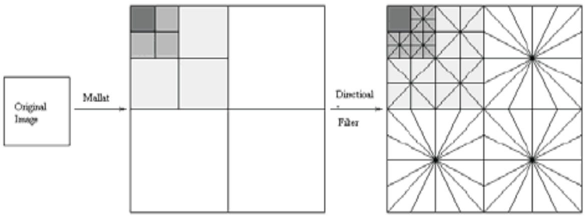

Figure 2. Computation of WBCT

III. PROPOSED ALGORITHM

The proposed algorithm applies WBCT on the input image and compresses the resulting coefficients using one of the three algorithms discussed below. The decomposition structure is repositioned to obtain a tree similar to the wavelet tree before applying SPIHT and TSIHT. The subbands are divided into codeblocks without repositioning for BPC. The SPIHT algorithm has been applied as it is, whereas the TSIHT algorithm has been modified slightly to get better performance. BPC has been applied after making a modification in the scalar quantization stage.

A. SPIHT

SPIHT is a wavelet-based progressive image compression technique [13] that reproduces high quality images at reduced bit-rates. To apply SPIHT algorithm, a tree structure called *spatial orientation tree* is defined in such a way that each node has either no offspring or four offsprings, which form a group of 2x2 adjacent pixels. The pixels in the highest level of the pyramid are tree roots, and their branching offsprings, except the leaves, also have 2x2 adjacent pixels grouped into blocks.

Let $c_{i,j}$ denote the wavelet transform coefficient at location (i,j) in the transformed image. The parent-offspring linkage, except at the highest and the lowest pyramid levels is,

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\} \qquad (1)$$

where $O(i,j)$ is the set of all immediate descendants of node (i,j). The set of *all* descendants of the node (i,j) is denoted by $D(i,j)$. The set of all descendants $L(i,j)$ excluding immediate descendants is $L(i,j) = D(i,j) - O(i,j)$.

While performing the SPIHT coding, three ordered lists are needed - List of Insignificant Sets (LIS), List of Insignificant Pixels (LIP), and List of Significant Pixels (LSP) to store the (in)significance information. In LIP and LSP, the entries represent individual pixels identified by coordinate (i,j). In LIS list, each entry represents a set either of type $D(i,j)$ or $L(i,j)$. The essence of SPIHT algorithm is to identify which coefficients are significant, sort selected coefficients in each *sorting pass*, and transmit the ordered refinement bits. Node tests and descendant tests are performed first with the maximum threshold 2^n where $n = \lfloor \log_2(\max\{|c_{i,j}|\}) \rfloor$. Then, it repeats with smaller thresholds, decrementing n by 1 iteratively, until the compressed bit rate reaches a predefined value.

B. TSIHT

TSIHT is a new algorithm [14] for progressive image coding, which is an improved version of the SPIHT. The advantages of TSIHT over SPIHT make it more favorable for hardware implementation and also in efficiently coding and compressing an image. The proposed TSIHT coding keeps low bit-rate quality as SPIHT and has three improved features:

Lesser memory requirement In SPIHT, large amount of memory may be occupied by the LSP, LIP and LIS lists. Instead, TSIHT uses three tag flags including TSP (Tag of Significant Pixels), TIP (Tag of Insignificant Pixels) and TST (Tag of Significant Trees) to distinct different entries in LSP, LIP and LIS respectively.

Improved refinement pass In TSIHT, the refinement pass is put before the sorting pass. Thus, TSIHT does not need to store last address or information of the refinement pass and is more efficient than SPIHT.

Efficient depth-first-search (DFS) In the sorting pass, SPIHT algorithm uses breadth-first-search (BFS) of the spatial orientation tree to traverse all the nodes. In TSIHT, DFS is used that searches the root node and each one of the branching nodes to the immediate descendants until it reaches the leaves. By using DFS, address generation of the ancestor-descendant coefficients is more efficient than SPIHT. Let TSP, TIP and TST be two-dimensional binary arrays, whose entries are either '0' or '1'. The overall TSIHT coding algorithm includes six steps as follows:

(i) Initialization: The threshold value is set according to,

$$n = \text{floor}(\log_2(\max(\text{coefficient\_value})))$$

and all the values of TSP, TIP and TST arrays are set to zero.

(ii) Refinement Output: The algorithm checks the TSP for any significant bit at that particular threshold *n*. If TSP=1, then the n^{th} most significant bit of the coefficient is output; otherwise, there is no output.

(iii) TIP Testing: TIP is used to store the list of insignificant pixels which have already been scanned for a particular threshold. When the threshold is changed, it is scanned again and the significance information and sign (if significant) are transmitted. TIP and TSP are updated.

(iv) TST Update: In this step, each pixel is checked for its significance at a particular threshold. The different cases are as follows:

- For the root, TST is always zero.
- From the next pixel onwards, if TST =0 and if the value of the pixel is greater than the threshold, TST of that pixel is set to '1'. If it is not significant, then the descendants of that pixel are checked for significance until a significant pixel is found and that particular pixel's TST is marked as '1'. The remaining pixels are not considered.
- After all traversal along a particular parent-child relation is done, the first pixel which was scanned but was insignificant is added to the TIP list.

(v) Spatial Orientation Tree Encoding: The Spatial Orientation Tree is encoded according to the following steps:

- If TSP = 0 and TIP = 0, then check for significance of the pixel.
- If significant, then set TSP = 1 and add it to the significant pixels list. Output '1', sign of the coefficient and value of the TST.
- If insignificant, then set TIP =1 and add it to the insignificant pixels list. Output '0' and value of the TST. Check the TST value of the pixel,
    - ✓ If TST=1, check the significance of the descendants. If there are any significant children, they are also added to the TSP list.
    - ✓ If TST = 0, there are no significant children, and the process is stopped.

(vi) Quantization Step Update: In this step, the threshold value is decreased for the next pass of the algorithm.

TSIHT coding algorithm is the same as SPIHT but uses different data structures and has lower bit rate than SPIHT; hence it is more efficient.

C. Bit Plane Coding

The JPEG2000 standard [15] has incorporated embedded block coding using fractional bit planes as the compression algorithm. The standard applies wavelet transform on the input image, and the resulting coefficients are scalar quantized. The quantization step size is dependent on the subband in which the coefficient appears, as proposed in the standard. The step size is progressively increased for lower levels (higher frequencies). The modification proposed here is in the scalar quantization stage where the step size is applied as per the standard for the highest three levels of decomposition and after three levels the same step size is maintained instead of increasing. This is done because here the contourlet transform is applied on the last two level subbands and a larger step size causes loss of high frequency information.

The principle of bit-plane coding is to first encode the most significant bits for all samples in the code-block, then the next most significant bits, and so forth until all bit-planes have been encoded. The algorithm uses context adaptive coding to efficiently represent the quantized transform coefficients. Context coding refers to bit plane coding taking into account the surrounding bit patterns and significant bit values in the neighborhood of the value under consideration. Bit plane coding primitives refer to four coding operations which form the foundation of the block coding algorithm. The primitives are used to code new information for a single sample in some bit-plane. For each bit-plane, the coding proceeds in a number of distinct passes, which are identified as *fractional bit-planes*. The different passes of the algorithm are:

Cleanup pass In this pass, the MSB of every sample is coded. The sign bit is also coded if it is found to be significant.

Significance Pass In this pass, the sub-block samples are visited in scan-line order, skipping over all samples which are either insignificant or do not have one of the eight neighbors to be significant (preferred neighbourhood). The samples, whose previous bit-plane value is 0 and has a preferred neighborhood gets coded in addition to those samples that are significant in this pass. The sign bit of the significant samples is also coded.

Magnitude Refinement Pass During this pass, the samples which were found to be insignificant in the previous two passes i.e. for which no information has been coded in the previous two passes are scanned and processed.

Normalization Pass Here, the LSB of all samples not considered in the previous three coding passes is scanned and coded. If a sample is found to be significant in this process, its sign is coded.

Each bit-plane within a code-block is scanned during the context coding process in a specific order. A key advantage of scalable compression is that the target bit-rate or reconstruction resolution need not be known at the time of compression. Since the algorithm deals with independent code blocks, the blocks can be processed simultaneously thereby reducing the implementation time. Block processing enables usage of smaller memory as opposed to large amount of memory required by other algorithms which process the image as a whole. It has a modest complexity and it is resilient to errors.

IV. SIMULATION RESULTS

The results of encoding the test image *zoneplate* with each of the three algorithms discussed above are presented in Table I and Table II. The DWT or WBCT is applied on the input image and the coefficients are compressed using SPIHT, TSIHT and Bit Plane Coding. The wavelet used in this paper is *bior4.4*, since it is found to produce superior results than the other wavelet families for images with contours. The *bior4.4* wavelet is nearly symmetric with linear phase [16].

The original image *zoneplate* is shown in Fig. 3. The reconstructed *zoneplate* images for SPIHT, TSIHT and Bit Plane Coding for DWT and WBCT are shown in Fig. 4 and Fig. 5 respectively. The reconstructed images are superior in visual quality and have higher PSNR for all the three algorithms, especially at low bit rates.

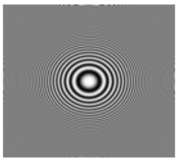

Figure 3. Original Image, *zoneplate*

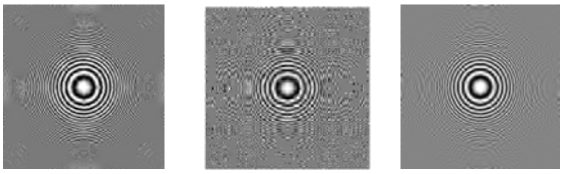

Figure 4. Reconstructed images, DWT
(a) SPIHT, PSNR = 11.51 dB, (b) TSIHT, PSNR = 11.85 dB,
(c) BPC, PSNR = 13.98 dB

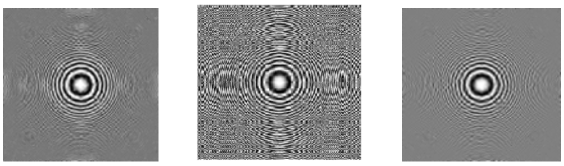

Figure 5. Reconstructed images, WBCT
(a) SPIHT, PSNR = 14.30 dB, (b) TSIHT, PSNR = 14.05 dB,
(c) BPC, PSNR = 14.27 dB

The SPIHT algorithm with WBCT surpasses DWT by up to 8dB for the *zoneplate* image since it has more contours. The TSIHT algorithm with WBCT improves the PSNR up to 2.5dB over the DWT. The bit plane coding algorithm also improves the reconstructed image quality with WBCT. The PSNR for bit plane coding is higher for WBCT by around 1 dB. The improvement in PSNR for all the three algorithms is variable, depending on the wavelet, bit rate, directional filter and the image.

The results obtained for the linear wavelet transform are compared with those obtained for the nonlinear wavelet based contourlet transform and it is found that the WBCT outperforms DWT for images with contours. The *bior4.4* wavelet is used in the pyramidal decomposition stage of WBCT and the directional filter used is ‘pkva’. The ‘pkva’ filter is found to give better result than ‘9-7’ or ‘5-3’.

TABLE I
RESULTS OF ENCODING, DWT

bpp	*zoneplate*, **PSNR dB**		
	SPIHT	**TSIHT**	**BPC**
0.1	9.08	9.32	9.76
0.2	9.81	9.87	10.27
0.3	10.41	10.37	11.52
0.4	11.06	10.84	12.73
0.5	11.51	11.46	13.98
0.6	12.74	12.25	14.49
0.7	14.15	13.47	14.97
0.8	16.43	14.66	15.57
0.9	16.82	15.22	16.23
1.0	17.71	15.81	16.85

TABLE II
RESULTS OF ENCODING, WBCT, DFILT = ‘PKVA’

bpp	*zoneplate*, **PSNR dB**		
	SPIHT	**TSIHT**	**BPC**
0.1	8.74	9.70	10.24
0.2	10.69	10.80	11.66
0.3	12.64	11.69	12.50
0.4	13.23	13.34	13.28
0.5	14.30	13.87	14.27
0.6	15.36	14.42	15.11
0.7	16.46	15.57	15.92
0.8	17.43	16.94	16.49
0.9	18.20	17.55	17.17
1.0	19.12	17.91	17.97

V. CONCLUSIONS

A new image compression model combining WBCT with SPIHT, TSIHT or Bit Plane Coding has been proposed which is found to be an improvement over the existing coding schemes based on DWT. With this compression scheme, significant improvements in PSNR values have been gained. Experiments also indicate an *enhancement in the visual results* when compared with the DWT based coders *in preserving details and textures in the coded images* [17]. Although SPIHT has been proved to be a more efficient algorithm than EZW, the problem of large memory requirements may restrict its applications. In this proposed work, WBCT/ TSIHT coding using tag flags can effectively *reduce* memory usage. For a typical 256×256 gray-scale image, TSIHT only needs 18 *K* bytes, while SPIHT needs 250 *K* bytes memory. In the given example, SPIHT requires almost *13.9 times* the memory of TSIHT coding, which makes TSIHT more convenient to be integrated into other image compression systems. Therefore, TSIHT is more suitable than SPIHT for hardware implementations.

The bit plane coding algorithm with WBCT gives higher PSNR values and also produces improvement in the quality of reconstructed images over DWT. The enhancement in visual quality is more important for images with textures and directional information and it is provided by contourlets. The possibility of coding images with preservation of directional details offers a wide range of applications such as medicine, mobile devices and face recognition. There is scope for further enhancements to be done to the compression model, in terms of applying the lifting scheme [18] and also in further optimizing the entire code to better improve the efficiency of the whole image compression scheme. Applying further compression to the output of the coder can reduce the bit rate considerably at the cost of increased complexity.

REFERENCES

[1] "Wavelets: Seeing the Forest and the Trees", *Beyond Discovery:The Path from Research to Human Benefit*, December2001. www.nationalacademies.org/publications/

[2] Subhasis Saha, "Image Compression – from DCT to Wavelets: A Review", *ACM Crossroads Student Magazine*, 2000. www.acm.org/crossroads/xrds6-/sahaimgcoding.html

[3] Minh N. Do, Martin Vetterli, "The Contourlet Transform: An Efficient Directional Multiresolution Image Representation", *IEEE Transactions on Image Processing*, Vol. 14, No. 12, pp. 2091-2106, Dec. 2005.

[4] Ramin Eslami, Hayder Radha, "Translation-Invariant Contourlet Transform and Its Application to Image Denoising", *IEEE Transactions on Image Processing*, Vol. 15, No. 11, pp. 3362 – 3374, Nov. 2006.

[5] Minh N. Do, Martin Vetterli, "Contourlets: A Directional Multiresolution Image Representation", *IEEE International Conference on Image Processing*, (ICIP2002), pp.I-357–I-360.

[6] Duncan D.-Y. Po, Minh N. Do, "Directional Multiscale Modeling of Images Using the Contourlet Transform", *IEEE Transactions on Image Processing*, Vol. 15, No.6, pp. 1610 – 1620, June 2006.

[7] R.C.Gonzalez, R.E.Woods, *Digital Image Processing*, New Delhi, Pearson Education Pvt. Ltd., 2nd Edition.

[8] Khalid Sayood, *Introduction to Data Compression*, New Delhi, Harcourt India Private Limited, 2nd edition, 2000.

[9] Vivien Chappelier, Christine de Beaulieu, "Image Coding with Iterated Contourlet and Wavelet Transforms", *IEEE International Conference on Image Processing* (ICIP 2004), Vol. 5, pp. 3157-3160, Oct. 2004.

[10] Ramin Eslami, Hayder Radha, "Wavelet-Based Contourlet Transform and its Application to Image Coding", *IEEE International Conference on Image Processing* (ICIP 2004), Vol. 5, pp. 3189-3192, Oct. 2004.

[11] Arthur L.da Cunha, Jianping Zhou, Minh N. Do, "The Nonsubsampled Contourlet Transform: Theory, Design and Applications", *IEEE Transactions on Image Processing*, Vol. 15, No.10, pp. 3089 – 3101, Oct. 2006.

[12] Ramin Eslami, Hayder Radha, "Wavelet-based Contourlet Coding Using an SPIHT-like Algorithm", *Conference on Information Sciences and Systems* (CISS), March 2004.

[13] A. Said, W.A. Pearlman, "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp. 243 – 250, June 1996.

[14] Tsung-Hsi Chiang, Lan-Rong Dung, "A VLSI Progressive Coding for Wavelet-based Image Compression", *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 2, pp. 569 – 577, May 2007.

[15] ISO Standard, *JPEG 2000 Image Coding System*, Final Committee Draft, ISO / IEC FCD 15444 – 1 :2000, V.1 March 2000.

[16] M.Antonini, M.Barlaud, P.Mathieu, I.Daubechies, "Image Coding using Wavelet Transform", *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 205-220, Feb. 1992.

[17] Ramin Eslami, Hayder Radha, "On Low Bit-Rate Coding Using the Contourlet Transform", *Thirty-seventh Asilomar Conference on Signals, Systems & Computers*, Vol. 2, pp. 1524-1528, Nov. 2003.

[18] Roger Claypoole, Goeffrey Davis, Wim Sweldens, Richard Baraniuk, "Nonlinear Wavelet Transforms for Image Coding", *Thirty-first Asilomar Conference on Signals, Systems & Computers*, Vol. 1, pp. 662-667, Nov. 1997.