Directionlets using in-phase lifting for image representation

Makur, Anamitra; Jayachandra, Dakala

2013

Jayachandra, D., & Makur, A. (2014). Directionlets Using In-phase Lifting For Image Representation. IEEE Transactions on Image Processing, 23(1), 240-249.

https://hdl.handle.net/10356/80050

https://doi.org/10.1109/TIP.2013.2288912

Directionlets Using In-phase Lifting For Image Representation

D Jayachandra *Student Member, IEEE,* and Anamitra Makur, *Senior Member, IEEE,*

***Abstract*— Directionlets allow construction of perfect reconstruction and critically sampled multi directional anisotropic basis, yet retaining the separable filtering of standard wavelet transform. Due to the spatially varying filtering and downsampling direction, it, however, is forced to apply spatial segmentation and process each segment independently. Because of this independent processing of the image segments, directionlets suffer from the following two major limitations when applied to, say, image coding. 1) Failure to exploit the correlation across block boundaries degrades the coding performance and also induces blocking artifacts, thus making it mandatory to use de-blocking filter at low bit rates. 2) Spatial scalability, i.e, minimum segment size or the number of levels of the transform, is limited due to independent processing of segments. We show that, with simple modifications in the block boundaries, we can overcome these limitations by, what we call, in-phase lifting implementation of directionlets. In the context of directionlets using in-phase lifting, we identify different possible groups of downsampling matrices that would allow construction of multi level transform without forcing independent processing of segments both with and without any modifications in the segment boundary. Experimental results in image coding show objective and subjective improvements when compared to the directionlets applied independently on each image segment. As an application, using both the in-phase lifting implementation of directionlets and the adaptive directional lifting, we have constructed an adaptive directional wavelet transform which has shown improved image coding performance over these adaptive directional wavelet transforms.**

***Index Terms*— Directionlets, Directional DWT, Directional lifting, Image representation, Image coding.**

I. Introduction

Standard separable wavelets are very successful for image representation over Fourier basis. However, due to its limited directionality and isotropic nature of the basis, the standard wavelet transform (WT) fails to exploit the correlation along the edges. Over the past decade or so, there has been lot of interest in developing directional transforms that can represent edges better than standard WT. The aim is to exploit the correlation along the edges by filtering and downsampling along the edge direction. Ideally, the goal is to build a critically sampled, multi-scale, and multi-directional transform. Critical sampling is important for image coding.

To go beyond standard WT for image representation, different directional transforms have been developed. Out of them, some are signal independent (non-adaptive) and some are signal dependent (adaptive). Most of the non-adaptive directional transforms, like curvelets [1] and contourlets [2], are over sampled. There are few critically sampled non-adaptive transforms, like the ones in [3], [4], but in these transforms, designing filter banks with good characteristics (for example with directional vanishing moments along arbitrary direction) is a nontrivial task. On the other hand, there are signal adaptive directional transforms that are critically sampled, multi-directional and multi-scale. Prominent examples in this class include orientation adaptive WT [5], directionlets [6], [7], directional lifting [8], [9], curved wavelets [10], etc. Most of the adaptive directional wavelet transforms that are based on directional lifting, like the ones in [8], [9], [10] and many more, keep the down sampling pattern same as that of standard WT, i.e, vertical downsampling followed by horizontal downsampling or vice-versa, and vary the filtering direction locally. By adapting the filtering direction to the local content direction as close as possible, these transforms try to contain the signal in the low pass approximation as much as possible. However, due to the possible mismatch between the downsampling and the filtering directions, these transforms may suffer from aliasing. On the other hand, orientation adaptive WT [5] and directionlets [6], [7] are the transforms that apply both filtering and downsampling along the edge direction. The orientation adaptive WT [5] allows filtering and downsampling along any two arbitrary directions using an invertible re-sampling involving interpolation of pixels at arbitrary locations, whereas, directionlets [6] allow filtering and downsampling along any two arbitrary rational directions by applying 1-D WT along the lines defined on integer lattices. Directionlets don't involve any interpolation. Both these conceptually similar methods apply spatially varying re-sampling followed by separable filtering, and hence, are forced to process each image segment independently. Recent works on directionlets focus on its application in solving different image processing problems like despeckling [11], watermarking [12], enhancement [13], fusion [14], etc. In this work, we study the limitations of directionlets due to independent processing of image segments, and give a lifting based implementation to overcome them.

Notations: Matrices and vectors are denoted by boldfaced letters. $\mathbf{M}^T$ represents the transpose of the matrix $\mathbf{M}$, and $diag\{a,b\} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$.

II. Background review and Motivation

A. Directionlets

In the following, we briefly review the construction of directionlets using the digital lines defined on integer lattices.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. E-mail: jaya0019@e.ntu.edu.sg and eamakur@ntu.edu.sg, respectively.

Let us say we want to apply 1-D WT along two rational directions, with slopes $r_1 = (b_1/a_1)$ and $r_2 = (b_2/a_2)$, where a_i and b_i are integers. Denote the directions in vector form as $\mathbf{d}_1 = \begin{bmatrix} a_1 & b_1 \end{bmatrix}^T$ and $\mathbf{d}_2 = \begin{bmatrix} a_2 & b_2 \end{bmatrix}^T$.

Directionlets apply 1-D WT on the co-lines generated by the full rank integer lattice Λ. The integer lattice Λ can be represented using the non-unique generating matrix $\mathbf{M}_\Lambda = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$. The pixel locations on the integer lattice are obtained as a linear combination of the direction vectors $\mathbf{d}_1$ and $\mathbf{d}_2$, where the weights are also integers. Given $\mathbf{M}_\Lambda$, the cubic integer lattice $\mathbb{Z}^2$ can be partitioned into $|det(\mathbf{M}_\Lambda)|$ number of cosets of the lattice Λ, where each coset is determined by the shift vector $\mathbf{s}_k \in \mathbb{Z}^2$ for $k = 0, 1, ..|det(\mathbf{M}_\Lambda)| - 1$. The integer lattice Λ and its cosets, partition the digital lines with slopes r_1 and r_2 into co-lines with the same slopes, respectively. The pixels in the co-lines with slopes r_1 and r_2 can be obtained using Eq.1 below.

$$\begin{bmatrix} i \\ j \end{bmatrix} = c_1 \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} + c_2 \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} + \begin{bmatrix} s_{k,1} \\ s_{k,2} \end{bmatrix} \tag{1}$$

On each coset determined by the shift vector $\mathbf{s}_k$, varying $c_1 \in \mathbb{Z}$ for each $c_2 \in \mathbb{Z}$ in Eq.1 gives the co-lines along r_1 (denoted as $CL_{\mathbf{s}_k}(r_1, n)$), and varying $c_2 \in \mathbb{Z}$ for each $c_1 \in \mathbb{Z}$ in Eq.1 gives the co-lines along r_2 (denoted as $CL_{\mathbf{s}_k}(r_2, n)$). These lines are free from any rounding operation, hence a pixel to next pixel distance is the same for all the pixels on these co-lines. Applying 1-D WT (filtering and downsampling) on the co-lines $CL_{\mathbf{s}_k}(r_1, n)$ will not disturb the edge coherence along the co-lines $CL_{\mathbf{s}_k}(r_2, n)$. This means, the retained pixels after downsampling along $CL_{\mathbf{s}_k}(r_1, n)$ belong to the same co-line $CL_{\mathbf{s}_k}(r_2, n)$ as they were before downsampling. To summarize, directionlets apply 1-D WT (filtering and downsampling) on the colines $CL_{\mathbf{s}_k}(r_1, n)$ and $CL_{\mathbf{s}_k}(r_2, n)$ separately on each coset of the integer lattice Λ represented by $\mathbf{M}_\Lambda$.

To construct anisotropic (elongated) directional basis, directionlet applies different number of 1-D WT along the directions $\mathbf{d}_1$ and $\mathbf{d}_2$ at each level in the 2-D WT. The resulting transform is referred as Skewed Anisotropic WT denoted by S-AWT($\mathbf{M}_\Lambda, n_1, n_2$), where n_1 and n_2 are the number of levels the 1-D WT is applied along the directions r_1 and r_2, respectively. In this work, we limit our study to the case where $n_1 = n_2 = 1$, and denote the resulting Skewed-Isotropic WT as S-IWT($\mathbf{M}_\Lambda$).

B. Motivation

As the directionality of edges in images is a local feature, to apply directionlets, the given image is divided into non-overlapping segments with edges in one or two dominant directions in each segment. Then for each segment, multi level S-IWT (or S-AWT in general) is applied along the two chosen transform directions of the segment. In essence, directionlet applies spatially varying downsampling and fixed separable 1-D filtering. As the downsampling pattern is varying across the segments, in the direct implementation of the directionlets [6], each image segment is processed independent of others. Therefore, the correlation across the segment boundaries cannot be utilized in the transform, which in turn will affect the image coding performance. Further, independent processing of segments leads to blocking artifacts at low bit rate image coding, which is not desirable. Also, the maximum number of levels in S-IWT is limited by the segment size, which in turn limits the segmentation process, and hence possibly affects the coding performance. In this work, using lifting factorization of filter banks, we aim to construct multilevel directionlets (particularly the S-IWT) without forcing independent processing of segments.

Lifting factorization [15] of filter banks, due to its structural perfect reconstruction (PR) property, added great flexibility to build space / time varying transforms. There are many adaptive transforms based on lifting, for example the directional lifting in [8], [9], lifting based spatial switching between arbitrary filter banks in [16], and many more, which adapt either the direction of filtering or the filter coefficients, but keep the downsampling pattern unchanged across the image segments. The authors in [17] built directional lifting with downsampling pattern that varies spatially between Row-Col, Col-Row and Quincunx. To handle the transition between the down sampling patterns, an interpolation based phase completion process is used. Inspired by the work in [17], to support filtering across the segments in directionlets, we conceptualize the lifting implementation of directionlets by separating the spatially varying downsampling and the invariant lifting steps of the 1-D filter bank, and call it as "in-phase lifting." As will be shown later, in-phase lifting implementation takes care of the transition between different sampling patterns by default. We propose to estimate missing polyphase components at any location from the available neighboring polyphase components to avoid usage of faraway pixels during the transition, similar to what is suggested in [17]. Within the directionlets, not all sampling matrices would allow switching between them without forcing independent processing of segments. Hence, within the scope of the in-phase lifting, we also determine different possible groups of sampling matrices that would allow construction of multi level directionlets without forcing independent processing of segments, both with and without the need to estimate any pixel during the transition.

Rest of the paper is organized as follows. The in-phase lifting implementation of directionlets is discussed in Section III and the scope of in-phase lifting in Section IV. In Section V, with in image coding, the advantages of in-phase lifting implementation of directionlets over direct implementation is discussed first, and then, a new adaptive directional wavelet transform based on in-phase lifting implementation of directionlets is constructed and its image coding performance is discussed. Finally, Section VI concludes the paper.

III. In-phase lifting

Directionlets, as discussed, apply spatially varying downsampling pattern while keeping the filters fixed. On the other hand, most of the lifting based adaptive transforms for image representation, adapts the filters to the image local behavior, while keeping the downsampling pattern fixed. In this section, using the lifting implementation of directionlets, we study the possibility of achieving spatially varying downsampling pattern, yet, not forcing independent processing of image segments.

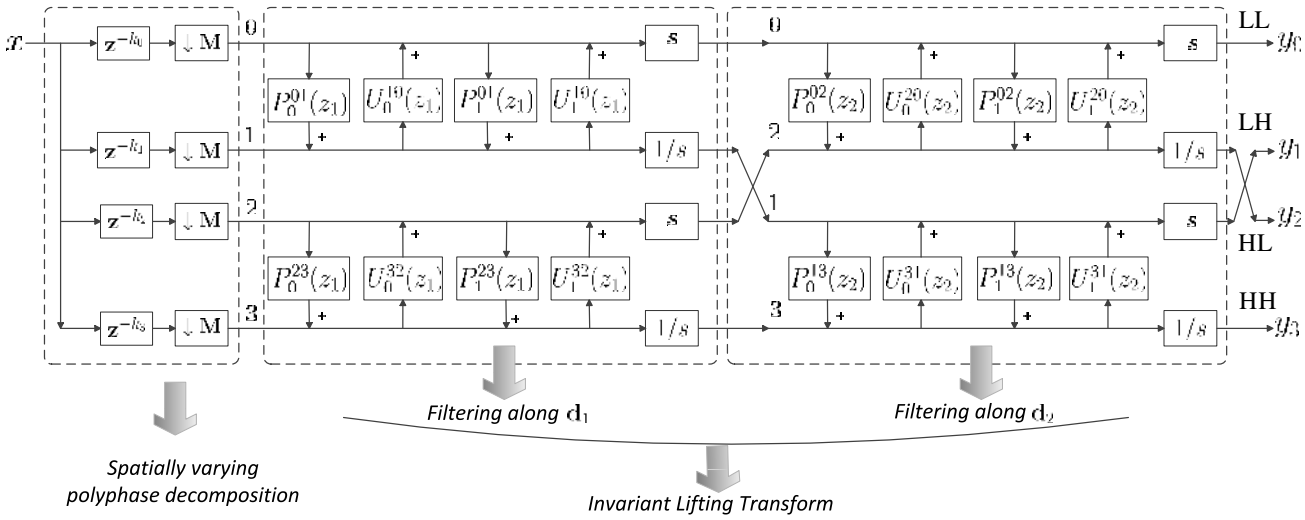


Fig. 1: Separable lifting implementation of 1-level of S-IWT($\mathbf{M}_\Lambda$), with $|det(\mathbf{M}_\Lambda)| = 1$.

A. Directionlets using lifting

First we discuss the lifting implementation of directionlets. To simplify the discussion we consider the direction vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ such that $|det(\mathbf{M}_\Lambda)| = 1$. One useful subset of matrices with $|det(\mathbf{M}_\Lambda)| = 1$ are of the form $\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}$, where $a \in \mathbb{Z}$. Varying a leads to rich subset of transform direcctions. For such examples, reader may refer to the image coding application in Section V. If $|det(\mathbf{M}_\Lambda)| > 1$, the cubic integer lattice $\mathbb{Z}^2$ is divided into $|det(\mathbf{M}_\Lambda)|$ number of cosets, and the same lifting based transform may be applied on each coset separately. Also we limit to the construction of S-IWT($\mathbf{M}_\Lambda$). The proposed lifting implementation that we are going to discuss can be easily extended to the construction of S-AWT($\mathbf{M}_\Lambda, n_1, n_2$). After 1-level of S-IWT($\mathbf{M}_\Lambda$) the resulting downsampling matrix will be given by $\mathbf{M} = \mathbf{D}_s\mathbf{M}_\Lambda$, where $\mathbf{D}_s = diag\{2, 2\}$.

Fig.1 shows the lifting implementation of 1-level of S-IWT($\mathbf{M}_\Lambda$) where a FB with 2 predict and 2 update steps followed by scaling is used. In Fig.1, the predict (or the update) lifting step $P_i^{jl}(\mathbf{z})$ (or $U_i^{jl}(\mathbf{z})$) is the i^{th} predict (or update) lifting step of the chosen 1-D FB applied from polyphase component j to polyphase component l. First, the given image segment is decomposed into polyphase components according to its chosen generating matrix $\mathbf{M}_\Lambda$. We number the polyphase components as $0, 1, 2$, and 3 in such a way that the 1-D WT (and hence its lifting steps) along $\mathbf{d}_1$ is always applied between the polyphase components $0, 1$ and $2, 3$, and the 1-D WT along $\mathbf{d}_2$ is always applied between the polyphase components $0, 2$ and $1, 3$. Accordingly, the resulting subbands are LL, LH, HL, and HH. The same transform can be repeated on the resulting LL subband to construct multilevel decomposition of the given segment.

As an example, Fig.2 shows both the polyphase decomposition and the corresponding transform directions for $\mathbf{M}_{\Lambda_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{M}_{\Lambda_1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. After applying 1-level of S-IWT($\mathbf{M}_\Lambda$), the polyphase components 0, 1, 2 and 3, denoted by the symbols ***circle, triangle, star*** and ***square***, will become LL, LH, HL and HH subbands, respectively. We call the resulting subband association of pixels as *subband pattern*. The double sided arrows indicate the filtering direction. Note that in this example, though the resulting downsampling matrices $\mathbf{M}_0$ and $\mathbf{M}_1$ represent the same downsampling pattern, i.e, the LL band coefficients of $\mathbf{M}_0$ and $\mathbf{M}_1$ are on the same downsampling grid, they lead to different subband patterns. All the generating matrices with $|det(\mathbf{M}_\Lambda)| = 1$ will lead to the same down sampling pattern. In general, different transform directions may lead to the same or different downsampling patterns.

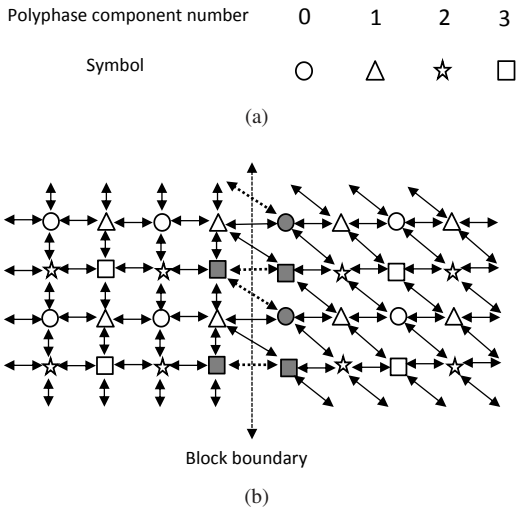


Fig. 2: (a) Symbols used to denote polyphase components $0, 1, 2$ and 3, which will respectively become LL, LH, HL and HH subbands. (b) Resulting polyphase components (Subband patterns) and the transform direction of the generating matrices $\mathbf{M}_{\Lambda_0}$ and $\mathbf{M}_{\Lambda_1}$, showing the interaction across the block boundary during the change from $\mathbf{M}_{\Lambda_0}$ to $\mathbf{M}_{\Lambda_1}$.

Our aim is to switch from one generating matrix to other without forcing independent processing of image segments. Consider switching from $\mathbf{M}_{\Lambda_0}$ to $\mathbf{M}_{\Lambda_1}$. As shown in Fig.2(b), the difference between the resulting subband patterns of $\mathbf{M}_{\Lambda_0}$ and $\mathbf{M}_{\Lambda_1}$ is that the pattern of the polyphase components $2, 3$ with respect to the polyphase components $0, 1$ is different. With such different subband patterns and different filtering directions, the resulting interaction across the block boundary is also shown in Fig.2(b). Clearly, there is phase incoherence during the change from $\mathbf{M}_{\Lambda_0}$ to $\mathbf{M}_{\Lambda_1}$, meaning, at some pixel locations across the boundary, in place of an expected polyphase component, another component is encountered. For example, while modifying the polyphase component 3 (square pixels) using the polyphase component 2 (star pixels) along horizontal direction, from either sides of the block boundary polyphase component 2 (star pixels) is expected, but polyphase component 3 (square pixels) is seen instead. In Fig.2(b), the shaded pixels are the ones experiencing phase incoherence whereas the dotted double sided arrows depict the filtering locations where there is phase incoherence. In general, switching between arbitrary generating matrices can lead to different scenarios of such phase incoherence.

B. In-phase lifting

To allow spatial switching between arbitrary transform directions (and hence the generating matrices) in directionlets, first we separate the polyphase decomposition and the lifting based filtering. As shown in Fig.1, the lifting based 1-D filtering is independent of the polyphase decomposition. Once the given image is divided into segments and optimal directions are chosen for each segment, then the entire image is divided into polyphase components according to the spatially varying generating matrices. The invariant lifting transform is applied on the resulting polyphase components. We define *in-phase lifting*, as the process of applying invariant lifting transform on a fixed set of polyphase components, in spite of the fact that the polyphase components from each segment are defined according to different generating matrices. Fig.3 depicts the concept of in-phase lifting.

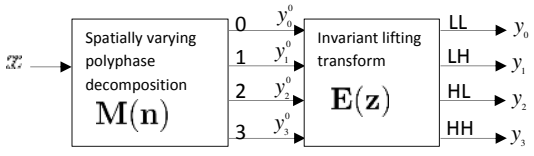


Fig. 3: In-phase lifting: Spatially varying polyphase decomposition followed by invariant lifting transform.

Formally, given the 2D signal $x(\mathbf{n})$, the spatially varying polyphase components $y_i^0(\mathbf{n})$, for $i = 0, 1, 2, 3$, are given by

$$y_i^0(\mathbf{n}) = x(\mathbf{M}(\mathbf{n})\mathbf{n} + \mathbf{k}_i), \quad \mathbf{k}_i \in \mathcal{N}(\mathbf{M}(\mathbf{n})). \tag{2}$$

Here $\mathcal{N}(\mathbf{M}(\mathbf{n}))$ denotes the set of all integer vectors $\mathbf{M}(\mathbf{n})\mathbf{x}$, with $\mathbf{x} \in [0, 1)$. Note that the downsampling matrix $\mathbf{M}$ is varying with space index $\mathbf{n}$. Let $\mathbf{Y}^0(\mathbf{z}) = [Y_0^0(\mathbf{z}) \quad Y_1^0(\mathbf{z}) \quad Y_2^0(\mathbf{z}) \quad Y_3^0(\mathbf{z})]^T$, where $Y_i^0(\mathbf{z})$ is the $\mathbf{z}$ transform of the polyphase component $y_i^0(\mathbf{n})$. Similarly, let $\mathbf{Y}(\mathbf{z}) = [Y_0(\mathbf{z}) \quad Y_1(\mathbf{z}) \quad Y_2(\mathbf{z}) \quad Y_3(\mathbf{z})]^T$, where $Y_i(\mathbf{z})$ is the $\mathbf{z}$ transform of the resulting subband signal $y_i(\mathbf{n})$. Now the invariant lifting transform, represented using the polyphase matrix $\mathbf{E}(\mathbf{z})$, applied on the polyphase components $y_i^0(\mathbf{n})$ can be expressed as

$$\mathbf{Y}(\mathbf{z}) = \mathbf{E}(\mathbf{z})\mathbf{Y}^0(\mathbf{z}) \tag{3}$$

The resulting spatially varying filters $\mathbf{H}(\mathbf{z}, \mathbf{n}) = [H_{LL}(\mathbf{z}, \mathbf{n}) \quad H_{LH}(\mathbf{z}, \mathbf{n}) \quad H_{HL}(\mathbf{z}, \mathbf{n}) \quad H_{HH}(\mathbf{z}, \mathbf{n})]^T$ are given by

$$\mathbf{H}(\mathbf{z}, \mathbf{n}) = \mathbf{E}(\mathbf{z}^{\mathbf{M}(\mathbf{n})})\mathbf{d}_{\mathbf{M}(\mathbf{n})}(\mathbf{z}) \tag{4}$$

where $\mathbf{d}_{\mathbf{M}(\mathbf{n})}(\mathbf{z})$ is the delay vector.

In-phase lifting makes sure that all the polyphase components are available even across the segment boundaries, and hence, solves the problem of phase incoherence. For the pixels well within a segment, in-phase lifting leads to filtering along the chosen transform directions of that segment. For the pixels across the segment boundaries, depending on the generating matrices, one of the following two scenarios can be seen. In one case, both the generating matrices may lead to the same subband pattern so that filtering across the segments is by default seamless, refer to Fig.5. In Section IV, we will identify all such groups of matrices. In the other case when the generating matrices lead to different subband patterns, across the segment boundaries some of the polyphase components may be coming from a geometrically different location than that of the same polyphase components that are well within the segments. Note that only the same polyphase components are used in in-phase lifting. However, they may be spatially far away pixels. To avoid this, we propose to estimate the pixel of required polyphase component as the average of the available pixels of the same polyphase component within 3×3 neighborhood, similar to the one proposed in [17].

The advantage of in-phase lifting implementation of directionlets over direct implementation will be experimentally validated within image coding application in Section V.

IV. Scope of In-phase lifting

In this section we analyze the scope of in-phase lifting within directionlets. We identify all the generating matrices that would allow switching between them, with and without any modifications in the segment boundaries, and would allow construction of multilevel transform, yet without forcing independent processing of segments. The following analysis considers all the generating matrices with $|det(\mathbf{M}_\Lambda)| \geq 1$. Throughout this section consider the case where we want to transform the spatially neighboring segments, say i and j, using the generating matrices, say $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$, respectively.

A. Switching between arbitrary generating matrices

As mentioned earlier, the directionlets apply transform separately on each coset of the integer lattice given by the chosen generating matrix, and by definition no interaction is allowed between the cosets. To switch between any two generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ using in-phase lifting, both the generating matrices should lead to the same number of subbands, and hence to the same number of cosets. Hence, we

can conclude that having $|det(\mathbf{M}_{\Lambda_i})| = |det(\mathbf{M}_{\Lambda_j})|$ is a necessary and sufficient condition to be able to switch between $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ in one level of the transform.

B. Possibility of constructing multilevel transform

Ability to switch between $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ in one level of the directionlet doesn't guarantee the possibility of constructing multilevel transform. To understand the required conditions on $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ to get multilevel transform, first we need to relate the resulting subbands from using any given generating matrix $\mathbf{M}_\Lambda$ with the cosets of the integer lattice given by the resulting downsampling matrix $\mathbf{M}$. It is well known that in a critically sampled filter bank, the cosets of the integer lattice given by the downsampling matrix are the polyphase components, and they will be transformed to subbands. The above result is true even in the case of directionlets. We state the result as a fact.

Fact 1: Given any arbitrary generating matrix $\mathbf{M}_\Lambda = \begin{bmatrix}\mathbf{d}_1 & \mathbf{d}_2\end{bmatrix} = \begin{bmatrix}a_1 & a_2\\ b_1 & b_2\end{bmatrix}$, after 1-level of S-IWT($\mathbf{M}_\Lambda$), the retained coefficients in each subband lie on the cosets of the integer lattice given by the downsampling matrix $\mathbf{M} = \begin{bmatrix}2\mathbf{d}_1 & 2\mathbf{d}_2\end{bmatrix}$, and the offset vectors of the cosets containing the LL, LH, HL and HH subbands are $\mathbf{k}_\mathbf{i}' = \mathbf{k_i}$, $\mathbf{k}_\mathbf{i}' = \mathbf{d}_1 + \mathbf{k_i}$, $\mathbf{k}_\mathbf{i}' = \mathbf{d}_2 + \mathbf{k_i}$, and $\mathbf{k}_\mathbf{i}' = \mathbf{d}_1 + \mathbf{d}_2 + \mathbf{k_i}$, respectively, where $\mathbf{k_i} \in \mathcal{N}(\mathbf{M}_\Lambda)$ for $i = 0, 1, ...|det(\mathbf{M}_\Lambda)| - 1$ are the offset vectors of the cosets of $\mathbf{M}_\Lambda$.

Using the division theorem for integer vectors [19], it is straightforward to verify that the offset vectors given in *Fact 1* are indeed the offsets vectors of the cosets of the integer lattice given by the downsampling matrix $\mathbf{M}$. Note that there are $|det(\mathbf{M}_\Lambda)|$ number of LL, LH, HL and HH subbands coming from $|det(\mathbf{M}_\Lambda)|$ number of cosets of $\mathbf{M}_\Lambda$. Also note that even in the case of S-AWT($\mathbf{M}_\Lambda$, n_1, n_2), *Fact 1* is still valid, except that 1-level of such directionlet transform will lead to more than 4 subbands on each coset of $\mathbf{M}_\Lambda$.

Now, consider any two generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ such that $|det(\mathbf{M}_{\Lambda_i})| = |det(\mathbf{M}_{\Lambda_j})| = K \geq 1$. Let us denote the cosets of the integer lattice given by $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$, respectively as, $Coset_k^i$ and $Coset_k^j$, where k is the coset number and i, j are the segment indices. On each coset the transform needs to be applied separately. To construct multilevel directionlets, LL subband of $Coset_k^i$ and $Coset_k^j$ should be on the same sampling grid. For this to happen, the generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ should represent the same downsampling pattern. Note that any two matrices representing the same downsampling pattern are related by $\mathbf{M}_{\Lambda_i} = \mathbf{M}_{\Lambda_j}\mathbf{U}$, where $\mathbf{U}$ is an unimodular matrix [19]. In this case, the cosets can be ordered from $k = 0$ to $k = K - 1$ such that $Coset_k^i$ and $Coset_k^j$ represent the same partitioning of the sampling grid, except with the possible difference in the coset offsets of the respective generating matrices. On each coset, LL subband coefficients start from the coset offset. As the offset of the $Coset_0$ of both the matrices is the same, the LL subband of $Coset_0^i$ and $Coset_0^j$ is on the same sampling grid. The LL subbands on the other cosets may be on different sampling grids depending on the coset offset. To allow multilevel decomposition, chose $\mathbf{M}_{\Lambda_i}$, say, as the primary generating matrix, and use the offset of $Coset_k^i$ to determine the LL subband of $Coset_k^j$ for all k. This will ensure LL subband of $Coset_k^i$ and $Coset_k^j$ is on the same sampling grid for all k. Hence, multilevel directionlets can be constructed using in-phase lifting. On the other hand, if $\mathbf{M}_{\Lambda_i} \neq \mathbf{M}_{\Lambda_j}\mathbf{U}$, then $Coset_k^i$ and $Coset_k^j$ are not the same, hence, in-phase lifting fails to construct multilevel transform without forcing independent processing of segments. We summarize the above discussion in the following theorem, and omit the proof as it is obvious from the above discussion.

Theorem 1: In-phase lifting allows construction of multilevel directionlets with switching between any two generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$, provided $|det(\mathbf{M}_{\Lambda_i})| = |det(\mathbf{M}_{\Lambda_j})|$, and $\mathbf{M}_{\Lambda_i} = \mathbf{M}_{\Lambda_j}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$.

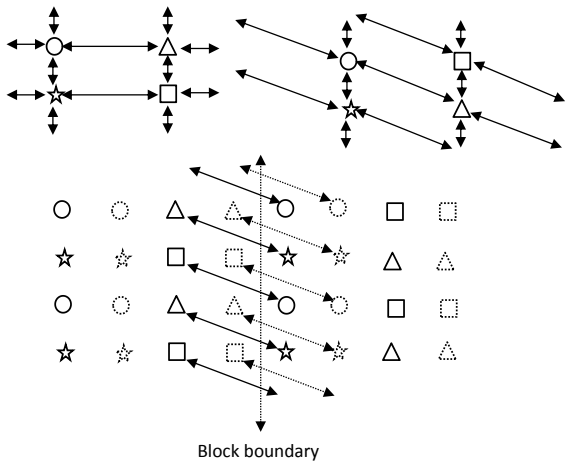


Fig. 4: Top Row: From left to right, indicates the transform directions and the resulting subband patterns of the generating matrices $\mathbf{M}_{\Lambda_2}$ and $\mathbf{M}_{\Lambda_3}$. Bottom row: Shows the subband patterns on both $Coset_0$ (symbols with solid lines) and $Coset_1$ (symbols with dotted lines) along with the locations (indicated by the double sided arrows) of phase incoherence. Notice that starting the LL subband from the same offset vector on $Coset_1$ leads to the same sub sampling pattern of LL subband on both segments, hence, allows iteration of the transform on LL subband.

For any given downsampling ratio ($|det(\mathbf{M}_\Lambda)|$), some distinct downsampling patterns are possible [20]. Each distinct downsampling pattern can be represented by infinitely many matrices. *Theorem* 1 implies that unless otherwise the given generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ represent distinct downsampling patterns, in-phase lifting enables construction of multilevel directionelets without forcing independent processing of the segments. As an example, let us consider that we want to transform the segments i and j using the generating matrices $\mathbf{M}_{\Lambda_2} = \begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$ and $\mathbf{M}_{\Lambda_3} = \begin{bmatrix}2 & 0\\ 1 & 1\end{bmatrix}$, respectively. Here, $|det(\mathbf{M}_{\Lambda_2})| = |det(\mathbf{M}_{\Lambda_3})| = 2$ and $\mathbf{M}_{\Lambda_3} = \mathbf{M}_{\Lambda_2}\mathbf{U}$, where the unimodular matrix $\mathbf{U} = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$. With determinant of 2, 3 distinct downsampling patterns are possible [20]. Both $\mathbf{M}_{\Lambda_2}$ and $\mathbf{M}_{\Lambda_3}$ represent horizontal downsampling pattern, at the same time supporting different transform directions. The offset vectors of $Coset_1^2$ and $Coset_1^3$ are $\begin{bmatrix}1 & 0\end{bmatrix}^T$ and $\begin{bmatrix}1 & 1\end{bmatrix}^T$, respectively. According to the *Theorem* 1, starting the LL

subband of $Coset_1^2$ and $Coset_1^3$ from either one of these offset vectors would allow construction of multilevel transform using in-phase lifting. Fig.4 shows the transform directions and the resulting subband patterns when the LL subband of $Coset_1^2$ and $Coset_1^3$ is started from the offset vector $\begin{bmatrix}1 & 0\end{bmatrix}^T$. Notice the same sub sampling pattern of LL subband on both segments. With that and handling the phase incoherence (at the locations indicated by double sided arrows in Fig.4) on both the cosets using in-phase lifting, allows the transform to iterate on the resulting LL subbands without forcing independent processing of segments.

In the case of $|det(\mathbf{M}_\Lambda)| = 1$, only one distinct downsampling matrix is possible. Although this case looks constrained, quite a number of useful transform directions are possible, and the same will be shown in the image coding application in Section V.

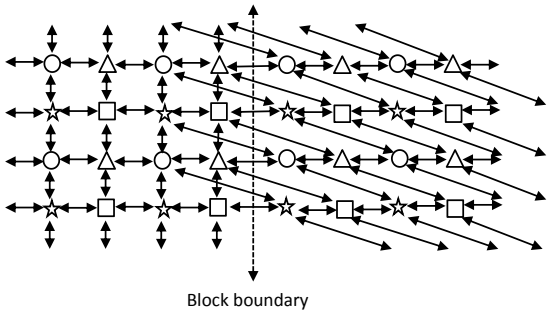


Fig. 5: Interaction across the block boundary during the change from $\mathbf{M}_{\Lambda_0}$ to $\mathbf{M}_{\Lambda_4}$. As the subband patterns are the same, filtering across the block boundary is seamless.

C. Enumeration of subband patterns

Within the matrices representing the same downsampling pattern, there are matrices that would lead to the same subband pattern when used in directionlets. For example, Fig.5 shows switching between $\mathbf{M}_{\Lambda_0} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ and $\mathbf{M}_{\Lambda_4} = \begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix}$. As can be seen, with both the sampling patterns leading to the same subband pattern of pixels, filtering across the segment boundary is by default seamless (no phase incoherence). It is easy to show that all the generating matrices with $|det(\mathbf{M}_\Lambda)| = 1$ can be classified into 6 different groups based on the resulting subband patterns. Fig.6 shows the 6 possible groups. If $|det(\mathbf{M}_\Lambda)| > 1$, the resulting subband patterns of pixels on any of its cosets belong to one of the 6 groups in Fig.6, and the subband pattern will be the same on all the cosets.

Given that $\mathbf{M}_{\Lambda_i} = \mathbf{M}_{\Lambda_j}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$, one can verify if they lead to the same subband pattern or not. One way is to verify if the cosets containing the resulting subbands of segment i and j are the same using the offset vectors in *Fact 1* and the division theorem for integer vectors [19]. Instead, here we give a simpler way to verify the same. On any one of the cosets of the integer lattice of $\mathbf{M}_{\Lambda_i}$ (or $\mathbf{M}_{\Lambda_j}$), let us denote the horizontal and vertical distance (in pixels) between the horizontally and vertically neighboring pixels in a subband as s_h and s_v, respectively. Denote them in vector form as $\mathbf{s_d} = \begin{bmatrix}s_h & s_v\end{bmatrix}^T$. Notice that as $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ represent the same downsampling pattern, resulting $\mathbf{s_d}$ is the same for both $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$. With that, *Lemma 1* below describes a way to verify if they lead to the same subband pattern, and the proof is omitted as it is simple to verify.

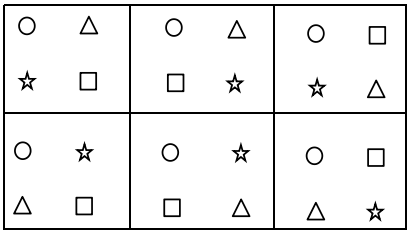


Fig. 6: The six different possible subband grouping of pixels using the generating matrices with $|det(\mathbf{M}_\Lambda)| = 1$. Switching between any two generating matrices leading to the same subband patterns needs no modification across the block boundary.

Lemma 1: Given $\mathbf{M}_{\Lambda_i} = \mathbf{M}_{\Lambda_j}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$, $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$ lead to the same subband pattern if and only if $\mathbf{M}_{\Lambda_i} \, mod \, \mathbf{s_d} = \mathbf{M}_{\Lambda_j} \, mod \, \mathbf{s_d}$. Here, $\mathbf{M}_\Lambda \, mod \, \mathbf{s_d} = \begin{bmatrix}a_1 \, mod \, s_h & a_2 \, mod \, s_h\\ b_1 \, mod \, s_v & b_2 \, mod \, s_v\end{bmatrix}$.

As an example, the generating matrices $\mathbf{M}_{\Lambda_0}, \mathbf{M}_{\Lambda_1}$ and $\mathbf{M}_{\Lambda_4}$ lead to $\mathbf{s_d} = \begin{bmatrix}2 & 2\end{bmatrix}^T$ after 1-level of S-IWT. As expected, $\mathbf{M}_{\Lambda_0} \, mod \, \mathbf{s_d} \neq \mathbf{M}_{\Lambda_1} \, mod \, \mathbf{s_d}$, indicating different subband patterns (refer to Fig.2), and $\mathbf{M}_{\Lambda_0} \, mod \, \mathbf{s_d} = \mathbf{M}_{\Lambda_4} \, mod \, \mathbf{s_d}$, indicating the same subband patterns (refer to Fig.5). Further, the other generating matrices $\mathbf{M}_{\Lambda_2}$ and $\mathbf{M}_{\Lambda_3}$ lead to $\mathbf{s_d} = \begin{bmatrix}4 & 2\end{bmatrix}^T$, and $\mathbf{M}_{\Lambda_2} \, mod \, \mathbf{s_d} \neq \mathbf{M}_{\Lambda_3} \, mod \, \mathbf{s_d}$, indicating different subband patterns (refer to Fig.4).

D. Summary

Consider any two generating matrices $\mathbf{M}_{\Lambda_i}$ and $\mathbf{M}_{\Lambda_j}$. The scope of in-phase lifting for directionlets, discussed above, can be summarized as follows.

(1) If $|det(\mathbf{M}_{\Lambda_i})| = |det(\mathbf{M}_{\Lambda_j})|$, in-phase lifting allows switching between them atleast in one level of the transform.

(2) If both the generating matrices represent the same downsampling pattern, i.e, they satisfy $\mathbf{M}_{\Lambda_i} = \mathbf{M}_{\Lambda_j}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$, then, in-phase lifting allows switching between them in multiple levels of the transform.

(3) Apart from representing the same downsampling pattern, if $\mathbf{M}_{\Lambda_i} \, mod \, \mathbf{s_d} = \mathbf{M}_{\Lambda_j} \, mod \, \mathbf{s_d}$, then the generating matrices lead to the same subband pattern, and in-phase lifting allows switching between them in multiple levels of the transform without any modifications in the segment boundary.

V. Image coding

In this section, within image coding, first we show the effect of in-phase lifting implementation of directionlets over the direct implementation of directionlets. Then, we come up with an adaptive directional wavelet transform using both the in-phase lifting implementation of directionlets and the adaptive directional lifting, and compare its performance with respect to these directional transforms.

In all the transforms, we use the lifting factorization of the 9/7 filter bank used in JPEG2000 [18] as 1-D WT. All the transforms are iterated to 5-levels, and SPIHT [21] coder is used to code the transform coefficients.

A. *Effect of in-phase lifting implementation of directionlets*

Consider the following 12 direction vectors.

$$\begin{aligned}\mathbf{d}_0 = \begin{bmatrix}1 & 0\end{bmatrix}^T, \mathbf{d}_1 = \begin{bmatrix}3 & 1\end{bmatrix}^T, \mathbf{d}_2 = \begin{bmatrix}2 & 1\end{bmatrix}^T,\\ \mathbf{d}_3 = \begin{bmatrix}1 & 1\end{bmatrix}^T, \mathbf{d}_4 = \begin{bmatrix}1 & 2\end{bmatrix}^T, \mathbf{d}_5 = \begin{bmatrix}1 & 3\end{bmatrix}^T,\\ \mathbf{d}_6 = \begin{bmatrix}0 & 1\end{bmatrix}^T, \mathbf{d}_7 = \begin{bmatrix}-1 & 3\end{bmatrix}^T, \mathbf{d}_8 = \begin{bmatrix}-1 & 2\end{bmatrix}^T,\\ \mathbf{d}_9 = \begin{bmatrix}-1 & 1\end{bmatrix}^T, \mathbf{d}_{10} = \begin{bmatrix}-2 & 1\end{bmatrix}^T, \mathbf{d}_{11} = \begin{bmatrix}-3 & 1\end{bmatrix}^T. \end{aligned} \quad (5)$$

Using these 12 directions we form 13 generating matrices as follows. Directions $\mathbf{d}_0$ and $\mathbf{d}_6$ form the standard generating matrix $\mathbf{M}_{\Lambda_0} = \begin{bmatrix}\mathbf{d}_0 & \mathbf{d}_6\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$. The close to horizontal directions $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_{10}$ and $\mathbf{d}_{11}$ are paired with horizontal direction $\mathbf{d}_0$. The close to vertical directions $\mathbf{d}_4$, $\mathbf{d}_5$, $\mathbf{d}_7$ and $\mathbf{d}_8$ are paired with vertical direction $\mathbf{d}_6$. That completes 9 generating matrices. Each of the diagonal directions $\mathbf{d}_3$ and $\mathbf{d}_9$ paired with both horizontal and vertical directions, $\mathbf{d}_0$ and $\mathbf{d}_6$, forms the rest of the 4 generating matrices. Note that all the 13 generating matrices satisfy $|det(\mathbf{M}_\Lambda)| = 1$. Along the horizontal ($\mathbf{d}_0$) and vertical ($\mathbf{d}_6$) directions the sampling density is maximum (pixel to pixel distance is minimum) compared to the sampling density along the other directions. Though it is possible to construct matrices with $|det(\mathbf{M}_\Lambda)| = 1$ using arbitrary directions, we choose to keep $\mathbf{d}_0$ or $\mathbf{d}_6$ as one of the transform directions, particularly for image coding. Pairs of directions that would form multiple cosets ($|det(\mathbf{M}_\Lambda)| > 1$) are not considered for image coding to avoid the division of spatially adjacent pixels into different cosets [7]. Also, at every level of the transform only one level of 1-D transform is applied along each direction, i.e, we limit to S-IWT($\mathbf{M}_\Lambda$) for image coding.

For a given image segment, the transform directions and hence the generating matrix that minimizes the following Lagrangian cost function is selected as the best generating matrix,

$$\mathbf{M}^*_\Lambda = \arg\min_{\mathbf{M}_\Lambda}\left\{D(\mathbf{M}_\Lambda) + \lambda R(\mathbf{M}_\Lambda)\right\} \quad (6)$$

where $D(\mathbf{M}_\Lambda)$ is the distortion induced in coding the segment using the generating matrix $\mathbf{M}_\Lambda$. For the given image segment, 3-levels of S-IWT($\mathbf{M}_\Lambda$) is applied and the largest 5% of the coefficients are retained. Then the distortion is approximated with the absolute sum of the discarded transform coefficients. $R(\mathbf{M}_\Lambda)$ is the number of bits used to represent the selection of $\mathbf{M}_\Lambda$, and λ is the Lagrange multiplier chosen empirically ($\lambda = 40$) in our experiments.

In our experiments, transform coefficients of all the transforms are coded using SPIHT[1] coder [21] to get embedded bit stream. In case of direct implementation of directionlets, where each segment is transformed independent of others, the subbands from all the segments are re-arranged such that overall they give the same tree structure as that of the standard WT. Note that such re-arrangement doesn't affect the inter-scale relation of the transform coefficients, and hence allows to use SPIHT like tree coding.

[1]Space-Frequency quantization [7] can be used to optimize the transform and quantization for every bit rate to produce better results. But, as our main purpose is to show the improvements with in-phase lifting, we simply use SPIHT [21] coder to produce embedded bit stream.

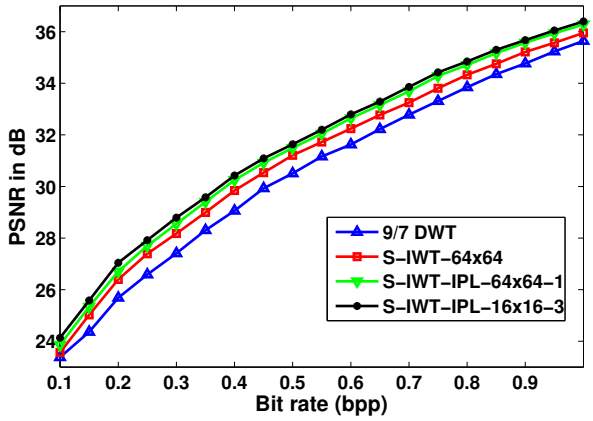
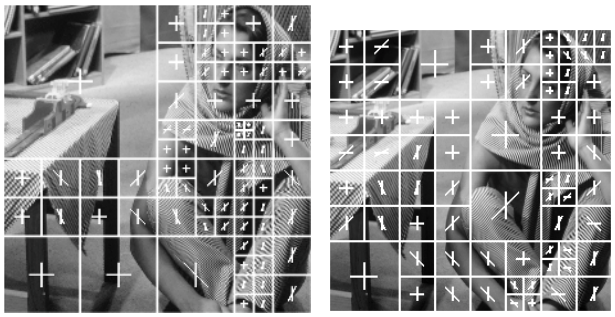


Fig. 7: Coding results of *Barbara* test image with different transforms using SPIHT coder. Notation: S-IWT-IPL-$m\times m$-n refers to the in-phase lifting implementation of S-IWT with a min block size of $m\times m$ in quad tree and direction selection at first n levels. S-IWT-$m\times m$ refers to the direct implementation of S-IWT. Clearly, S-IWT-IPL-16×16-3 out performs others.

In direct implementation of directionlets, minimum segment size is limited by the number of transform levels, and vice-versa. For a given image, quad tree segmentation upto the minimum block size 64×64 is applied. In the quad tree, the best sub division and its best coding mode is selected by comparing the Lagrangian cost in Eq.6 with the Lagrangian cost of sub dividing and coding each sub block with its best coding mode. On each of the resulting image segments a 5-level S-IWT is applied, independently. We denote this transform as S-IWT-64×64. The corresponding in-phase lifting implementation is denoted as S-IWT-IPL-64×64-1, the 1 at the end is used to denote that direction selection is performed only at level 1 of the transform. Fig.7 shows image coding results with these transforms for the test image *Barbara*. As expected, due to the ability to exploit the correlation across the segments, in-phase lifting implementation outperforms the direct implementation.

Compared to the independent processing of segments in direct implementation, in-phase lifting improves the spatial scalability of directionlets. Particularly, it allows direction selection at multiple levels and allows the use of smaller segments so that directionlets can be well adapted to the locally varying directionality in images. We implement a transform using a minimum block size of 16×16 and direction selection at first three levels. Denote the transform as S-IWT-IPL-16×16-3. For the transform levels from three and above, the directions selected at level three are used. Fig.8 shows the resulting segmentation along with the selected pair of directions at first two levels of the transform. The chosen



(a) (b)

Fig. 8: Direction selection at (a) level 1, and (b) level 2 using quad tree with minimum block size of 16×16.

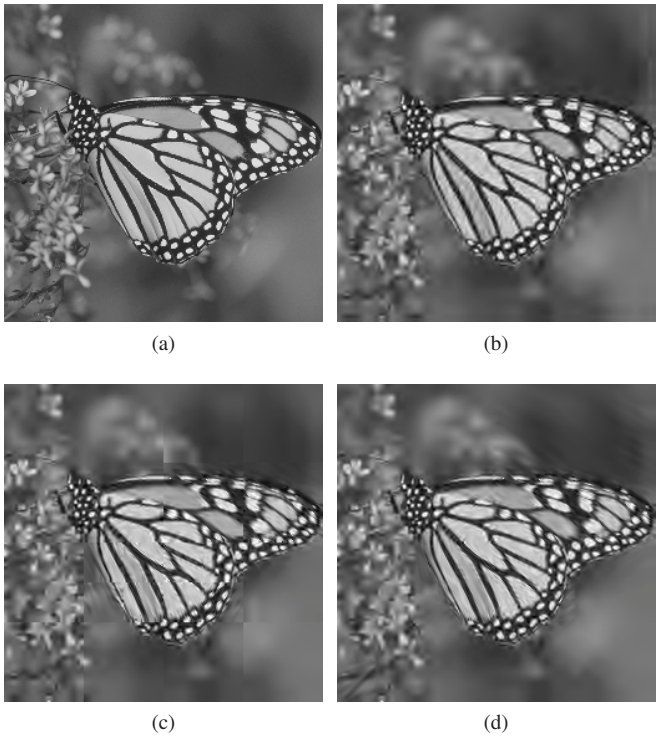
transform directions are well adapted to the local content direction. From the segmentation at level two, shown in Fig.8(b), we can observe that the transform directions are well adapted to the changes in the content orientation after 1-level of the transform. From the coding results in Fig.7, we can see that the direction selection at multiple levels and using block sizes upto 16×16 in the transform S-IWT-IPL-16×16-3 has improved the results as compared to the transform with direction selection only at level one in S-IWT-IPL-64×64-1.

(a) (b)

(c) (d)

Fig. 9: (a) Original *Barbara* image. Corresponding reconstructed images at 0.1 bpp with SPIHT coding of 5-level transforms of (b) 9/7 DWT (PSNR: 23.38 dB), (c) Direct implementation of directionlets, S-IWT-64×64 (PSNR: 23.56 dB), and (d) In-phase lifting implementation of directionlets, S-IWT-IPL-16×16-3 (PSNR: 24.13 dB).

From Fig.7, for *Barbara* image, which has strong directional content, S-IWT-IPL-64×64-3 gave around 1.4 dB PSNR gain over DWT and around 0.6 dB PSNR gain over S-IWT-64×64. Fig.11 shows the difference in PSNR of different directional transforms with respect to S-IWT-64×64. For *Lena*, *Monarch* and *Pentagon*, which have moderate directional content, S-IWT-IPL-16×16-3 consistently gave around 0.3 dB PSNR gain over DWT and S-IWT-64×64. Similar improvements have been observed for other test images as well. Fig.9 and Fig.10 shows the reconstructed *Barbara* and *Monarch* images at 0.1 bits per pixel (compression ratio of 80). Clearly, with in-phase lifting implementation, the blocking artifacts are eliminated and hence the visual quality is improved. With that, we can say the in-phase lifting implementation S-IWT-IPL-16×16-3, consistently outperforms S-IWT-64×64 and DWT both in terms of PSNR and visual quality.



(a) (b)

(c) (d)

Fig. 10: (a) Original *Monarch* image. Corresponding reconstructed images at 0.1 bpp with SPIHT coding of 5-level transforms of (b) 9/7 DWT (PSNR: 24.87 dB), (c) Direct implementation of directionlets, S-IWT-64×64 (PSNR: 24.70 dB), and (d) In-phase lifting implementation of directionlets, S-IWT-IPL-16×16-3 (PSNR: 25.08 dB).

B. Adaptive directional wavelet transform using in-phase lifting

Adaptive directional lifting [8], [9], denoted as ADL transform, is one of the most successful lifting based adaptive transforms for image coding. ADL, just like in the standard WT, applies vertical downsampling followed by horizontal downsampling or vice-versa. Then, instead of applying lifting steps just along vertical and horizontal directions, the lifting steps are applied along a locally optimal direction so that the correlation along the edge structure is exploited. Both ADL and directionlets have their own advantages. For a given local edge in some direction, by adapting the lifting direction, ADL tries to push the edge completely into the low pass approximation (LL band). However, due to the possible difference

between downsampling and filtering directions, ADL may suffer from aliasing. Directionlets, on the other hand, have the advantage of applying both downsampling and filtering along the same direction. However, as only one of the transform directions can be adapted to the edge direction, the edge cannot be completely pushed into the LL band. Intuitively, we can say that ADL may be good for unidirectional edges and directionlets may be good even for curved edges. As it is more probable to find unidirectional edges at the finer (top) levels of a multi level wavelet transform, ADL may be good for top levels. It has been observed that ADL [8], [9], applied only at first 2 levels of a transform has shown good coding results, which has also been observed in our experiments. To take advantage of both the in-phase lifting implementation of directionlets (S-IWT-IPL-16×16-3) and ADL, by appropriately combining both these transforms, we define a new transform denoted as ADL-S-IWT-IPL. In ADL-S-IWT-IPL, ADL is applied only for the first level, and S-IWT-IPL is applied for the rest with direction selection at only one level.

We consider ADL both with and without using pixels at fractional locations for prediction.

ADL-Fullpel: In vertical downsampling, except the directions $\mathbf{d}_0$, $\mathbf{d}_4$, $\mathbf{d}_8$, rest of the 9 directions in Eq.5 are used. Similarly, in horizontal downsampling, except the directions $\mathbf{d}_2$, $\mathbf{d}_6$, $\mathbf{d}_{10}$, rest are used. With limited number of directions, it may not be possible to push an edge completely into LL band. For example, for a diagonal edge, say along $\mathbf{d}_3$, lifting steps will be applied along the direction $\mathbf{d}_3$ during vertical downsampling, but after downsampling, the resulting edge direction $\mathbf{d}_2 = \begin{bmatrix}2 & 1\end{bmatrix}^T$ is not supported by the lifting directions in horizontal downsampling; lifting has to be applied along any one of the available directions.

ADL-Subpel: In vertical downsampling, the following 9 subpel directions $\begin{bmatrix}\pm 1 & 1\end{bmatrix}^T$, $\begin{bmatrix}\pm 3/4 & 1\end{bmatrix}^T$, $\begin{bmatrix}\pm 1/2 & 1\end{bmatrix}^T$, $\begin{bmatrix}\pm 1/4 & 1\end{bmatrix}^T$, and $\begin{bmatrix}0 & 1\end{bmatrix}^T$ are used. Similarly, in horizontal downsampling 9 subpel directions around the horizontal direction are used. We use the 6-tap interpolation filter of the H.264 video coding standard [22] to interpolate the pixels at subpel locations. For a diagonal edge, say in the direction $\begin{bmatrix}1 & 1\end{bmatrix}^T$, just as in ADL-Fullpel, lifting is applied along the same diagonal direction in vertical downsampling. Fortunately, the resulting direction $\begin{bmatrix}1 & 1/2\end{bmatrix}^T$ after downsampling is supported in horizontal downsampling. With that, we can say ADL with subpel directions can push the diagonal edges completely into the LL band. However, notice that edge directions close to horizontal are not supported in vertical downsampling, and vice-versa.

For both ADL-Fullpel and ADL-Subpel, we apply a quad tree segmentation with minimum block size of 16×16, similar to that applied for S-IWT-IPL-16 × 16-3. Direction selection is applied at first two levels of the 2-D transform, and for the rest, lifting is applied along the standard directions. For any given image segment, the lifting direction giving the minimum Lagrangian cost, defined in Eq.6, is selected as the best direction. The distortion D(.) in Eq.6 is calculated as the absolute sum of the coefficients in the high pass band after 1-level of 1-D transform with directional lifting.

ADL-S-IWT-IPL: ADL-Subpel is applied for the first level of the transform, expecting it to push unidirectional edges into the LL band. Then for the levels 2 to 5, in-phase lifting implementation of directionlets (S-IWT-IPL-16×16) is applied with direction selection at only level 2. Note that the transform directions selected at level 2 are applied to the levels 3 to 5 as well, and hence, S-IWT-IPL-16 × 16 is expected to give directional subband decomposition suitable for the local content direction.

For each of these directional transforms, statistics of the selected directions is obtained by measuring over 30 test images. The direction selection information is coded using an entropy coder designed with these statistics.

Fig. 11 shows the differential PSNR results of the above directional transforms with respect to the direct implementation of directionlets (S-IWT-64×64) for test images *Barbara, Lena, Monarch*, and *Pentagon*. As mentioned earlier, S-IWT-IPL-16×16-3 consistently performs better than S-IWT-64×64. Between ADL and S-IWT-IPL-16 × 16-3, both ADL-Fullpel and ADL-Subpel performs better for *Barbara*, which has strong directional edges. Whereas, S-IWT-IPL-16 × 16-3 performs better for images *Lena, Monarch*, and *Pentagon*, which have moderate edges. The new adaptive directional wavelet transform, ADL-S-IWT-IPL, with the addition of ADL-Subpel at level one, for *Barbara* image, shows comparable performance with respect to ADL-Subpel and significant improvements over S-IWT-IPL-16 × 16-3. For other images, ADL-S-IWT-IPL shows improved performance over both ADL-Subpel and S-IWT-IPL-16 × 16-3.

ADL-S-IWT-IPL shows the application of in-phase lifting implementation of directionlets in the context of adaptive directional wavelets for image coding, and it may encourage further development of better adaptive directional wavelet transforms suitable for different applications, not just limited to image coding.

VI. Conclusion

We have shown the possibility of implementing directionlets without forcing independent processing of segments by separating adaptive polyphase decomposition and invariant lifting transform, which we called as in-phase lifting. Particularly, we have shown that using the generating matrices representing unique down sampling pattern, multilevel directionelets can be constructed using in-phase lifting, yet not forcing independent processing of segments. The in-phase lifting implementation may in general be useful to construct adaptive transforms with spatially varying re-sampling patterns.

In image coding, we have shown that in-phase lifting implementation of directionlets successfully eliminates blocking artifacts and improves coding performance by exploiting the correlation across the segment boundaries. Appropriately using both the in-phase lifting implementation of directionlets and the adaptive directional lifting, we have constructed a new transform which has shown improved coding performance over these directional transforms.

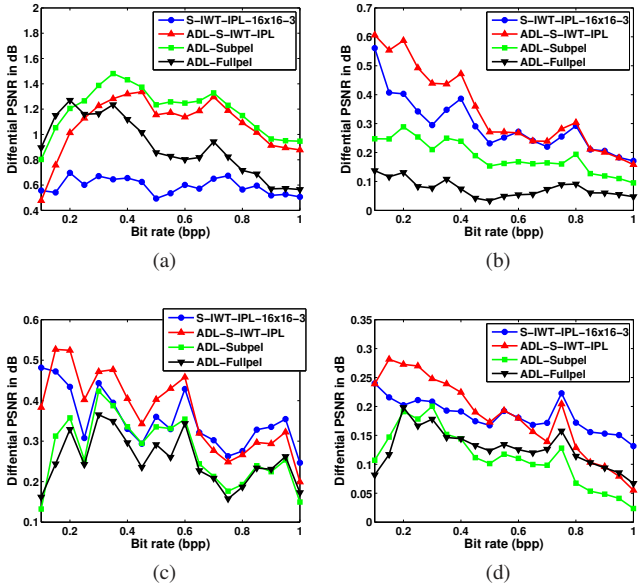


Fig. 11: Difference in PSNR results with respect to S-IWT-64×64 for test images (a) *Barbara*, (b) *Lena*, (c) *Monarch (Butterfly)* , and (d) *Pentagon*. All the transforms are applied to 5 levels and coded using SPIHT.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] E. J. Cand'es and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," in Curve and Surface Fitting, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Saint-Malo: Vanderbilt University Press, 1999.

[2] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091-2106, Dec. 2005.

[3] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 882-893, April 1992.

[4] T. T. Nguyen and S. Oraintara, "A class of multiresolution directional filter bank," *IEEE Trans. Signal Process.*, vol. 55, no. 3, pp. 949-961, Mar. 2007.

[5] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images, *IEEE Trans. Image Process.*, vol. 3, no. 4, pp. 421-437, Jul. 1994.

[6] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic multi-directional representation with separable filtering," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1916-1933, Jul. 2006.

[7] V. Velisavljevic, B. Beferull-Lozano, and M. Vetterli, "Space-frequency quantization for image compression with directionlets," *IEEE Trans. Signal Process.*, vol. 16, no. 7, pp. 1761-1773, Jul. 2007.

[8] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 416-427, Feb. 2007.

[9] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for image compression," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1289-1302, May 2007.

[10] D. Wang, L. Zhang, A. Vincent, and F. Speranza, "Curved wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2413-2421, Aug. 2006.

[11] R. Sethunadh and T. Thomas, "SAR image despeckling in directionlet domain based on edge detection," *IET Electronics Letters*, vol. 49, no.6, Mar. 2013.

[12] W. He, J. Duan, J. Liu and J. Wang, "A Novel Watermarking Scheme using Directionlet," *5th International Congress on Image and Signal Processing*, pp. 557-561, Oct. 2012.

[13] S. Ananda, R. Shantha Selva Kumaria, S. Jeevab, and T. Thivyac, "Directionlet transform based sharpening and enhancement of mammographic X-ray images," *Biomedical Signal Processing and Control*, vol. 8, no. 9, pp. 391-399, Jul. 2013.

[14] X. Zhou, X. Yin, R. Liu and W. Wang, "Infrared and visible image fusion technology based on directionlets transform," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1-4, Feb. 2013.

[15] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245-267, 1998.

[16] D. Jayachandra and A. Makur, "Boundary Handling Mechanism for Lifting Based Spatial Adaptation of Filter Banks," *Proc. SPIE 8295, 82951G (2012)*.

[17] J. Xu and F. Wu, "Subsampling-Adaptive Directional Wavelet Transform for Image Coding," *Data Compression Conference*, 2010.

[18] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, pp. 36-58, Sep. 2001.

[19] P. P. Vaidyanathan, "Multirate Systems and Filter Banks." Englewood Cliffs, NJ: Prentice-Hall, 1993.

[20] Z. Lei, A. Makur, "Enumeration of Downsampling Lattices in Two-Dimensional Multirate Systems," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 414-418, Jan. 2008.

[21] A. Said and W.A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits & Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

[22] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst.Video Technology*, vol. 13, no. 7, pp. 560-576, Jul. 2003.

D. Jayachandra received the B.Tech degree in electronics and communications engineering from NIT, Warangal, India in 2003. From 2003 to 2008, he was with Emuzed India Pvt. Ltd. (now part of Aricent group), Bangalore, India. He is currently pursuing Ph.D degree in electrical and electronic engineering from Nanyang Technological University, Singapore. His current research interests include image/video processing and compression, 1-D and 2-D wavelets and filter banks for image processing, biomedical image segmentation and registration.

Anamitra Makur received the B.Tech. degree from Indian Institute of Technology, Kharagpur, and the M.S. and Ph.D. degrees from California Institute of Technology, Pasadena. He worked in Indian Institute of Science, Bangalore, until 2002 in various capacities, the most recent being a Professor. He also held visiting positions at University of California at Santa Barbara, University of Kaiserslautern, Griffith University at Brisbane, and Keio University at Yokohama. Since 2002, he is an Associate Professor at Nanyang Technological University, Singapore. His current research interests include compressed sensing, signal/image/video compression, multirate signal processing, and image processing. Dr. Makur is the recipient of several awards, notable being the 1998 Young Engineer award from the Indian National Academy of Engineering, and the best paper award in the IEEE APCCAS 2006 conference. He was also the co-author in the student paper award of the IEEE ICASSP 2006 conference. He has served as the Associate Editor of the IEEE Transactions on Signal Processing during 2001-2006.