

A New Transform For Irregular Roi Coding Of Two Dimensional Images

Anaswara V. Nath¹, Archana V. R.², Sreedhu T. Sasi³, Sreekumar K.⁴

^{1,2,3}(M.Tech students, Electronics and communication, College Of Engineering, Cherthala, Alappuzha, kerala)

⁴(Asst .prof, Electronics and communication, College Of Engineering, Cherthala, Alappuzha, kerala, India)

Abstract: Image compression is the application of data compression on digital images. Image compression can be lossy compression or lossless compression. Raw data from an image sensor obviously contains information about a scene, but it is not intrinsically recognizable to the human eye. In signal processing, a lapped transform is a type of linear discrete block transformation where the basis functions of the transformation overlap the block boundaries. We propose a new shape-adaptive (SA) transform ,shape-adaptive reversible integer lapped transform (SA-RLT) method. Region-of interest (ROI) coding algorithms have also been proposed, since observers are always more concerned about some special areas, i.e., ROI than the background (BG) area. Based on SA-RLT and object-based set partitioned embedded block coder (OBSPECK), a new ROI compression scheme can be designed for 2D images.

Keywords: Blocking Artifacts, Multilifting scheme, Region of interest (ROI), RTDLT, SA- DCT, SA- DWT, SA-RLT.

I. INTRODUCTION

With the emergence of multimedia applications, functions such as access, searching, indexing and manipulation of visual information at the semantically meaningful object level, are becoming very important in research and standardization efforts. There have been continuous efforts in developing coding techniques for arbitrarily shaped objects.

Compression techniques have been widely researched since image will take up a great deal of storage space and bandwidth particularly with the increase of spatial resolution. Among various compression techniques, lossy-to-lossless compression technique exhibits great flexibility, because it is able to compress images at a high compression ratio at the cost of losing some minor information and it is also able to realize lossless compression without any information distortion [1]. Region-of interest (ROI) coding algorithms have also been proposed, since observers are always more concerned about some special areas (i.e., ROI) than the background (BG) area[2].

In the JPEG2000-ROI coding scheme, ROI and BG areas are transformed together, and then, ROI coefficients are scaled up to improve its quality. This will lead to blurring artifacts along the boundary of ROI and BG areas, resulting in degradation of visual effect. Blurring artifacts have been avoided, since just a same number of coefficients are needed to encode for special regions.

SA transform, as one of the key technologies in region based coding scheme, has obtained attentions since many years ago for its good performance in object/region-based image coding. Different from conventional transform methods such as discrete cosine transform (DCT) and discrete wavelet transform (DWT) for rectangular areas [3], SA transforms can be adaptive to arbitrarily shaped regions. SA-DCT proves to be efficient in decorrelation and with low complexity in implementation. It is a good choice for object-based image coding except that, in some cases, it is criticized for blocking artifacts inherent in block transform. SA-DWT has also been researched since DWT possesses special features such as multiresolution to locate high-frequency details and global-transform character which will avoid blocking artifacts. In SA-DWT, an arbitrary-length wavelet filter is performed on contiguous segments after scanning each column in the vertical direction, and then, the process is repeated along the horizontal direction.

One of the disadvantages of the global transform is that the computational complexity and memory requirement increases linearly with size of the image [4]. This problem seems to be more obvious nowadays, particularly when a high resolution image is required. It will be an excellent scheme if this scheme keeps low memory requirement property of the block transform while its performance is comparable with the global transform. Based on such considerations we explore another approach for SA transform, i.e. SA-RLT.

Section 2 deals with traditional lapped transforms and various shape adaptive transforms such as SA-DCT and SA-DWT. In Section 3 we discussed about the proposed methodology. Section 4 presents the results obtained. Conclusion and its future extensions are presented in section 5.

II. PRIOR ART

Transform coding, one of the most popular image compression techniques, generally divides the original image into subimages called blocks. Each block is transformed and the selected large coefficients are quantized and then transmitted/stored. One of these techniques' drawbacks is that the discontinuities at the block boundaries are visible. Such a discontinuity implies the appearance of high frequency components which can be eliminated by using low pass filters. The advantage of using a low pass filtering technique is that no additional information is needed, so the bit rate is not increased. However, it results in blurred images.

2.1 BLOCK TRANSFORM

In traditional block-transform processing, such as in signal, image and audio coding the signal is divided into blocks of M samples, and each block is processed independently[5]. Let the samples in the m-th block be denoted as:

$$x_m = [x_1, x_2, \dots, x_M]$$

For x_m and let the corresponding transform vector be:

$$y_m = [y_1, y_2, \dots, y_M]$$

For a real unitary transform A, $A^T = A^{-1}$. The forward and inverse transforms for the m-th block are $y_m = Ax_m$ and $x_m = A^T y_m$. The rows of A, denoted as a_n^T , and are called the basis vectors because they form an orthogonal basis for the M-tuples over the real field. The transform vector coefficients y_n represent the corresponding weights of vector x_m with respect to this basis.

2.2 LAPPED TRANSFORM

The idea of a lapped transform (LT) maintaining orthogonality and nonexpansion of the samples was developed in the early 80s at MIT by a group of researchers unhappy with the blocking artifacts so common in traditional block transform coding of images. The new idea was to extend the basis function beyond the block boundaries. Hence creating an overlap, in order to eliminate the blocking effect. This idea was to create overlapping blocks, so that it would be the same as if there was no overlap, and that the transform would maintain orthogonality. Cassereau introduced the Lapped Orthogonal Transform (LOT), and Malvar gave the LOT its design strategy and a fast algorithm. It was later pointed by Malvar the equivalence between an LOT and a multirate filter bank. Based on cosine modulated filter banks, modulated lapped transforms were designed. Recently a new class of LTs with symmetric bases were developed yielding the class of generalized LOTs. We, however, refer to LTs for paraunitary uniform FIR filter banks with fast implementation algorithms based on special factorization on the basis functions[6],[7].

2.2.1 ORTHOGONAL LAPPED TRANSFORM

There are M basis functions for the DCT or the LT although Fig.1 shows just one of them.

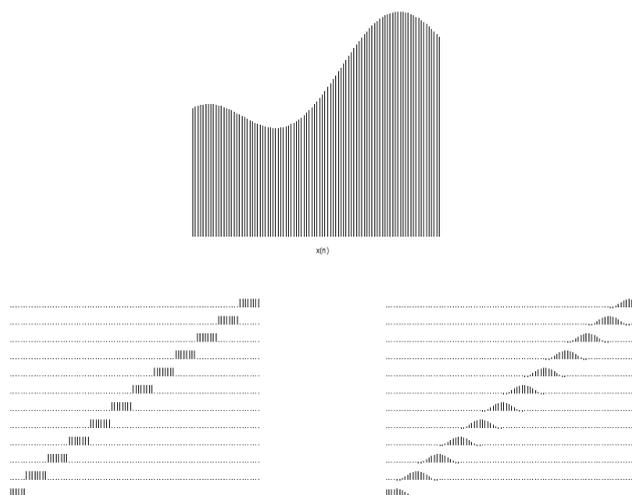


Fig.1: The example discrete signal $x(n)$ is to be projected onto a number of basis. Left: Spatially displaced versions of the first DCT basis. Right: Spatially displaced versions of the first basis of a typical short LT

An example of the bases for $M=8$ is shown in Fig.2. where we plot the bases for the DCT and for the LOT. We can see that not only are the LOT bases longer but they are also smoother than the DCT counterpart.

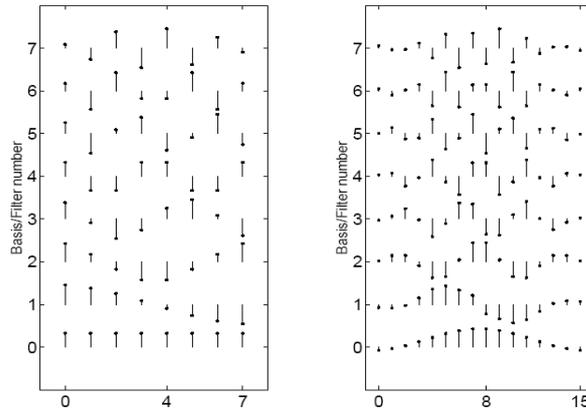


Fig.2: Bases for the 8-point DCT ($M=8$) (left) and for the LOT (right) with $M=8$

For lapped transforms the basis vectors have length L , such that $L > M$, extending across block boundaries. Thus, the transform matrix is no longer square and most of the equations valid for block transforms do not apply to LT. We concentrate on orthogonal LTs and consider $L=NM$, where N is the overlap factor. N , M and hence L are integers. As in the block transforms, we define transform matrix with the orthonormal basis vectors as its rows. In lapped transform, transformation matrix P of dimensions $M \times L$ can be divided into square $M \times M$ submatrices $P = [P_1 \dots P_N]$ as $P = [P_1 \dots P_N]$. The orthogonality property does not hold because P is no longer square matrix.

We divide the signal into blocks, each of size M , we would have the vectors x_m and y_m as eqns. (1) and (2). These blocks are not used by the LTs in a straightforward manner. The actual vector which is transformed by the matrix P has to have L samples. At block number m , it is composed of the samples of x_m plus $L - M$ samples. These samples are chosen by picking $(L - M)/2$ samples at each side of the block x_m , as in Fig.3 for $N=2$.

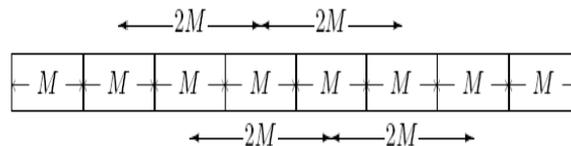


Fig.3: The signal samples are divided into blocks of M samples

The lapped transform uses neighboring block samples, as in this example for $N=2$, i.e. $L=2M$, yielding an overlap of $(L - M)/2 = M$ samples on the either side of the block. However the number of transform coefficients at each block is M , and in this respect there is no change in the way we represent the transform-domain blocks y_m . The input vector of length L is denoted as v_m , which is centered around the block x_m and is denoted as:

$$v_m = [x_{m-1} \dots x_m \dots x_{m+1}]^T$$

Then, we have $y_m = P v_m$. The inverse transform is not direct as in the case of block transform. i.e., with the knowledge of y_m we do not know the samples in the support region of v_m , and neither in the support region of x_m . We can reconstruct a vector $\hat{v}_m = P^T y_m$ where $\hat{v}_m \neq v_m$. To reconstructs the original sequence, accumulate the results of the vectors \hat{v}_m , in a sense that a particular sample $x(n)$ was included in the region of support of the corresponding v_m .

2.3 SHAPE ADAPTIVE DCT

The shape-adaptive DCT algorithm proposed is based on predefined orthogonal sets of DCT basis functions [8]. In contrast this algorithm is fast to compute and does not require more computations than a normal DCT [9].

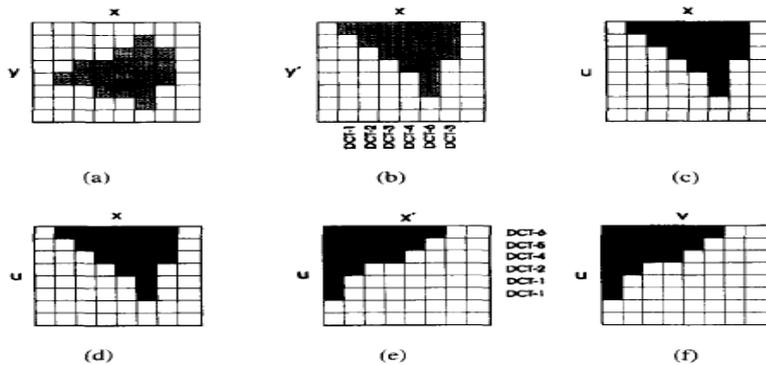


Fig.4. Successive steps involved to perform a SA-DCT forward transformation on an arbitrarily shaped image foreground segment.

The basic concept of the proposed method is outlined in Fig.4 for coding an arbitrarily shaped image foreground segment contained within an 8 x 8 reference block. Fig.4(a) shows an example of an image block segmented into two regions, foreground (dark) and background (light). To perform vertical SA-DCT transformation of the foreground, the length (vector size N , $0 < N < 9$) of each column j ($0 < j < 9$) of the foreground segment is calculated, and the columns are shifted and aligned to the upper border of the 8 x 8 reference block Fig.4(b). Dependent on the vector size S of each particular column of the segment, a DCT transform matrix DCT- N :

$$C_{k,p} = \begin{cases} \frac{1}{\sqrt{2}} & p = 0 \\ C_0 = 1 & \text{otherwise} \end{cases}$$

where $k,p=0,\dots,N-1$ containing a set of N basis vectors is selected. Here $c_0 = \sqrt{\frac{1}{2}}$ if $p = 0$, and $C_0 = 1$ otherwise; p denotes the p th DCT basis vector. The N vertical DCT-coefficients c_j for each segment column data x_j are calculated according to the formula:

$$c_j = \sum_{k=0}^{N-1} x_j C_{k,p}$$

After SA-DCT in vertical direction, the lowest DCT coefficients for the segment columns are found along the upper border of the 8 x 8 reference block Fig.4(c). To perform the horizontal DCT transformation Fig.4(e), the length of each row is calculated, and then, the rows are shifted to the left border of the 8 x 8 reference block. A horizontal DCT is adapted to the size of each row is then calculated using (4) and (5). Horizontal SA-DCT transformation is performed along vertical SADCT coefficients with the same index. Fig. 4(f) shows the final location of the resulting DCT coefficients within the 8 x 8 image block.

The final number of DCT coefficients is identical to the number of pels contained in the image segment. Also, the coefficients are located in comparable positions as in standard 8 x 8 block. The DC coefficient is located in the upper left border of the reference block, and, dependent on the actual shape of the segment, the remaining coefficients are concentrated around the DC coefficient. Since the contour of the segment is transmitted to the receiver, the decoder can perform the shape-adapted inverse DCT as the reverse operation.

2.4 SHAPE ADAPTIVE DWT

Shape Adaptive Discrete Wavelet Transform (SA-DWT) [11] decomposes the arbitrarily shaped objects in the image domain to a hierarchical structure with a set of subbands of varying resolutions. Each subband has a corresponding shape mask associated with it to specify the locations of the valid coefficients in that subband.

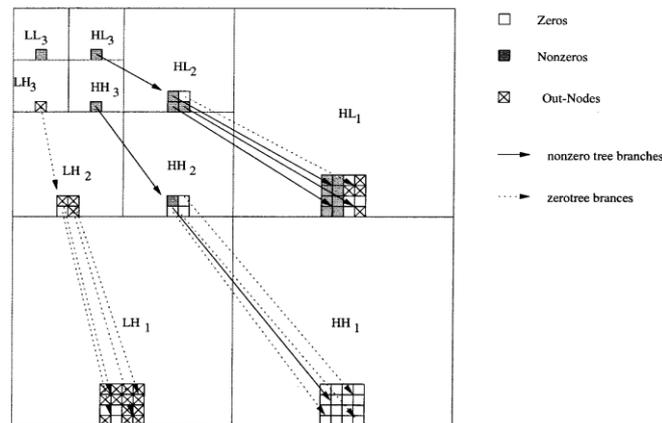


Fig 5: Parent child relation of wavelet trees in a SA DWT bands

As shown in the figure, there are three types of nodes in a tree: zeros, non-zeros, and out-nodes (with don't care values). The major task is to extend the conventional ZTC (EZW or ZTE) methods to the case with out-nodes. A simple way is to set those don't care values to zeros and then apply the conventional EZW coding method. However this requires bits to code the out-nodes, such as a don't care tree (parent and all of its children have don't care values). This is a waste of bits because out-nodes do not need to be coded as the shape mask already indicates their status. Therefore we should treat out nodes differently from zeros [12]. Although we do not want to use bits to code an out-node, we have to decide what to do with its children nodes.

One way is not to code any information about the status of the children nodes of the don't care node. In this way, we always assume that it has four children to be examined further. When the decoder scans to this node, it will be informed by the shape information that this node is a don't care node and it will continue to scan its four children nodes. By doing so, all the don't care nodes in a tree structure need not be coded. This approach performs well when there are only sparse valid nodes in a tree structure. One disadvantage of this approach is that, even if a don't care node has four zerotree root children, it still needs to code four zerotree root symbols instead of one.

Another way is to selectively treat an out-node as a zero. But this is equivalent to creating another symbol for coding some don't care values. We decide to use the method of not coding out-nodes. The extensions of ZTC coding to handle the SA-DWT coefficients are then given as follows. Encoding and decoding SA-DWT coefficients is the same as encoding and decoding regular wavelet coefficients, except with keeping track of the locations of the wavelet coefficients according to the shape information. As with encoding and decoding regular wavelet coefficients, a zerotree symbol is used to determine whether encoding and decoding are needed for its children nodes. The difference is that some zerotree nodes correspond to the pixel locations outside the shape boundary and no bits should be used for these out-nodes. The following description applies to both EZW and ZTE[13] coding schemes for each coding pass.

At the root layer of the wavelet tree, the shape information is examined to determine whether a node is out-node. 1) If it is out-node, no bits are used for that node and the four children of this node are marked to-be-encoded (TBE) in encoding or to-be-decoded (TBD) in decoding. 2) Otherwise, a symbol is encoded/decoded for this node using an adaptive arithmetic encoder/decoder. a) If the symbol is either isolated-zero (IZ) or value (VAL), the four children of this node are marked TBE/TBD; otherwise, the symbol is either zerotree-root (ZTR) or valued-zerotree-root (VZTR) and the four children of this node are marked no-code (NC). b) If the symbol is VAL or VZTR, a nonzero wavelet coefficient is encoded/decoded for this node; otherwise, the symbol is either IZ or ZTR and the wavelet coefficient is set to zero for this node.

At any layer between the root layer and the leaf layer, the shape information is examined to determine whether a node is an out-node. 1) If it is an out-node, no bits are used for this node and the four children of this node are marked as either TBE/TBD or NC depending on whether this node itself is marked TBE/TBD or NC, respectively. 2) Otherwise: a) if it is marked NC, no bits are used for this node and the wavelet coefficient is zero for this node and the four children nodes are marked NC; b) otherwise, a symbol is encoded/ decoded for this node using an adaptive arithmetic encoder/decoder. i) If the symbol is either IZ or VAL, the four children of this node are marked TBE/TBD; otherwise, the symbol is either ZTR or VZTR and the four children of this node are marked NC. ii) If the symbol is VAL or VZTR, a nonzero wavelet coefficient is encoded /decoded for this node using an arithmetic encoder/decoder; otherwise, the symbol is either IZ or ZTR and the wavelet coefficient is set to zero for this node.

At the leaf layer, the shape information is examined again to determine whether a node is an out-node.1) If it is an out-node, no bits are used for this node.2) Otherwise: a) if it is marked NC, no bits are used for this node and the wavelet coefficient is set to zero for this node; b) or, adaptive arithmetic encoder/decoder.

III. Proposed Method

An alternative for the present SA transform methods is introduced. Wang et al. have proposed reversible integer time domain lapped transform (RTDLT)[15],[16]. Here we can extend RTDLT to SA RTDLT, which can be applied in progressive lossy-to-lossless compression for arbitrarily shaped image areas. We refer to SA RTDLT as SA reversible integer lapped transform (SA-RLT) for convenience. SA-RLT is block based, and its structure can be easily designed. The number of transform coefficients after SA-RLT is equal to that of pixels in the ROI area. Based on SA-RLT, we designed a new ROI coding scheme for 2-D RSIs. ROI and BG areas are transformed and coded separately, while their visual effects can be flexibly controlled by a quality factor. Moreover, we can realize lossless compression of the entire image or just the ROI area when needed.

3.1 REVERSIBLE TIME DOMAIN LAPPED TRANSFORM

Reversible Time Domain Lapped Transform (RTDLT) [17] belongs to block transform ;thus, the source image will be segmented into adjacent nonoverlapping blocks before transforming. All lapped transforms can be viewed as post- and pre-processing of the DCT coefficients with the quantizer in between. LOT has been shown to comprise of either i) cross – boundary post-processing of a certain block transforms output or ii) cross-boundary pre-processing of a block transforms input.

Through a series of elementary matrix manipulations, the type-II fast LOT can be viewed as a combination of the common block-based DCT/IDCT framework with simple time-domain pre- and post-filtering. Fast lapped transforms can be constructed in polyphase form from components with well-known fast-computable algorithms such as the DCT and the DST. One of the most elegant solution is the type-II fast LOT whose the analysis polyphase matrix can be rewritten as:

$$R_{NR}^{-1} \begin{pmatrix} I & 0 & 0 \\ 0 & D_M & 0 \\ 0 & 0 & I \end{pmatrix} C_M^{-1}$$

where,

$$P = \begin{pmatrix} I & J & 0 & I & J \\ 0 & J & 0 & I & J \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

$$V_{NR} = \begin{pmatrix} C_M^T & 0 \\ 0 & C_M^T \end{pmatrix}$$

Since the diagonal matrix D_M only inverts the polarity of the transform coefficient, it can be ignored. Finally,

if we define $\hat{P} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & D_M & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$ as the permuted advance chain, then the modified

LOT polyphase matrix becomes $R_{NR}^{-1} \hat{P} C_M^{-1}$, where $\hat{P}^{-1}(Z)P$ can be interpreted as time-domain pre-processing across block boundaries. The new LT structures is illustrated in figure , where the

resulting basic functions are, discounting $\frac{M}{2}$ sign changes from D_M exactly the type-II fast LOTs. The synthesis

polyphase matrix is simply the inverse of analysis polyphase matrix. $C_M^{II^T}$ is the M-point type-II IDCT matrix. $R_{NR} = \begin{pmatrix} I & 0 & 0 \\ 0 & D_M & 0 \\ 0 & 0 & I \end{pmatrix}$

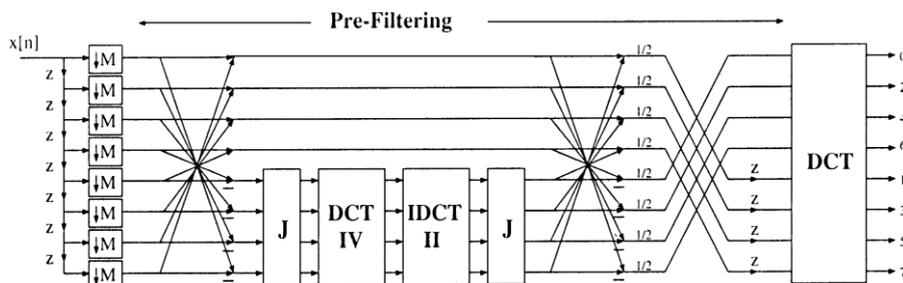


Fig.6: LOT via time domain pre- and post-processing

In the decomposition stage, P act as the pre-filter working across the block boundaries, taking away interblock correlation: the pre-processed time samples are then fed to the DCT to be encoded as usual. In the reconstruction stage, P^{-1} serves as the post-filter, reconstructing the data in an overlapping manner, hence alleviating blocking artifacts [11]. The symmetry of the basis functions is guaranteed by the specific structure of P as in (7), regardless of the choice of the free-parameter matrix V . The advance chain $\lambda^{-1}(Z)$ extends the processing across the block boundary [17].

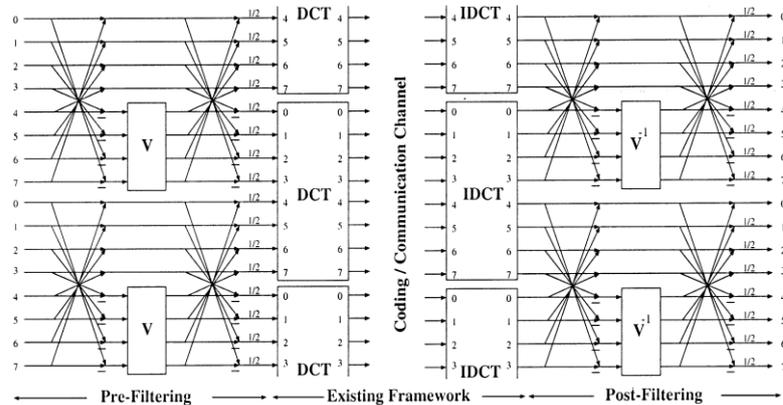


Fig.7: Global viewpoint of LT as pre- and post-filtering at DCT/IDCT block boundaries.

Pre and post-filtering are operating in the time domain, completely outside of the existing block-based architecture. Because of this characteristic, we label this LT family in the time domain lapped transform.

3.2 SHAPE ADAPTIVE-RLT

Based on RTDLT, we propose SA-RLT with the same concept as used in SA-DCT and SA-DWT. However SA-RLT has its spectral advantages [17]. The first one is that SA-RLT belongs to block transform; as a result, its computational memory requirement is lower than that of global transforms, and its hardware implementations can be parallel processed. The second one is that SA-RLT exploits the correlation between the neighboring blocks; therefore, it can improve the efficiency of the conventional block based transforms. Moreover, SA-RLT can realize reversible integer to integer transform. How to realize SA-RLT is discussed as follows.

We first compute pre-filter and DCT matrices of different sizes using (6)-(8). Segmenting image into 8×8 blocks has been proved to be the best tradeoff between transform performance and computational complexity in conventional DCT based compression schemes such as JPEG. Therefore, in our scheme, basic transform matrices with a size of $2 \times 2 - 8 \times 8$ are computed first.

Second, we design a multilifting scheme [16] for prefilter and DCT matrices obtained from the first step. To do this, we use the matrix factorization method to factorize basic matrices into TERMS. Suppose that matrix L is a lower TERM factorized from basic transform matrix with a size of 3×3 .

In this way, reversible integer prefilter and DCT for 1-D arbitrary-length vector have been obtained. A 2-D transform can be obtained by cascading 1-D transform along the vertical and horizontal directions, respectively. It should be noted that the lengths of vertical and horizontal vectors may be different, and each length should be figured out by scanning the vector before transform. Up to now, we have achieved reversible integer prefilter and DCT for arbitrarily shaped image area.

In one arbitrarily shaped image area, there exist two types of blocks. The first type of block is incomplete, which is along the area boundary, the second type of block is full, which is inside the image area. We employ different strategies on these two types of blocks. On one hand, we use SA-RDCT for incomplete blocks on the other hand; we use RTDLT for full blocks. Thus, strictly speaking, SA-RLT means the integration of SA-RDLT and RTDLT. In fact, we have taken experiments to perform SA-R-filters on incomplete blocks along object edges, but the performance improvement is very limited. Therefore, to reduce complexity, we perform SA-RLT employing the aforementioned strategies. There exists no truncation error in the transformation process, since SA-RLT can realize completely reversible integer to integer transform in both incomplete and full blocks in arbitrarily shaped image area. Therefore, SA-RLT can be applied in both lossy and lossless compression.

3.3 SA-RLT BASED ROI CODING FOR RSI

In this section, how to design ROI coding scheme based on SA-RLT and how to control ROI quality is discussed. One advantage of the proposed scheme over JPEG2000-ROI coding is that it codes ROI and BG separately, therefore, bit rate can be controlled more flexibly, and ROI can be coded losslessly without BG. First, we apply SA-RLT to ROI and BG, respectively. Coefficients will be distributed in a blocky structure after transforming. Most energy will be concentrated in the left-upper corner of each block. In order to code the transform coefficients of SA-RLT using wavelet-based codec, we reorganize the blocky coefficients into subband structure like that in wavelet transform. Second, we code these coefficients using object-based set partitioned embedded block(OB-SPECK). OB-SPECK is an efficient algorithm for coding arbitrarily shaped objects. It can be proved that OB-SPECK performs better than the state of the art image compression techniques such as traditional SPIHT, SPECK and JPEG2000 for region-based digital mammography. Moreover, it possesses characteristics of low complexity and embedded/progressive bit streams. A binary shape mask, which corresponds to the ROI area, is coded by OB-SPECK independently and transmitted with the whole bit stream.

3.3.1 OBJECT-BASED SPECK ALGORITHM

A straight forward extension of the SPECK algorithm for coding of ROI of arbitrary shape is that we set all the coefficients outside the objects in each subband to zero. Then the original SPECK algorithm can be applied as if the support of the object were rectangular. No modification of the algorithm would be required. This method is an efficient, since one bit must be transmitted to tell the decoder that the node or branch outside is insignificant under each threshold.

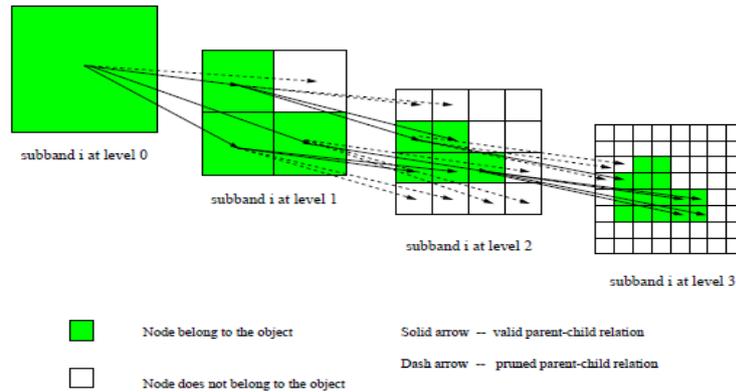


Fig.8: Parent child relation in OB-SPECK Algorithm

In this scheme [13], the shape information of objects area in the wavelet domain is integrated into the coding process. Similar to the object-based wavelet decomposition, the shape image is also decomposed into a pyramid of subbands, called the shape mask pyramid. In this way, the regions which belong to the object in each subband are known by both the encoder and the decoder. Each pixel of the shape mask has a 2-bit mask value: 1 bit is used to distinguish if the current wavelet coefficient is within the object, and the other bit is to tell if its child branch is within the object.

When the spatial orientation tree is constructed, which node and/or child branch are inside/outside the video object is known. Before the coding process, we prune the node and branch which are outside the video objects. During the sorting pass, those nodes and branches are not added into any list of LSP, LIP. Therefore, no information about these nodes and branches are transmitted. When the encoder and decoder scan these nodes and branches, they will be informed by the shape mask pyramid and skip over them.

IV. Results And Discussions

To evaluate the performance of the proposed method, we take 8-bit remote sensing images [18]. The original images and the ROI definitions are as shown in the Fig.9.

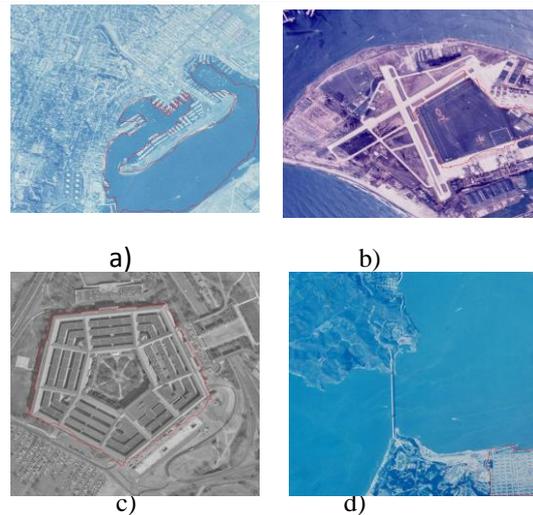


Fig.9: RSI and defined ROI is enclosed inside the red line

SADCT, SADWT and SARLT can be coded using same coding algorithm.i.e OBSPECK algorithm. Hence their efficiency can be compared directly. In lossy compression, peak signal to noise ratio (PSNR) is used as the objective performance criterion:

$$PSNR = \frac{\sum_{i=1}^T (x_{max})^2}{\sum_{i=1}^T (x_i - \hat{x}_i)^2}$$

where T is the total number of pixels inside the ROI, x_i and \hat{x}_i stand for the pixels in the original and the reconstructed image areas respectively. x_{max} is equal to 255 in 8-bit case.

Experiments of lossless compression on both ROI areas and entire images had been carried out in MATLABR2013a software. Results is listed in table shown below.

Table 1: Comparison of lossless compression performance (bitrate, in bits per pixel)

IMAGES	ROI AREA			ENTIRE IMAGE		
	SA-DCT	SA-DWT	SA-RLT	DCT	DWT	RTDLT
a	2.35	1.46	1.99	4.3	4.21	4.19
b	1.59	1.48	1.45	4.9	4.5	4.51
c	1.78	1.68	1.67	5.9	5.79	5.6
d	1.95	1.89	1.87	4.1	3.89	3.9

It can be seen that for entire image SA-RLT performs comparably to SA-DWT and even better than SA-DCT. But when considering ROI area SA-RLT is far even better than SA-DCT and SA-DWT. So from this we had SA-RLT is better when our region of interest is irregular in shape.

V. Conclusion

We have evaluated the performance of SA RLT based compression scheme for ROI coding of remote sensing images. Through a series of elementary matrix manipulations, a large class of lapped transforms can be constructed as a combination of pre-filtering and post-filtering in the current block-based DCT framework. The pre- and post-filter in this framework are the exact inverse of each other, and together with DCT, they form invertible lapped transforms with arbitrary amount of overlapping samples. This new framework provides several advantages. Existing block based infrastructure can be kept intact. Coding efficiency is improved by taking into account interblock spatial correlation in the pre-filtering stage. Blocking artifacts present in block transforms are eliminated. As a future work, we can extend 2D images to the hyper spectral images.

References

- [1] B.Penna, T.Tillo, E.Magli and G.Olmo, Progressive 3D coding of hyperspectral images based on JPEG 2000, *IEEE Geosci. remote sens.lett.*, vol 3, no.1, pp.125-126, Jan.2006
- [2] Mehnaz Tabassum, Mohammad Shorif Uddin, Extraction of ROI in Geographical Map Image, *Journal of Emerging Trends in Computing and Information Sciences*, Volume 2, No.5, May 2011, (2079-8407)
- [3] V.Srinivasa rao, Dr P.Rajesh Kumar, G.V.H.Prasad, M.Prema Kumar, S.Ravichand, Discrete Cosine Transform Vs Discrete Wavelet Transform: An Objective Comparison of Image Compression Techniques for JPEG Encoder, *Published in International Journal of Advanced Engineering and Applications*, Jan. 2010
- [4] C. Kwan, B. Li, R. Xu, X. Li, T. Tran, and T. Nguyen, A Complete Image Compression Scheme Based on Overlapped Block Transform with Post-processing, Intelligent Automation Inc..
- [5] Henrique S. Malvar, Biorthogonal and Nonuniform Lapped Transforms for Transform Coding with Reduced Blocking and Ringing Artifacts, *IEEE transactions on signal processing*, vol. 46, no. 4, April 1998.
- [6] W. C. Fong, S. C. Chan, Arumugam Nallanathan and K. L. Ho, Integer Lapped Transforms and Their Applications to Image Coding, *IEEE transactions on image processing*, vol. 11, no. 10, October 2002
- [7] Aldo Maalouf, Mohamed-Chaker Larabi, Low-Complexity Enhanced Lapped Transform for Image Coding in JPEG XR / HD Photo, *Proceedings of the 1998 IEEE International Conference*, Vol. 5, 1998, pp. 2565-2568.
- [8] Thomas Sikora and BClai Makai, Shape-Adaptive DCT for Generic Coding of Video, *IEEE transactions on circuits and systems for video technology*, vol. 5, no.1, February 1995
- [9] Hui-Cheng Hsu, Kun-Bin Lee, Nelson Yen-Chung Chang, and Tian-Sheuan Chang, Architecture Design of Shape-Adaptive Discrete Cosine Transform and Its Inverse for MPEG-4 Video Coding, *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 3, march 2008.
- [10] Shipeng Li and Weiping Li, Shape-Adaptive Discrete Wavelet Transforms for Arbitrarily Shaped Visual Object Coding, *IEEE transactions on circuits and systems for video technology*, vol. 10, no. 5, august 2000.
- [11] Marco Cagnazzo, Sara Parrilli, Giovanni Poggi, and Luisa Verdoliva, Costs and Advantages of Object-Based Image Coding with Shape-Adaptive Wavelet Transform, *EURASIP Journal on Image and Video Processing Volume 2007*, Article ID 78323.
- [12] Mnica penedo, William A.Pearlman, Pablo G.Tahoces, Miguel Souto, and Juan j.vidal, Region-based Wavelet coding Methods for digital mammography, *IEEE Transactions on medical imaging*, vol.22, no.10, October 2003.
- [13] K.P.Soman, K.I.Ramachandran and N.G.Resmi, *Insight into Wavelets: From theory to practice* (PHI learning private limited 2011).
- [14] Lei Wangn, Licheng Jiao, Jiayi Wu, Guang ming Shi, Yanjun Gong, Lossy-to-lossless image compression based on multiplier-less reversible integer time domain lapped transform, *International journal on Signal Processing: Image Communication* 2010, 622-632.
- [15] Masaaki Ikehara, Trac D. Tran, and Truong Q. Nguyen, A Family of Lapped Regular Transforms with Integer Coefficients, *IEEE transactions on signal processing*, vol. 50, no.4, April 2002
- [16] Trac D. Tran, Jie Liang, and Chengjie Tu, Lapped Transform via Time-Domain Pre- and Post-Filtering, *IEEE transactions on signal processing*, vol. 51, no. 6, June 2003
- [17] Alfred Mertins and Tanja Karp, Modulated, Perfect Reconstruction Filterbanks With Integer Coefficients, *IEEE transactions on signal processing*, vol. 50, no. 6, June 2002
- [18] <http://sipi.usc.edu/database/aerials>