Coset Sum: an alternative to the tensor product in wavelet construction

Youngmi Hur and Fang Zheng

Abstract—A multivariate biorthogonal wavelet system can be obtained from a pair of multivariate biorthogonal refinement masks in Multiresolution Analysis setup. Some multivariate refinement masks may be decomposed into lower dimensional refinement masks. Tensor product is a popular way to construct a decomposable multivariate refinement mask from lower dimensional refinement masks.

We present an alternative method, which we call coset sum, for constructing multivariate refinement masks from univariate refinement masks. The coset sum shares many essential features of the tensor product that make it attractive in practice: (1) it preserves the biorthogonality of univariate refinement masks, (2) it preserves the accuracy number of the univariate refinement mask, and (3) the wavelet system associated with it has fast algorithms for computing and inverting the wavelet coefficients. The coset sum can even provide a wavelet system with faster algorithms in certain cases than the tensor product. These features of the coset sum suggest that it is worthwhile to develop and practice alternative methods to the tensor product for constructing multivariate wavelet systems. Some experimental results using 2-D images are presented to illustrate our findings.

Index Terms—Coset sum, fast algorithm, interpolatory mask, refinement mask, tensor product, wavelet mask, wavelet system.

I. INTRODUCTION

One of the most common tools for constructing wavelets is Multiresolution Analysis (MRA) [1]. In MRA, a multivariate biorthogonal wavelet system can be obtained from a pair of multivariate biorthogonal refinement masks. The tensor product has been the prevailing method for deriving a pair of multivariate biorthogonal refinement masks from a pair of biorthogonal univariate refinement masks.

In this paper we are interested in studying the operators that map lower dimensional refinement masks to higher dimensional refinement masks. Throughout this paper, the multidimensional (multi-D) refinement masks that can be decomposed into lower dimensional refinement masks by such operators are referred to as *decomposable*. One such operator is the tensor product. The multi-D refinement masks obtained via tensor

The authors are with the Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218-2682 USA (e-mail: hur@jhu.edu; fzheng2@jhu.edu). product are called *tensor product (or separable) refinement* masks. Since the word "separable" is reserved for the tensor product by the definition in the literature, we use the word "decomposable" to indicate more general case than the tensor product. It should be noted that a "nonseparable" refinement mask only means it is not a tensor product refinement mask, and it can still be a "decomposable" refinement mask. Tensor product can also be used to construct multi-D wavelet masks, which are called *tensor product (or separable) wavelet masks* (cf. §II-B).

In MRA setup, construction of multi-D biorthogonal wavelet systems can be done by two steps: (i) construction of multi-D biorthogonal refinement masks (or refinable functions); (ii) construction of multi-D wavelet masks. To construct a nonseparable multi-D wavelet system, one can try making the refinement masks nonseparable in step (i) or making wavelet masks nonseparable in step (ii). Since, once a pair of multivariate biorthogonal refinement masks are given, the matrix extension problem of finding wavelet masks can always be solved by using Quillen-Suslin theorem (see, for example, [2]), the main effort so far for constructing nonseparable wavelets has been made in step (i). However, we note that Quillen-Suslin theorem serves only as a guide since in the process of determining the wavelet masks, some parameters still need to be specified.

Although there have been many methods for constructing nonseparable multi-D wavelets [3]–[16], constructing nonseparable multi-D wavelet systems is highly nontrivial. Many of these methods work only for low spatial dimensions (2-D or 3-D) and they cannot be easily extended to other dimensions. Others assume that the wavelets or refinable functions have a special form (e.g. the refinable function has a box spline factor) and cannot be easily generalized to other cases.

One of the disadvantages of the above approaches for constructing nonseparable wavelets is that they construct a pair of multi-D biorthogonal refinement masks essentially from scratch, which can be quite complicated, especially for high spatial dimensions. A simpler way to obtain multi-D biorthogonal refinement masks is to use an operator that maps 1-D biorthogonal refinement masks. Most of the existing non-separable wavelet construction methods (e.g. [17]–[23]) that use decomposable refinement masks employ operators such as the McClellan transform for quincunx or other 2-channel sampling lattices.

Most multi-D wavelet systems that are used in practice nowadays are separable wavelet systems constructed by the

Copyright (c) 2012 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This work was supported in part by the National Science Foundation under Grant DMS-111590.

This paper was presented in part at the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, November 2011, at the Joint Mathematics Meetings, Boston, MA, January 2012, at the SIAM Conference on Applied Linear Algebra, Valencia, Spain, June 2012, and at the International Conference on Wavelets and Applications, St. Petersburg, Russia, July 2012.

	TABLE I
COMPARISON BETWEEN TENSOR PRO	dduct and coset sum (spatial dimension $n \geq 2$)
$(R, \tilde{R}; univariate refinement masks)$	Coset sum C_n (<i>R</i> , \tilde{R} : univariate refinement masks:

Tensor product \mathcal{T}_n (<i>R</i> , \tilde{R} : univariate refinement masks)	Coset sum C_n (R , \tilde{R} : univariate refinement masks; \tilde{R} : <i>interpolatory</i>)
$\mathcal{T}_n[R]$ can be decomposed into the <i>product</i> of R	$\mathcal{C}_n[R]$ can be decomposed into the sum of R
$\mathcal{T}_n[R]$ is interpolatory iff R is interpolatory	$\mathcal{C}_n[R]$ is interpolatory iff R is interpolatory
$\mathcal{T}_n[R]$ and $\mathcal{T}_n[\tilde{R}]$ are biorthogonal iff R and \tilde{R} are biorthogonal	$\mathcal{C}_n[R]$ and $\mathcal{C}_n[\tilde{R}]$ are biorthogonal iff R and \tilde{R} are biorthogonal
$\mathcal{T}_n[R]$ and R have the same accuracy number	$\mathcal{C}_n[ilde{R}]$ and $ ilde{R}$ have the same accuracy number
$\mathcal{T}_n[R]$ can be decomposed into <i>non-univariate</i> refinement masks	$\mathcal{C}_n[R]$ can be decomposed <i>only</i> into <i>univariate</i> refinement masks
Complexity constant in associated wavelet algorithm increases with n	Complexity constant in associated wavelet algorithm is independent of n

tensor product of 1-D wavelet systems. In §II-B we briefly discuss the use of tensor product in constructing biorthogonal wavelet systems. As we can see from there, the tensor product construction of wavelet systems is extremely simple. This is one of the major reasons the tensor product has been so popular in constructing multi-D wavelets in practice. However the separable wavelet systems have limitations: (i) they have a strong directional bias along lines parallel to the coordinate directions, (ii) they are not very local¹.

Our goal in this paper is to present an alternative method to the tensor product for constructing decomposable multi-D refinement masks. We call the new method as *coset sum*. We show that, under an appropriate circumstance, the coset sum shares many attractive features of the tensor product. First, it preserves the biorthogonality of univariate refinement masks. Second, it preserves the accuracy number of the univariate refinement mask. Third, it has a corresponding wavelet system which has fast algorithms for computing and inverting the wavelet coefficients. In fact, it turns out that these algorithms are faster, in certain cases, than the known algorithms based on tensor product wavelet systems.

Let us elaborate on the last point in more detail. Suppose that we consider two wavelet systems that are constructed from the same pair of 1-D biorthogonal refinement masks, by using tensor product and coset sum. For the tensor product wavelet system, the associated algorithm has complexity $(\alpha + \beta)nN$ (cf. §II-B), where α and β are the number of nonzero coefficients of the 1-D lowpass filters for decomposition and reconstruction, respectively, n is the spatial dimension, and N is the size of an initial data to be analyzed. Thus, the constant in the complexity bound (cf. Complexity discussion in §IV-B for the definition) in this case is $(\alpha + \beta)n$ and it grows linearly with the spatial dimension. On the other hand, as we can see from §IV-B, the complexity constant of the algorithm associated with the coset sum wavelet system we construct in this paper has complexity constant $\frac{3}{2}\alpha + 2\beta$, which is smaller than $(\alpha + \beta)n$ as long as $n \ge 2$. We note that the complexity constant for the coset sum case does not increase even if the spatial dimension increases. For more details, we refer to §IV-B.

The main difference between the coset sum method and

the tensor product method is that a "sum" is used in obtaining the coset sum multi-D refinement masks instead of a "product" used in the tensor product refinement masks. Another difference is that, on the contrary to the tensor product case, the coset sum refinement mask cannot be decomposed into non-univariate refinement masks. Table I summarizes the comparison between the tensor product and the coset sum.

Some experimental results using 2-D images are included to show the potential usefulness of the coset sum wavelet systems we construct in this paper (cf. §IV-C). They show that our wavelet systems can be potentially useful for effectively approximating a certain class of images with strong directional content. They also reveal some of the limitations of our wavelet systems, which include the lack of rotational symmetry [26]. For details, we refer to §IV-C.

The rest of the paper is organized as follows. In §II we briefly overview some relevant concepts on wavelet construction. In §III we introduce the coset sum method and discuss its properties. In §IV we also introduce a particular class of coset sum wavelet systems, together with the associated fast algorithms and some experimental results using our wavelet systems. We summarize our results and present some observations in §V. Appendix contains technical details including all the proofs of the theorems in this paper.

II. PRELIMINARIES

In this section we review some relevant concepts.

A. Refinement masks and wavelet masks

In this paper we refer to a Laurent trigonometric polynomial as a *mask*, and a mask τ with $\tau(0) = 1$ as a *refinement mask*. Refinement masks can be used to obtain refinable functions (see, for example, [27]), which can in turn be used to construct wavelet systems [1].

Refinement masks τ and τ^{d} are *biorthogonal* if they satisfy the following biorthogonal relation:

$$\sum_{\gamma \in \pi\Gamma} (\overline{\tau}\tau^{\mathrm{d}})(\omega + \gamma) = 1, \quad \forall \omega \in \mathbb{T}^n := [-\pi, \pi]^n, \quad (1)$$

where $\Gamma := \{0,1\}^n$ and the overline is used to denote the complex conjugate. In this case, we refer to τ and τ^d as *primal* and dual refinement masks, respectively.

¹One way to measure the localness of a wavelet system is to compute the sum of the volumes of the supports of its mother wavelets (cf. [24], [25]).

1

A refinement mask τ is *interpolatory* if the condition

$$\sum_{\gamma\in\pi\Gamma}\tau(\omega+\gamma)=$$

holds. Thus refinement masks τ and τ^{d} are biorthogonal if and only if $\overline{\tau}\tau^{d}$ is interpolatory. Interpolatory masks are widely used in subdivision schemes and wavelet constructions (for example, see [28] and references therein).

In this paper we say that a filter $h : \mathbb{Z}^n \to \mathbb{R}$ is associated with a mask τ if h and τ are connected via the relation $\tau(\omega) = \frac{1}{2^n} \sum_{k \in \mathbb{Z}^n} h(k) e^{-ik \cdot \omega}$ for $\omega \in \mathbb{T}^n$.

It is straightforward to see that τ is interpolatory if and only if the associated filter h satisfies

$$h(k) = \begin{cases} 1, & \text{if } k = 0, \\ 0, & \text{if } k \in 2\mathbb{Z}^n \backslash 0, \end{cases}$$
(2)

to which we refer as the interpolatory condition for the filter.

For a refinement mask τ , the number of zeros of τ at $\gamma \in \pi\Gamma'$ with $\Gamma' := \Gamma \setminus 0 = \{0,1\}^n \setminus 0$ is referred to as the *accuracy number* [29]. Throughout the paper we assume that all refinement masks have at least accuracy number one, since almost all of the refinement masks used in practice satisfy this condition.

We recall that the Laurent polynomials $\{t_j, t_j^d : j = 1, \dots, l\}$ are called the *wavelet masks* associated with a pair of biorthogonal refinement masks (τ, τ^d) if they satisfy the Mixed Unitary Extension Principle (MUEP) conditions [30]: for every $\omega \in \mathbb{T}^n$,

$$\overline{\tau(\omega+\gamma)}\tau^{\mathrm{d}}(\omega) + \sum_{j=1}^{l} \overline{t_j(\omega+\gamma)}t_j^{\mathrm{d}}(\omega) = \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi\Gamma'. \end{cases}$$
(3)

We refer to t_j , $j = 1, \dots, l$, and t_j^d , $j = 1, \dots, l$, as primal and dual wavelet masks, respectively. When $l = 2^n - 1$, the masks that satisfy the MUEP conditions can be used to construct biorthogonal wavelet systems. We refer to such $(\tau, (t_j)_{j=1,\dots,2^n-1})$ and $(\tau^d, (t_j^d)_{j=1,\dots,2^n-1})$ as the combined biorthogonal masks. A (MRA-based) biorthogonal wavelet system is then obtained from these combined biorthogonal masks, under some simple additional conditions [31], [32].

For a wavelet mask t, the number of zeros of t at $\omega = 0$ is referred to as the *number of (discrete) vanishing moments* [33]. It is well known (see, for example, [33]) that for the combined biorthogonal masks $(\tau, (t_j)_{j=1,\dots,2^n-1})$ and $(\tau^d, (t_j^d)_{j=1,\dots,2^n-1})$ whose refinement masks have at least maccuracy, every primal wavelet mask t_j and dual wavelet mask t_j^d , $j = 1, \dots, 2^n - 1$, has at least m vanishing moments. The number of vanishing moments is closely related to the approximation performance of the wavelet system [34].

B. Tensor product wavelet construction

We recall that the *n*-D tensor product (or separable) refinement mask from *n* (possibly distinct) univariate refinement masks R_1, R_2, \dots, R_n can be written as, for $\omega = (\omega_1, \omega_2, \dots, \omega_n) \in \mathbb{T}^n$,

$$\mathcal{T}_n[R_1, R_2, \cdots, R_n](\omega) := R_1(\omega_1)R_2(\omega_2)\cdots R_n(\omega_n).$$
(4)

When $R = R_1 = R_2 = \cdots = R_n$, we also use the notation $\mathcal{T}_n[R]$. If we let H and h be the filters associated with the masks R and $\mathcal{T}_n[R]$ respectively, they satisfy, for $k = (k_1, k_2, \cdots, k_n) \in \mathbb{Z}^n$,

$$h(k) = H(k_1)H(k_2)\cdots H(k_n).$$

It is well known that the *n*-D refinement masks constructed using tensor product preserve many useful properties of univariate refinement masks. For example, if we let R and \tilde{R} be univariate refinement masks, then

(i) \$\mathcal{T}_n[R]\$ is interpolatory if and only if \$R\$ is interpolatory,
(ii) \$\mathcal{T}_n[R]\$ and \$\mathcal{T}_n[\tilde{R}]\$ are biorthogonal if and only if \$R\$ and \$\tilde{R}\$ are biorthogonal,

(iii) $\mathcal{T}_n[R]$ and R have the same accuracy number.

Now we pose the following question. Can we find another method that satisfies all of the above properties? An affirmative answer is provided by the coset sum, which we introduce and study in the next section. Before introducing the coset sum, let us review the usual approach for constructing biorthogonal wavelet systems.

Construction of 1-D biorthogonal wavelet systems is well understood. Given a pair of 1-D biorthogonal refinement masks S_0 and U_0 , one sets the wavelet masks as

$$S_1(\omega) := e^{-i\omega} \overline{U_0(\omega + \pi)}, \quad U_1(\omega) := e^{-i\omega} \overline{S_0(\omega + \pi)}$$
(5)

for $\omega \in \mathbb{T}$. Then the univariate pairs (S_0, S_1) and (U_0, U_1) satisfy the MUEP conditions (cf. (3)) [31].

On the other hand, given a pair of multivariate biorthogonal refinement masks, constructing a multivariate biorthogonal wavelet system is not so trivial since one needs to find $2^n - 1$ primal wavelet masks t_j 's and $2^n - 1$ dual wavelet masks t_j^d 's.

The usual construction of multi-D biorthogonal wavelet systems is done by the tensor product. Given a pair of 1-D biorthogonal refinement masks S_0 and U_0 , one sets the *n*-D refinement masks as

$$\tau := \mathcal{T}_n[S_0], \quad \tau^{\mathbf{d}} := \mathcal{T}_n[U_0]$$

and the n-D wavelet masks as

$$t_{\nu} = \mathcal{T}_n[S_{\nu_1}, S_{\nu_2}, \cdots, S_{\nu_n}], \quad t_{\nu}^{d} = \mathcal{T}_n[U_{\nu_1}, U_{\nu_2}, \cdots, U_{\nu_n}]$$

for all $\nu = (\nu_1, \nu_2, \dots, \nu_n) \in \Gamma'$. Then the two refinement masks τ and τ^d are also biorthogonal, and $(\tau, (t_{\nu})_{\nu \in \Gamma'})$ and $(\tau^d, (t_{\nu}^d)_{\nu \in \Gamma'})$ satisfy the MUEP conditions (cf. (3)). Here $\Gamma' = \{0, 1\}^n \setminus 0$ is used as before, and the univariate masks S_1 and U_1 are the ones defined in (5). The biorthogonal wavelet systems obtained from these masks are called *tensor product* (or separable) wavelet systems.

It is well known that tensor product wavelet systems have fast algorithms for computing and inverting wavelet coefficients (see, for example, [35]), to which we refer as the *fast tensor product wavelet algorithms*. These algorithms have linear complexity O(N), where N is the size of the input data. More precisely, if α is the number of nonzero entries of the filter associated with S_0 and β is the number of nonzero entries of the filter associated with U_0 , then the algorithms for computing and inverting the corresponding tensor product wavelet coefficients have complexity $(\alpha + \beta)nN$, where n

III. COSET SUM

A. Introduction to coset sum

We present an alternative method, called *coset sum*, to the tensor product in wavelet construction. Instead of the "product" in the tensor product, we propose to use a "sum" to construct multivariate refinement masks from univariate refinement masks.

Let R be a univariate refinement mask and let H be the univariate filter associated with R. For $\nu \in \Gamma'$, the map

$$\mathbb{T}^n \to \mathbb{C} : \omega \mapsto \frac{1}{2^{n-1}} R(\omega \cdot \nu),$$

where $\omega \cdot \nu$ is the inner product in \mathbb{R}^n , is an *n*-D Laurent trigonometric polynomial. The normalization factor $\frac{1}{2^{n-1}}$ is used to place $R(\omega \cdot \nu)$ in the *n*-D space. In terms of filters, the above can be understood as aligning the 1-D filter *H* along the ν direction:

$$\mathbb{Z}^n \to \mathbb{R}: k \mapsto \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases}$$

Since we want to consider all the directions in Γ' , a possible candidate for the coset sum definition can be given as

$$\mathbb{T}^n \to \mathbb{C} : \omega \mapsto A + \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} R(\omega \cdot \nu).$$

Since we want the coset sum to map a 1-D *refinement* mask to an *n*-D *refinement* mask, by plugging in $\omega = 0$, we obtain $A = -1 + \frac{1}{2^{n-1}}$ and get to the following definition.

Definition 1: We define the coset sum C_n that maps a 1-D refinement mask R to an n-D refinement mask $C_n[R]$ as follows: for $\omega \in \mathbb{T}^n$

$$\mathcal{C}_n[R](\omega) := \frac{1}{2^{n-1}} \left(1 - 2^{n-1} + \sum_{\nu \in \Gamma'} R(\omega \cdot \nu) \right),$$

where $\Gamma' = \Gamma \backslash 0 = \{0, 1\}^n \backslash 0$.

Remark 1. We call the refinement mask obtained by the coset sum method as the coset sum refinement mask. The set $\Gamma = \{0,1\}^n$ used in the definition is a complete set of representatives of the distinct cosets (hence the name "coset sum") of the quotient group $\mathbb{Z}^n/2\mathbb{Z}^n$. It is easy to observe that, because *n*-D masks are 2π -periodic, the set $\{0,1\}^n$ used in this paper prior to the above definition (for example, for biorthogonality condition, interpolatory condition, definition of accuracy number, and MUEP conditions) can be replaced, without changing the meaning of the statements, by any other complete set of representatives of the distinct cosets of the quotient group $\mathbb{Z}^n/2\mathbb{Z}^n$ as long as the set contains 0. As a result, the set $\{0,1\}^n$ used in the above coset sum definition can be replaced by any such an alternative set. The set $\{0,1\}^n$ is chosen for the discussion in this paper (with the exception of Example 3 below and discussions in §IV-C) because it makes the support of the associated filter the smallest. Depending on applications, choosing a different set of representatives can

make more sense. We emphasize that even if all the results in our paper (including Theorem 1, 2, and the fast coset sum wavelet algorithms in later part of the paper) are presented using the set $\Gamma = \{0,1\}^n$, they will stay intact for other choices for Γ .

Remark 2. We recall that the sum in the left-hand side of the biorthogonality condition in (1) is taken over the set $\pi\Gamma$, which can be considered as a set of coset representatives of $2\pi(\frac{1}{2}\mathbb{Z}^n/\mathbb{Z}^n)$. The set of coset representatives has been previously used in the wavelet literature, mostly in relation with this biorthogonality condition. For example, a new algorithm called a coset by coset (CBC) is proposed in [36] for obtaining dual masks with arbitrary number of accuracy given an interpolatory primal mask, and the coset representatives are used in [37] for an explicit, flexible, and easy implementation of interpolatory subdivision schemes.

The coset sum for the first few low dimensions are given as follows:

$$\mathcal{C}_1[R](\omega_1) = R(\omega_1),$$

$$C_2[R](\omega_1, \omega_2) = \frac{1}{2} \{-1 + R(\omega_1) + R(\omega_2) + R(\omega_1 + \omega_2)\},\$$

$$\mathcal{C}_{3}[R](\omega_{1},\omega_{2},\omega_{3}) = \frac{1}{4} \{-3 + R(\omega_{1}) + R(\omega_{2}) + R(\omega_{1} + \omega_{2}) + R(\omega_{3}) + R(\omega_{1} + \omega_{3}) + R(\omega_{2} + \omega_{3}) + R(\omega_{1} + \omega_{2} + \omega_{3}) \}.$$

We note that the coset sum formula in the above definition can also be written as

$$-1 + \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma} R(\omega \cdot \nu) \tag{6}$$

or

$$\frac{1}{2^{n-1}} \left(\frac{1}{2} + \sum_{\nu \in \Gamma'} \left(R(\omega \cdot \nu) - \frac{1}{2} \right) \right). \tag{7}$$

The filter h associated with the coset sum refinement mask $C_n[R]$ is connected to the univariate filter H via

$$h(k) = \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z} \backslash 0, \nu \in \Gamma', \\ 2^n - (2^n - 1)(2 - H(0)), & \text{if } k = 0, \\ 0, & \text{for all other } k \in \mathbb{Z}^n. \end{cases}$$
(8)

If the univariate filter H associated with R is interpolatory, the *n*-D filter *h* associated with $C_n[R]$ is also interpolatory and it can be expressed as

$$h(k) = \begin{cases} H(K), & \text{if } k = K\nu \text{ for some } K \in \mathbb{Z}, \nu \in \Gamma', \\ 0, & \text{ for all other } k \in \mathbb{Z}^n. \end{cases}$$

In particular, the restriction of the *n*-D filter *h* to ν direction, for each $\nu \in \Gamma'$, is the 1-D filter *H*.

Now we give a few very simple examples of constructing multi-D refinement filters from univariate refinement filters.

Example 1: *n***-D Haar refinement filter: the only filter that can be obtained using either the tensor product or the coset sum.** Consider the 2-D Haar refinement filter

$$h(k) = \begin{cases} 1, & \text{if } k = (0,0), (1,0), (0,1) \text{ or } (1,1), \\ 0, & \text{otherwise.} \end{cases}$$



Fig. 1. Constructions of 2-D Haar refinement filter (Tensor product and Coset sum) (cf. Example 1)

Let H be the 1-D Haar refinement filter

$$H(K) = \begin{cases} 1, & \text{if } K = 0 \text{ or } K = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then h can be obtained from H either by

- (I) (Tensor Product Case) aligning the filter H along y = 0line (x-axis) and y = 1 line (see Figure² 1(a)), or by
- (II) (Coset Sum Case) aligning the filter H along y = 0 line (x-axis), x = 0 line (y-axis), and y = x line (see Figure 1(b)).

Since the support of the 2-D tensor product refinement filter will always be a rectangle and the support of the 2-D coset sum refinement filter will always be the union of three line segments in different directions, it is easy to see that, up to the integer translation, the 2-D Haar refinement filter is the only 2-D filter that can be obtained using either the tensor product or the coset sum. It is straightforward to show that, for arbitrary spatial dimension n, the n-D Haar refinement filter is the only filter that can be obtained using either the tensor product or the coset sum.

Example 2: Refinement filter associated with an n**-D piecewise-linear box spline.** Let us consider the 2-D refinement filter h associated with a 2-D piecewise-linear box spline [38]:

$$h(k) = \begin{cases} 1, & \text{if } k = (0,0), \\ \frac{1}{2}, & \text{if } k = \pm (1,0), \pm (0,1), \text{ or } \pm (1,1), \\ 0, & \text{otherwise.} \end{cases}$$

Let H be the refinement filter associated with a 1-D piecewiselinear spline:

$$H(K) = \begin{cases} 1, & \text{if } K = 0, \\ \frac{1}{2}, & \text{if } K = \pm 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then *h* can be obtained from *H* by aligning the filter *H* along y = 0 line (*x*-axis), x = 0 line (*y*-axis), and y = x line (see Figure 2). In other words, $h = C_2[H]$. In fact, it is easy to see that for the *n*-D refinement filter *h* associated with an *n*-D piecewise-linear box spline, we have $h = C_n[H]$.

Example 3: Refinement filter supported on different directed line segments. We consider n = 2 and choose the same univariate filter H as in Example 2, but choose the 2-D filter h differently:

$$h(k) = \begin{cases} 1, & \text{if } k = (0,0), \\ \frac{1}{2}, & \text{if } k = \pm (1,2), \ \pm (2,1), \text{ or } \pm (-1,1), \\ 0, & \text{otherwise.} \end{cases}$$

 2 In the figures of filters drawn in this paper, the bold-faced number is used to represent the value of the filter at the origin.



Fig. 2. Construction of 2-D piecewise-linear box spline refinement filter (Coset sum) (cf. Example 2)

				0	0	0	$\frac{1}{2}$	0
				0	$\frac{1}{2}$	0	0	$\frac{1}{2}$
$\frac{1}{2}$	1	$\frac{1}{2}$	\longrightarrow	0	0	11	0	0
2		2		$\frac{1}{2}$	0	0	$\frac{1}{2}$	0
				0	$\frac{1}{2}$	0	0	0

Fig. 3. 2-D coset sum refinement filter supported on different directed line segments (cf. Example 3)

Then $h = C_2[H]$ with Γ chosen differently (cf. Remark 1 after Definition 1):

$$\Gamma = \{(0,0), (2,1), (1,2), (-1,1)\}.$$

In particular, h can be obtained from H by aligning the filter H along y = x/2 line, y = 2x line, and y = -x line (see Figure 3). Note that the filter h is supported on the line segments that are not parallel to the coordinate directions.

B. Properties of coset sum refinement masks

In this subsection, we study the properties of the multi-D refinement masks obtained by the coset sum method.

The following theorem shows that the refinement masks obtained by the coset sum share many important properties with the tensor product refinement masks.

Theorem 1: Let C_n be the coset sum, and let R and \hat{R} be univariate refinement masks.

- (a) $C_n[R]$ is interpolatory if and only if R is interpolatory.
- (b) Suppose that one of R and R is interpolatory. Then C_n[R] and C_n[R̃] are biorthogonal if and only if R and R̃ are biorthogonal.
- (c) Suppose that R is interpolatory. Then $C_n[R]$ and R have the same accuracy number.

Proof: See Appendix A.

Below we add a few remarks on Theorem 1.

Remark on Theorem 1(b). The interpolatory condition in part (b) cannot be omitted. To see this, we consider the univariate refinement mask associated with Daubechies wavelet system of order 2 [39], and let

$$R(\omega) = \tilde{R}(\omega) = \cos^2(\frac{\omega}{2}) \left(\frac{1+\sqrt{3}}{2} + \frac{1-\sqrt{3}}{2}e^{-i\omega}\right), \omega \in \mathbb{T}.$$

Then R (hence \tilde{R}) is not interpolatory, and R and \tilde{R} are biorthogonal. However it is easy to see that $C_2[R]$ and $C_2[\tilde{R}]$ are not biorthogonal.

Remark on Theorem 1(c). For general (not necessarily interpolatory) R, the accuracy number of $C_n[R]$ is at least $\min\{m_1, m_2\}$ where m_1 is the accuracy number of R and



Filter associated with $C_2[U_4]$

Fig. 4. Refinement filters associated with the masks U_4 and $\mathcal{C}_2[U_4]$ in Example 4

 m_2 is the order that 1 - R has a zero at the origin. This statement can be proved using similar arguments as in the proof of Theorem 1(c), and we omit the proof.

The Deslauriers-Dubuc mask [40] of order $2k \ (k \in \mathbb{N})$ is defined as

$$U_{2k}(\omega) := \cos^{2k}(\frac{\omega}{2}) P_k(\sin^2(\frac{\omega}{2})),$$
(9)
$$P_k(x) := \sum_{j=0}^{k-1} \frac{(k-1+j)!}{j!(k-1)!} x^j.$$

The mask U_{2k} is interpolatory and has accuracy number 2k. We now present a family of biorthogonal coset sum refinement masks based on the Deslauriers-Dubuc interpolatory masks.

Example 4: A family of *n*-D biorthogonal coset sum refinement masks. For each $k \in \mathbb{N}$, we choose U_{2k} in (9) as a univariate interpolatory refinement mask. By Theorem 1(a)(c), $C_n[U_{2k}]$ is an *n*-D interpolatory refinement mask with accuracy number 2k. It is straightforward to see that for each $k \in \mathbb{N}$,

$$S_{2k} := U_{2k}(3 - 2U_{2k}) \tag{10}$$

is biorthogonal³ to U_{2k} . By Theorem 1(b), $C_n[U_{2k}]$ is biorthogonal to $C_n[S_{2k}]$. Since S_{2k} has at least 2k accuracy and $1-S_{2k}$ has a zero of order at least 2k at the origin, by the Remark on Theorem 1(c), $C_n[S_{2k}]$ has at least 2k accuracy. The filters for the case k = n = 2 are depicted in Figure 4 and 5. Using the standard tool in wavelet literature (see, for example, [44] and references therein), one can show that both $C_2[U_4]$ and $C_2[S_4]$ generate the refinable functions that are in $L^2(\mathbb{R}^2)$ (cf. Figure 6).

Similar to the tensor product case, the coset sum can actually take different univariate refinement masks. However, since the cardinality of the set Γ' is $2^n - 1$, we have $2^n - 1$

	- 512	0	512	$-\frac{52}{512}$	$-\frac{120}{512}$	512	512	512	$-\frac{120}{512}$	$-\frac{52}{512}$	512	0	- 512
Filter associated with S_4													
			\mathcal{C}_2 (Coset sum)										
	0	0	0	0	0	0	$-\frac{2}{512}$	0	0	0	0	0	$-\frac{2}{512}$
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	$\frac{36}{512}$	0	0	0	$\frac{36}{512}$	0	0
	0	0	0	0	0	0	$-\frac{32}{512}$	0	0	$-\frac{32}{512}$	0	0	0
	0	0	0	0	0	0	$-\frac{126}{512}$	0	$-\frac{126}{512}$	0	0	0	0
	0	0	0	0	0	0	$\frac{288}{512}$	$\frac{288}{512}$	0	0	0	0	0
	$-\frac{2}{512}$	0	$\frac{36}{512}$	$-\frac{32}{512}$	$-\frac{126}{512}$	$\frac{288}{512}$	$\frac{1064}{512}$	$\frac{288}{512}$	$-\frac{126}{512}$	$-\frac{32}{512}$	$\frac{36}{512}$	0	$-\frac{2}{512}$
	0	0	0	0	0	$\frac{288}{512}$	288 512	0	0	0	0	0	0
	0	0	0	0	$-\frac{126}{512}$	0	$-\frac{126}{512}$	0	0	0	0	0	0
	0	0	0	$-\frac{32}{512}$	0	0	$-\frac{32}{512}$	0	0	0	0	0	0
	0	0	$\frac{36}{512}$	0	0	0	$\frac{36}{512}$	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	$-\frac{2}{512}$	0	0	0	0	0	$-\frac{2}{512}$	0	0	0	0	0	0
				-						~ 1			

Filter associated with $C_2[S_4]$

Fig. 5. Refinement filters associated with the masks S_4 and $\mathcal{C}_2[S_4]$ in Example 4



Fig. 6. The refinable functions associated with the coset sum refinement masks $C_2[U_4]$ (left) and $C_2[S_4]$ (right) in Example 4

different directions to consider, instead of n different coordinate directions for the tensor product case. In such a case the n-D coset sum refinement can be written as

$$\mathcal{C}_{n}[(R_{\nu})_{\nu\in\Gamma'}](\omega) := \frac{1}{2^{n-1}} \left(1 - 2^{n-1} + \sum_{\nu\in\Gamma'} R_{\nu}(\omega\cdot\nu) \right),$$
(11)

where R_{ν} , $\nu \in \Gamma'$, are possibly distinct univariate refinement masks for different direction ν .

Let $n = n_1 + n_2 + \cdots + n_m, n_j \ge 1$ for $j = 1, 2, \cdots, m$. Then the tensor product refinement mask in (4) can be written as the product of possibly non-univariate lower dimensional tensor product refinement masks as follows: for $\omega = (\omega_1, \omega_2, \cdots, \omega_n) \in \mathbb{T}^n$,

$$\begin{aligned} \mathcal{T}_n[R_1,\cdots,R_n](\omega) \\ &= \mathcal{T}_{n_1}[R_1,\cdots,R_{n_1}](\omega_1,\cdots,\omega_{n_1}) \cdot \\ &\quad \mathcal{T}_{n_2}[R_{n_1+1},\cdots,R_{n_1+n_2}](\omega_{n_1+1},\cdots,\omega_{n_1+n_2}) \cdot \\ &\quad \cdots \mathcal{T}_{n_m}[R_{n_1+\cdots+n_{m-1}+1},\cdots,R_n](\omega_{n_1+\cdots+n_{m-1}+1},\cdots,\omega_n). \end{aligned}$$

On the contrary, the coset sum refinement mask *cannot* be written as the sum of non-univariate lower dimensional coset sum refinement masks.

We can also consider a hybrid of the coset sum and the tensor product : for $n = n_1 + n_2 + \cdots + n_m, n_j \ge 1, j =$

³Given a refinement filter, a dual refinement filter is not uniquely determined in general. The specific choice of the dual filter of U_{2k} as in (10) can be obtained, for example, from Proposition 2.1 in [41]. See also Theorem 2 in [42] for an alternative derivation based on a critical representation of the Laplacian pyramid ([43]).



Multivariate refinement masks

Fig. 7. The tensor product multivariate refinement masks are not the only decomposable refinement masks. The coset sum provides a systematic way to construct other types of decomposable refinement masks. The other decomposable refinement masks include the ones constructed by the existing approaches (cf. discussion in §1). The multivariate Haar refinement mask is essentially the only mask that can be obtained by using either the tensor product or the coset sum (cf. Example 1).

$$1, 2, \cdots, m,$$

$$\mathcal{C}_{n_1}[R](\omega_1, \cdots, \omega_{n_1}) \cdot \mathcal{C}_{n_2}[R](\omega_{n_1+1}, \cdots, \omega_{n_1+n_2}) \cdot$$

$$\cdots \mathcal{C}_{n_m}[R](\omega_{n_1+\dots+n_{m-1}+1}, \dots, \omega_n).$$
(12)

Similar statements to the ones of Theorem 1 can be made for the coset sum refinement mask in a generalized sense as in (11) and for the hybrid refinement mask as in (12). We omit the statements and the proofs as they are similar to the ones of Theorem 1.

The diagram in Figure 7 illustrates the relation among the tensor product, the coset sum, and the decomposable multi-D refinement masks. We note that the type of decomposable refinement masks that can be obtained by coset sum is different from the one by the aforementioned existing methods [17]–[23] since coset sum works for 2^n -channel sampling lattices (cf. §IV-B).

IV. APPLICATION: COSET SUM WAVELET SYSTEMS

In this section we introduce a special class of wavelet systems that can be derived from coset sum refinement masks in a very simple manner, and present their properties, including fast algorithms, together with some experimental results.

A. Coset sum wavelet systems

Since the coset sum provides a way to construct a pair of multivariate biorthogonal refinement masks from univariate ones, it can be combined with any procedure for finding wavelet masks to construct a multivariate biorthogonal wavelet system. It is well known (for example, see [45] and Example 5 below) that for a given pair of n-D biorthogonal refinement masks, different biorthogonal wavelet systems can be obtained by choosing wavelet masks differently. The specific choice we make in this paper is guided by the simplicity of the form of the primal wavelet masks (cf. (13) and the discussion below). Use of other criteria may result in a totally different type of "coset sum" wavelet systems, hence discussing about properties of coset sum wavelet systems makes sense only after

the wavelet masks are specifically chosen. Below we present our approach for determining the wavelet masks.

Suppose that S and U are 1-D biorthogonal refinement masks, and that U is interpolatory. Theorem 1(b) implies that the n-D coset sum refinement masks $C_n[S]$ and $C_n[U]$ are biorthogonal. Moreover, from (7) and the assumption that U is interpolatory, we see that the restriction of the n-D mask $C_n[U]$ to ν direction, $\nu \in \Gamma' = \{0,1\}^n \setminus 0$, is given by $U(\omega \cdot \nu)$ for $\omega \in \mathbb{T}^n$ (up to constants), which is essentially a 1-D mask. Hence, as in the 1-D wavelet construction (cf. (5)), one can attempt to define the multivariate wavelet masks $t_{\nu}, \nu \in \Gamma'$, (note that we have $2^n - 1$ wavelet masks) of the form

$$t_{\nu}(\omega) = e^{-i\omega \cdot \nu} \overline{U(\omega \cdot \nu + \pi)}, \quad \omega \in \mathbb{T}^{n}.$$
 (13)

The next theorem shows that the above approach leads to the construction of n-D biorthogonal wavelet systems.

Theorem 2: Suppose that S and U are 1-D biorthogonal refinement masks, and that U is interpolatory. Define n-D biorthogonal refinement masks as

$$\tau := \mathcal{C}_n[S], \quad \tau^{\mathrm{d}} := \mathcal{C}_n[U],$$

and *n*-D primal wavelet masks t_{ν} , $\nu \in \Gamma'$, as in (13). Then there exist dual wavelet masks t_{ν}^{d} , $\nu \in \Gamma'$, such that $(\tau, (t_{\nu})_{\nu \in \Gamma'})$ and $(\tau^{d}, (t_{\nu}^{d})_{\nu \in \Gamma'})$ are *n*-D combined biorthogonal masks.

Proof: See Appendix B.

Remark 1. We refer to the biorthogonal wavelet system constructed from the n-D combined biorthogonal masks in Theorem 2 as the *canonical coset sum wavelet system*. As we discussed previously, there may be many other coset sum wavelet systems associated with the same coset sum refinement masks. Throughout this paper, the word "canonical" is suppressed when no confusion arises.

Remark 2. The exact form of the dual wavelet masks t_{ν}^{d} , $\nu \in \Gamma'$, of the canonical coset sum wavelet system in Theorem 2 is not important for understanding our results in this paper, but knowing it may be useful in some other contexts. By carefully inspecting the proof of Theorem 2, we see that the dual wavelet masks t_{ν}^{d} , $\nu \in \Gamma'$, have the form

$$t^{\mathrm{d}}_{\nu}(\omega) = 2^{-n+1} e^{-i\omega \cdot \nu} (1 - 2\tau^{\mathrm{d}}(\omega) \overline{S^{o}(\omega \cdot \nu)}), \ \omega \in \mathbb{T}^{n}, \ (14)$$

where
$$S^o := (S - S(\cdot + \pi))/2$$
 is the odd part of S.

Remark 3. We recall that there is a nonseparable multi-D wavelet construction method based on the traditional lifting scheme ([46]) proposed by J. Kovačević and W. Sweldens [47]. A key ingredient of their construction is a class of n-D filters called Neville filters, which are used to build the predict and update filters. Given suitable n-D Neville filters, their method can construct the associated n-D biorthogonal wavelet systems. It turns out that if the Neville filters are extracted from the n-D biorthogonal coset sum refinement masks, the above canonical coset sum wavelet systems can also be obtained by their method, and our fast algorithms associated with these wavelet systems (cf. §IV-B) can be viewed as realization of a special case of their fast transform. However it should be noted that the Neville filters extracted

from the *n*-D biorthogonal coset sum refinement masks cannot be obtained from [47]. Also note that other coset sum wavelet systems besides the canonical ones cannot be constructed by their method regardless of the choice of the Neville filters. \blacksquare

Canonical coset sum wavelet systems have many potentially useful properties. The most distinctive property is that they can be associated with fast algorithms, which is explained in detail in the next subsection. Another (related) property is that they can be much more local than tensor product wavelet systems. The easiest way to see this property is probably through the following example.

Example 5: *n*-**D** coset sum Haar wavelet systems. The simplest choice for the univariate refinement mask is the 1-D Haar refinement mask $R(\omega) = \frac{1}{2} + \frac{1}{2}e^{-i\omega}$, $\omega \in \mathbb{T}$, which is biorthogonal to itself and interpolatory. Let $\tau = \tau^{d} = C_n[R]$ be the *n*-D Haar refinement mask, which can be obtained either by coset sum or tensor product (cf. Example 1). Then from Theorem 2 and the remarks after it, we obtain the canonical coset sum wavelet system whose *n*-D biorthogonal refinement masks are τ and τ^{d} , and whose primal and dual wavelet masks are, for $\nu \in \Gamma'$ and $\omega \in \mathbb{T}^n$,

$$t_{\nu}(\omega) = \frac{e^{-i\omega \cdot \nu} - 1}{2}, \quad t_{\nu}^{\mathrm{d}}(\omega) = \frac{e^{-i\omega \cdot \nu} - \tau^{\mathrm{d}}(\omega)}{2^{n-1}}$$

We refer to this wavelet system as the n-D (canonical) coset sum Haar wavelet system. Each of the wavelet masks of this wavelet system has one vanishing moment. We note that this wavelet system is the same as the piecewise-constant biorthogonal wavelet system introduced in [24] (up to constants), which is shown to be far more local than the tensor product Haar wavelet system, in high spatial dimensions.

Remark. We recall that orthogonality is a special case of biorthogonality. We note that the n-D Haar refinement mask (cf. Example 1 and 5) is orthogonal, whereas the n-D canonical coset sum Haar wavelet system (cf. Example 5) is not orthogonal. In fact, it is not possible to construct n-D canonical coset sum wavelet system that is orthogonal. This can be seen from the facts that the 1-D refinement mask we start with for such a wavelet system has to be interpolatory and orthogonal, and that there is no 1-D interpolatory orthogonal refinement mask (in the dyadic dilation) other than the Haar one (see, for example, [41]), whose associated n-D canonical coset sum wavelet system is not orthogonal as we just established.

A drawback of the *n*-D coset sum Haar wavelet system in the previous example is that the wavelet masks have only one vanishing moment. In order to construct *n*-D biorthogonal wavelet systems with larger number of vanishing moments, one needs to have *n*-D biorthogonal refinement masks with larger number of accuracy (cf. §II-A). In general, constructing *n*-D biorthogonal refinement masks with large number of accuracy can be cumbersome, especially when *n* is large, since it involves solving a large number of linear equations. Since coset sum can preserve the biorthogonality and the accuracy number simultaneously, it allows one to bypass solving these linear systems to get biorthogonal refinement masks with large number of accuracy. Thus it is often easier to construct *n*-D wavelet systems based on the coset sum than other *n*-D

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	$\frac{1}{16}$	0	$-\frac{9}{16}$	1	$-\frac{9}{16}$	0	$\frac{1}{16}$
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
(a)	Wavel	et filte	er asso	ciated wi	ith th	ne mask	$t_{1,0}(\omega$	(ω_1, ω_2)
0	0	0	0	$\frac{1}{16}$	(0 0	0	0
0	0	0	0	0	(0 0	0	0
0	0	0	0	$-\frac{9}{16}$	(0 0	0	0
0	0	0	0	1	(0 0	0	0
0	0	0	0	$-\frac{9}{16}$	(0 0	0	0
0	0	0	0	0	(0 0	0	0
0	0	0	0	$\frac{1}{16}$	(0 0	0	0
0	0	0	0	0	(0 0	0	0
0	0	0	0	0	(0 0	0	0
(b)	Wavel	et filte	er asso	ciated w	ith tl	ne mask	$t_{0,1}(\omega$	(ω_1,ω_2)
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	16 0
0	0	0	0	0	0	<u> </u>	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	9	0	0	0	0
0	0	0	0	$\frac{16}{0}$	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	16 0	0	0 0	0	Ŭ 0	0	0
0	0	0	0	0 0	0	0 0	0	0
(c)	Wavel	et filte	er asso	ciated wi	ith th	ne mask	τ _{1 1} (μ	(ω_1, ω_2)

Fig. 8. Primal coset sum wavelet filters of Example 6 for 2-D with 4 vanishing moments (n = k = 2)

wavelet systems, for large number of vanishing moments. The next is an example of such coset sum wavelet systems.

Example 6: A family of n**-D coset sum wavelet systems with larger number of vanishing moments.** We choose the univariate refinement masks U_{2k} (interpolatory) and S_{2k} as in (9) and (10), respectively, and apply Theorem 2. Then with the primal wavelet masks given as

$$t_{\nu}(\omega) = e^{-i\omega \cdot \nu} \sin^{2k}(\frac{\omega \cdot \nu}{2}) P_k(\cos^2(\frac{\omega \cdot \nu}{2})), \quad \nu \in \Gamma',$$

with $\Gamma' = \{0,1\}^n \setminus 0$, there exist dual wavelet masks t_{ν}^d , $\nu \in \Gamma'$, such that $(\mathcal{C}_n[S_{2k}], (t_{\nu})_{\nu \in \Gamma'})$ and $(\mathcal{C}_n[U_{2k}], (t_{\nu}^d)_{\nu \in \Gamma'})$ are *n*-D combined biorthogonal masks. It is easy to check that each of the wavelet masks of this wavelet system has 2k vanishing moments. All the primal wavelet filters are supported on the union of $2^n - 1$ line segments along ν direction for $\nu \in \Gamma'$. For example, if n = 2, then $\Gamma' = \{(1,0), (0,1), (1,1)\}$ and





Fig. 9. The magnitude of the primal coset sum masks $C_2[S_4], t_{(1,0)}, t_{(0,1)}$, and $t_{(1,1)}$ in Example 6

Fig. 10. The magnitude of the primal tensor product masks that are comparable to the masks in Figure 9

the primal wavelet masks for the case k = 2 are given as

$$\begin{split} t_{(1,0)}(\omega_1,\omega_2) &= e^{-i\omega_1} \sin^4(\frac{\omega_1}{2}) \left(1 + 2\cos^2(\frac{\omega_1}{2})\right), \\ t_{(0,1)}(\omega_1,\omega_2) &= e^{-i\omega_2} \sin^4(\frac{\omega_2}{2}) \left(1 + 2\cos^2(\frac{\omega_2}{2})\right), \\ t_{(1,1)}(\omega_1,\omega_2) &= e^{-i(\omega_1+\omega_2)} \sin^4(\frac{\omega_1+\omega_2}{2}) \\ &\cdot \left(1 + 2\cos^2(\frac{\omega_1+\omega_2}{2})\right). \end{split}$$

The associated wavelet filters are depicted in Figure 8. The magnitude of the primal masks $C_2[S_4], t_{(1,0)}, t_{(0,1)}$, and $t_{(1,1)}$ (i.e. the magnitude of the frequency responses of the filters associated with the primal masks) are depicted in Figure 9. The magnitude of the corresponding tensor product primal masks are given in Figure 10 for comparison.

Since $C_2[U_4]$ and $C_2[S_4]$ generate the refinable functions that are in $L^2(\mathbb{R}^2)$ (cf. Example 4 and Figure 6), and since we have FIR filters, the 2-D coset sum wavelet system generated from the combined biorthogonal masks $(C_2[S_4], t_{(1,0)}, t_{(0,1)}, t_{(1,1)})$ and $(C_2[U_4], t_{(1,0)}^d, t_{(0,1)}^d, t_{(1,1)}^d)$ is also in $L^2(\mathbb{R}^2)$.

It should be noted that the above properties (the space localization property discussed in Example 5, and the frequency responses, the vanishing moments, and the smoothness-in the sense of whether or not a wavelet system belongs to L^2 discussed in Example 6) of canonical coset sum wavelet systems may not hold true for other coset sum wavelet systems.

B. Fast coset sum wavelet algorithms

Next we show that the canonical coset sum wavelet system can be associated with the fast algorithm with linear complexity whose complexity constant does not grow with the spatial dimension. When presenting and analyzing our algorithm below, we use mostly filters instead of masks that we have been used so far, as this approach will be more useful in practice. Fast Coset Sum Wavelet Algorithms. Let S and U be biorthogonal univariate refinement masks, where U is interpolatory. Let G and H be the filters associated with the refinement masks S and U, respectively. In particular, H is interpolatory (cf. (2)).

$$\begin{array}{l} \text{(1) Decomposition Algorithm:} \\ a_G = -2^n + 2 + (2^n - 1)G(0) \\ \text{for } j = J, J - 1, \cdots, 1 \\ \text{for } k \in \mathbb{Z}^n \\ y_{j-1}(k) \\ = \frac{1}{2^n} (a_G y_j(2k) + \sum_{\nu \in \Gamma'} \sum_{L \in \mathbb{Z} \setminus 0} G(L) y_j(2k + L\nu)) \quad (\texttt{i}) \\ \text{end} \\ \text{for } \nu \in \Gamma' \text{ and } k \in \mathbb{Z}^n \\ w_{\nu,j-1}(k) \\ = \frac{1}{2} (y_j(2k + \nu) - \sum_{m \equiv 1} H(m) y_j(2k + (1 - m)\nu)) \quad (\texttt{ii}) \\ \text{end} \end{array}$$

end

(2) Reconstruction Algorithm:

for
$$j = 1, \dots, J - 1, J$$

for $k \in \mathbb{Z}^n$
 $y_j(2k)$
 $= y_{j-1}(k) - \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} \sum_{L \in \mathbb{Z}} G(2L+1) w_{\nu,j-1}(k+L\nu)$
(iii)

end

for
$$\nu \in \Gamma'$$
 and $k \in \mathbb{Z}^n$
 $y_j(2k + \nu)$
 $= 2w_{\nu,j-1}(k) + \sum_{m \equiv 1} H(m)y_j(2k + (1 - m)\nu)$ (iv)
end

end

Given coarse coefficients y_j at level j, Decomposition Algorithm first computes the lower level coarse coefficients y_{j-1} ,

and then the wavelet coefficients $w_{\nu,j-1}, \nu \in \Gamma' = \{0,1\}^n \setminus 0$. The coefficients y_{j-1} and $w_{\nu,j-1}$ are obtained by filtering (using the n-D filter g associated with the coset sum refinement mask $\mathcal{C}_n[S]$ for y_{j-1} and the *n*-D filter h_{ν} associated with the primal coset sum wavelet mask t_{ν} for $w_{\nu,j-1}$ followed by downsampling, as is typically done in wavelet decomposition process (see, for example, [39]). Since the *n*-D mask $C_n[S]$ can be written in terms of 1-D mask S (cf. Definition 1), the associated n-D filter g can be written in terms of 1-D filter G (cf. (8)). Similarly, from the fact that the n-D mask t_{ν} can be written in terms of 1-D mask U (cf. (13)), the associated *n*-D filter h_{ν} can be written in terms of 1-D filter H. Taking into account these observations, we get the above simple expressions for y_{j-1} and $w_{\nu,j-1}$ in Step (i) and (ii), respectively. In Step (ii) and (iv), $m \equiv 1$ is used to mean that m is congruent to 1 in modulo 2, i.e., m is an odd integer.

Reconstruction Algorithm recovers y_j from y_{j-1} , and $w_{\nu,j-1}$, $\nu \in \Gamma'$. It first recovers y_j at even points (cf. Step (iii)) and then at all other points (cf. Step (iv)). Step (iii) is a key step in making our algorithm fast (cf. Complexity discussion below). It is easy to show that the identity in Step (iii) holds true for our canonical coset sum wavelet system (see Appendix C for proof), but it need not be true for other coset sum wavelet systems. Step (iv) is simply a reverse process of Step (ii) and is possible since the only y_j values we need at this step are the values at even points, and these are already computed in Step (iii).

For a given pair of 1-D masks S and U that generates the canonical coset sum wavelet system, the filters H and G in the algorithm can be computed easily. For example, for the n-D coset sum Haar wavelet system in Example 5, both H and G are the 1-D Haar refinement filter (cf. Example 1). For the coset sum wavelet system in Example 6 that is generated from S_4 (cf. (10)) and U_4 (cf. (9)), the filters are given as

606

$$H(K) = \begin{cases} 1, & K = 0, \\ \frac{9}{16}, & K = \pm 1, \\ -\frac{1}{16}, & K = \pm 3, \\ 0, & \text{otherwise,} \end{cases} G(K) = \begin{cases} \frac{950}{512}, & K = 0, \\ \frac{288}{512}, & K = \pm 1, \\ -\frac{126}{512}, & K = \pm 2, \\ -\frac{32}{512}, & K = \pm 3, \\ \frac{36}{512}, & K = \pm 4, \\ -\frac{2}{512}, & K = \pm 6, \\ 0, & \text{otherwise} \end{cases}$$

We note that the above algorithms for the canonical coset sum wavelet systems are not redundant: the number of coefficients after the decomposition algorithm is approximately the same as the number of input samples, assuming that the filter length of each filter involved in the algorithm is negligible compared to the number of input samples.

Complexity. We measure complexity by counting the number of operations needed in order to fully derive y_{j-1} , and $w_{\nu,j-1}$, $\nu \in \Gamma'$, from y_j , and add the number of operations needed for the reconstruction. Here, we count only multiplicative operations such as multiplication and division, as counting additive operations gives a similar result.

As in the fast tensor product wavelet algorithms discussed in §II-B, the complexity here is linear, i.e. $\sim CN$, with N the number of nonzero entries in y_J , and C some constant independent of y_J . We refer to this constant as the *constant* in the complexity bound or simply as the *complexity constant* throughout this paper.

We now estimate the complexity constant for fast coset sum wavelet algorithms by computing the mean number of operations per single entry in y_J . Suppose α and β are the numbers of nonzero entries of the filters G and H, respectively. Then, the number of operations that are needed to process the portion of y_J that lies on the vertices of a unit cube is the sum of

- 2n-1 (for computing a_G),
- $(2^n 1)(\alpha 1) + n + 1$ (for Step (i)),
- $2(2^n-1)\beta$ (for Step (ii) and (iv)), and
- $(2^n 1)\frac{\alpha + 1}{2} + n 1$ (for Step (iii)).

After computing the sum, we divide it by 2^n , which is the number of vertices in the unit cube, in order to obtain the cost per entry of performing one complete cycle of decomposition/reconstruction. As a result, we get

$$\frac{3}{2}\alpha + 2\beta$$

as an upper bound for the cost per entry. Therefore, the algorithm has complexity $(\frac{3}{2}\alpha + 2\beta)N$, and the constant in the complexity bound in this case is $\frac{3}{2}\alpha + 2\beta$, which does not increase as the spatial dimension *n* increases. A similar argument is used in [25] to compute the complexity constant for the algorithm introduced there.

Contrary to the complexity constant of the fast coset sum wavelet algorithm that we just computed, in the tensor product case the constant grows with the dimension (cf. \S II-B). There are a couple of components that make the coset sum wavelet algorithm this fast. First, as we discussed in §IV-A (cf. (13)), the wavelet masks of the coset sum wavelet system are essentially univariate. Second, as we can see from the above algorithms (cf. Step (iii)), the reconstruction step can be done by completely bypassing the dual wavelet filters. This is reminiscent of the Laplacian pyramid [43] (cf. Appendix B) and its variant [42], which have trivial reconstruction steps that are simply reverse processes of decomposition steps. As a consequence, our algorithm inherits an asymmetry in the roles of the lowpass filters from the Laplacian pyramid. Hence the 1-D lowpass filters G and H in our fast coset sum wavelet algorithm play different roles.

Remark 1. It is well known that any (MRA-based) biorthogonal wavelet system (associated with FIR filters) has decomposition and reconstruction algorithms with linear complexity (see, for example, [1], [29], [31], [35]). In fact, as we alluded to earlier, our fast coset sum decomposition algorithm is nothing but this generic decomposition algorithm for the given canonical coset sum wavelet system. However, our fast coset sum reconstruction algorithm is fundamentally different from this generic reconstruction algorithm: the dual coset sum wavelet filters (cf. (14)) that are not used for our reconstruction algorithm are used for the generic one. As a result, our canonical coset sum wavelet system in Theorem 2 has two different algorithms (the fast coset sum wavelet algorithm and the generic one) and the generic algorithm is always slower than the fast coset sum wavelet algorithm. **Remark 2.** For any biorthogonal wavelet system, multiplying the primal part with some constant factors and dividing the dual part with the same factors will still make a biorthogonal wavelet system. As the functions in these two systems differ only by constants, it is clear that the two systems are essentially the same and most of their properties—including the support and the smoothness—are kept the same.

For the above fast coset sum wavelet algorithms, this means that the decomposition step can be rewritten with explicit *normalization factors* c, d > 0 as

$$y_{j-1}^{new}(k) = cy_{j-1}(k), \quad w_{\nu,j-1}^{new}(k) = dw_{\nu,j-1}(k)$$

where $y_{j-1}(k)$, $w_{\nu,j-1}(k)$ are defined as in Step (i)-(ii), and that the reconstruction step can be modified accordingly: the expressions in the right-hand side of Step (iii)-(iv) can be rewritten in terms of

$$\frac{1}{c}y_{j-1}^{new}(k), \quad \frac{1}{d}w_{\nu,j-1}^{new}(k)$$

in place of $y_{j-1}(k)$, $w_{\nu,j-1}(k)$ that are currently used. In this sense, our original fast coset sum wavelet algorithms can be considered as a special case when c = d = 1. These normalizations are used throughout this paper except in §IV-C (see the discussion below and the footnote in the subsection).

When normalization factors are used for the wavelet algorithms, most properties of the algorithms are not affected. For example, fast coset sum wavelet algorithms with normalization factors will still have the linear complexity with the complexity constant that is independent of n. However the use of different normalization factors may result in different performance in practice [48]. For example, when the algorithms are used for nonlinear approximation with multiple levels (cf. §IV-C), the coefficients are multiplied by constant factors and these factors propagate recursively to other coefficients in lower levels and, as a result, the use of normalization factors may change the relative size of the coefficients.

Below we compare the fast tensor product wavelet algorithms with the fast coset sum wavelet algorithms, both based on the Deslauriers-Dubuc mask and its dual mask in §III-B.

Example 7: Fast tensor product wavelet algorithms vs. fast coset sum wavelet algorithms. In this example, we compare the algorithms for two different families of *n*-D wavelet systems constructed from the same univariate refinement masks by using two different methods: (I) the tensor product and (II) the coset sum. We consider the same univariate refinement masks as in Example 4 and 6, i.e. U_{2k} (interpolatory) and S_{2k} as in (9) and (10), respectively. It is easy to see that the number of nonzero entries of the filter associated with S_{2k} is $\alpha = 8k - 3$, and the number of nonzero entries of the filter associated with U_{2k} is $\beta = 2k + 1$.

Then complexity constant for each algorithm is given as follows:

- (I) (Tensor Product Case) From §II-B, the complexity constant for the fast tensor product algorithm is $(\alpha + \beta)n = (10k - 2)n$, which grows linearly with the dimension.
- (II) (Coset Sum Case) From the above Complexity discussion, the complexity constant for the fast coset sum



(a) Original image "part of lena"





(b) PSNR of reconstructed image



(c) Reconstructed by tensor product

(d) Reconstructed by coset sum

Fig. 11. Comparison of approximation power of tensor product and coset sum for (a) original image "part of lena": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, PSNR = 25.7 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0), (0,1), (1,1)\}$, showing sharper edges and better visual quality, with improved PSNR = 26.5 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%). This experiment shows that the reconstructed images by coset sum have higher PSNR (solid blue), hence better approximation quality than those by tensor product (dotted red) over the range 0.5%-20% for image "part of lena".

wavelet algorithm is $\frac{3}{2}\alpha + 2\beta = \frac{3}{2}(8k-3) + 2(2k+1) = 16k - \frac{5}{2}$, which *does not* grow with the dimension.

Therefore, remarkably, if we fix k (hence the number of vanishing moments of the wavelet system) and increase the dimension n, then the *complexity constant stays the same for the coset sum case, whereas it increases for the tensor product case.*

C. Experiments

In this subsection we present some experimental results of the canonical coset sum wavelet system, in comparison with the tensor product wavelet system. We have implemented the fast coset sum wavelet algorithms in Matlab. The program takes a pair of 1-D biorthogonal refinement filters as input and works for 2-D images. We compare our Matlab program with the standard Matlab implementation of 2-D fast tensor product wavelet algorithms: *wavedec2* (for decomposition) and *waverec2* (for reconstruction) in Wavelet Toolbox [49]. ⁴For the experiments in this subsection, we use two different 2-D

⁴When comparing the implementation of two different wavelet systems, it is important to use the same normalization factors as they may affect the performance (cf. Remark 2 after Complexity discussion). Normalization factors c = d = 2 are used for implementing our coset sum wavelet system since these are the normalization factors used for the tensor product Matlab implementation when seen in terms of a 2-D generalization of the related 1-D concepts (i.e. the DC and Nyquist gains) [50].



Fig. 12. Comparison of approximation power of tensor product and coset sum for (a) original image "wood45": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, with blurry recovered content and PSNR = 24.3 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0), (0,1), (1,1)\}$, showing better approximation to the original image and better improved PSNR = 28.2 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%). The improvement of PSNR in this example is larger than that in "part of lena" example due to the stronger directional content in image "wood45".

wavelet systems obtained from the same 1-D filters, U_4 (shown in Figure 4) and S_4 (shown in Figure 5), but using two different methods, coset sum and tensor product. For the coset sum wavelet system, we initially choose $\Gamma' = \{(1,0), (0,1), (1,1)\}$ as the nonzero coset representatives. The two wavelet systems constructed this way are discussed in Example 6, and the complexity constants of their algorithms are compared in Example 7.

We first compare the running time of the fast coset sum wavelet algorithm with the fast tensor product wavelet algorithm. We apply the two wavelet systems constructed as above to test images, "part of lena"⁵ in Figure 11(a), and "wood45"⁶ in Figure 12(a), both of which have directional content along the diagonal direction. Here, the diagonal direction, or 45° from the positive x-axis, is chosen because it can highlight the benefit of our coset sum wavelet system over the tensor product wavelet system: it is one of the directions that may be captured well by our coset sum system since $\tan 45^\circ = \frac{1}{1}$ and $(1,1) \in \Gamma'$, while it is one of the directions that may not be captured well by the tensor product system since it is not a coordinate direction. Both are of size 256×256 , and we



(c) Reconstructed by tensor product

(d) Reconstructed by coset sum

Fig. 13. An example of image with multiple directions. Comparison of approximation power of tensor product and coset sum for (a) original image "wood": 5-level-down decomposition and reconstruction using 3% largest coefficients. (c) The reconstructed image by tensor product, PSNR = 30.4 dB. (d) The reconstructed image by coset sum with $\Gamma' = \{(1,0), (0,1), (1,1)\}$, PSNR = 31.4 dB. (b) PSNR of reconstructed images over different percentage of retained coefficients (0.5%-20%).

perform 5-level-down decomposition and reconstruction. The running time for "part of lena" is about 0.0279 seconds (s) on average for tensor product algorithm and about 0.0161 s on average for coset sum algorithm, on a Mac 4G 1333MHz laptop. The running time for "wood45" is about 0.0283 s on average for tensor product and about 0.0162 s on average for coset sum. We also tried several other images, both with and without directional content, for various levels of decomposition and reconstruction, and obtained essentially the same results: the coset sum algorithms were faster than the tensor product ones. These experiments confirm our theoretical finding in the previous subsection (cf. Example 7).

We now compare the approximation power of these two wavelet systems. For this, we first decompose a fixed image using the two wavelet systems, then recover the image from the M-largest decomposed coefficients (in magnitude), and finally compare the Peak-Signal-to-Noise-Ratio (PSNR) of the two reconstructed images. The reconstructed image with higher PSNR indicates better approximation to the original image. For the image "part of lena", the reconstructed image using coset sum system shows sharper edges along the diagonal direction and better visual quality than those of tensor product system, and has a slightly higher PSNR (Figure 11(a)(c)(d)). In this experiment, we found that as long as the percentage of retained coefficients is not too large, the coset sum system has slightly higher PSNR (see Figure 11(b) for the range 0.5%-20%). For higher percentage, the coset sum system showed either comparable or slightly worse performance. Another example using

 $^{^5}$ This image is obtained from the image "lena" (512 \times 512) in the image repository http://links.uwaterloo.ca/Repository.html by taking its central part (of size 256 \times 256).

⁶ This image is obtained from the image "wood.000" (512×512) in the SIPI Image Database http://sipi.usc.edu/database/database.php?volume=rotate by rotating 45° clockwise, and taking its central part (of size 256×256).



Fig. 14. Reconstructed images by curvelet from a 5-level-down decomposition and retaining 3% largest coefficients, for original images "part of lena" (Figure 11(a)), "wood45" (Figure 12(a)) and "wood" (Figure 13(a)). Experiments are done by using the wrapping-based method ([51]) implemented in CurveLab Toolbox ([52]) with the default parameter setting. PSNR for reconstructed images are: (a) 25.4 dB (b) 27.7 dB and (c) 30.7 dB.

the texture image "wood45" is also presented (Figure 12). The reconstructed image using coset sum system shows even better performance in this example in terms of PSNR, which is probably due to its stronger directional content. Contrary to the previous experiment with "part of lena", in this experiment, the coset sum system showed consistently higher PSNR for all the percentages. From this experiment, we see that coset sum wavelet system shows promising results when applied to images with strong directional content that matches with the directions of the coset sum primal wavelet filters.

We recall that the directional preference of the coset sum primal wavelet filters can be specified by the associated coset representatives in Γ' . If a dominant direction of the given image does not match with the preferred directions of the coset sum, the coset sum wavelet system may no longer perform well. In such a case, a different set Γ' may be used to match the image's direction (cf. Remark 1 after Definition 1 in §III-A). For example, if the dominant direction is -60° from the positive x-axis, then since $\tan(-60^{\circ}) = -\sqrt{3} \approx \frac{2}{(-1)}$, the coset representative (-1, 2) can be used in place of (1, 0) in the default nonzero coset representatives $\Gamma' = \{(1, 0), (0, 1), (1, 1)\}$.

For many images, it may not be possible to match the directions of the image with the directions of the coset sum. In order to see how the coset sum system would perform for these images in comparison with the tensor product system, we apply the two systems to the test image, "wood"⁷ in Figure 13(a). It has 5 different directional content $(25^{\circ}, 60^{\circ}, 95^{\circ}, 95^{\circ}, 95^{\circ})$ 130° and 165° from the positive x-axis), and it is impossible for us to choose the nonzero coset representatives that match all the directions presented in the image. The reconstructed image using coset sum with the default coset representatives shows sharper edges along the directions near 45° , such as 60° and 25°, and better visual quality than those of tensor product system, and has a slightly higher PSNR (see Figure 13(a)(c)(d)). For the directions that are significantly different from the preferred directions of the coset sum, such as 130° , it does not show sharp edges anymore, but the reproduced image using tensor product does not show sharp edges either.

We found that the overall PSNR result of this image is similar to that of "part of lena": the reconstructed image by coset sum has slightly higher PSNR as long as the percentage of retained coefficients is not too large (see Figure 13(b) for the range 0.5%-20%), but for higher percentage it showed either comparable or slightly worse performance.

We notice that in the reconstructed image in Figure 13(d)certain directions are pronounced more strongly than others despite that the original image in Figure 13(a) does not have that characteristic. This is due to the lack of rotational symmetry ([26]) of the coset sum refinable functions with the default coset representatives (cf. Figure 6 and 9(a)). A remedy for this can be obtained by choosing a set Γ' that gives (roughly) equi-angled directions. For example, when n = 2, by setting $\Gamma' = \{(1, 1), (-4, 1), (1, -4)\}$, a rotational symmetry can be roughly achieved as it gives equi-angled directions $45^{\circ}, 165^{\circ}$, and 285° from the positive x-axis. However, in general it is not easy to overcome the lack of rotational symmetry of the coset sum refinable functions. For example, if we use the above non-default choice of Γ' for $\mathcal{C}_2[S_4]$ and $\mathcal{C}_2[U_4]$ in Example 4, our computation shows that the refinable function associated with $C_2[S_4]$ is still in $L^2(\mathbb{R}^2)$, but the one associated with $C_2[U_4]$ is not. Therefore obtaining coset sum refinable functions that are in $L^2(\mathbb{R}^n)$ with rotational symmetry may not be always possible even for the case of n=2.

As a passing remark, we make a brief comment on comparison to the curvelet system [53], which is a state-of-theart system for representing 2-D and 3-D data effectively using their geometric structure. Before presenting the image experiments using curvelets, we note that any comparison between the curvelet system and the coset sum wavelet system should be made with care as they are very different in nature. For one thing, the curvelet system is not a wavelet system constructed by using a method that works for any multi-D, which is our main interest in this paper. Besides, the curvelet system is highly redundant and its fast algorithm is slower than that of the tensor product and the coset sum system.

With all these in mind, we perform the curvelet transform to the above test images using the Matlab implementation (*fdct_wrapping.m* for decomposition and *ifdct_wrapping.m* for reconstruction) of 2-D discrete curvelets ([51]) in CurveLab

⁷This image is produced by overlying 5 rotated versions of the image "wood.000", which is used to generate the image "wood45" in Figure 12(a), and taking its central part (of size 256×256).

Toolbox [52]. Even after fixing the decomposition level and the percentage of retained coefficients, there are still some parameters to be chosen in the curvelet codes, and the PSNR of reconstructed images is quite sensitive to the choice of these parameters. For reconstructed images using curvelet system with the default parameter setting, the PSNR is either between the PSNR of tensor product and that of coset sum ("wood45" and "wood"), or slightly lower than the PSNR of tensor product ("lena") (see Figure 14). In terms of the visual quality, the curvelet system is superior to the other two systems in both capturing different directional content and keeping rotational symmetry in an image (see Figure 14(c)), but it may add some strong directional artifacts to the reconstructed image (see Figure 14(a)(b)). We conclude that a complete comparison between the coset sum system and the curvelet system requires more thorough study on them.

V. SUMMARY AND OUTLOOK

In this paper we presented the coset sum as an alternative method to the tensor product in constructing decomposable multivariate refinement masks. The decomposable refinement mask constructed by coset sum can be written as the sum, instead of the product, of the univariate refinement masks. We showed that the coset sum can provide many important features of the tensor product, such as preserving the biorthogonality of the univariate refinement masks and the availability of a wavelet system with fast algorithms.

Since the coset sum provides a way to obtain a pair of multivariate biorthogonal refinement masks, it can be combined with any method for finding wavelet masks to construct a (MRA-based biorthogonal) multivariate wavelet system. There has been only limited progress in a systematic construction of non-tensor based multivariate wavelet systems. The coset sum adds a new opportunity to this end.

By specifying wavelet masks as described in §IV-A, we constructed a particular class of coset sum wavelet systems that can be associated with fast algorithms. Such algorithms are referred to as fast coset sum wavelet algorithms.

The fast tensor product wavelet algorithm has linear complexity, but the constant in the complexity bound increases as the spatial dimension increases. On the other hand, the constant in the (linear) complexity bound for the fast coset sum wavelet algorithm is independent of the dimension. Thus, when the spatial dimension is high, the coset sum wavelet algorithm can be *faster* than the tensor product wavelet algorithm.

Coset sum is not necessarily the only alternative to the tensor product. Rather, despite of its limitations in processing images, its existence with desirable features suggests that it may be worthwhile to develop and practice alternative methods to the tensor product for constructing multivariate wavelet systems.

Appendix

A. Proof of Theorem 1

1) Proof of part (a): Suppose H and h are the filters associated with masks R and $C_n[R]$. If R is interpolatory,

it is straightforward to show that $C_n[R]$ is interpolatory. If $C_n[R]$ is interpolatory, by (2), h(0) = 1, and h(k) = 0 if $k \in 2\mathbb{Z}^n \setminus 0$. Then by (8), H(0) = 1. Moreover, H(K) = 0 at all other even points, because if $H(K) \neq 0$ at some even point $K \in 2\mathbb{Z} \setminus 0$, then $h(k) = H(K) \neq 0$ at $k = K\nu \in 2\mathbb{Z}^n \setminus 0$, which contradicts to that $C_n[R]$ is interpolatory. Therefore R is also interpolatory.

2) Proof of part (b): Without loss of generality, we may assume \tilde{R} is interpolatory. We want to show that, $C_n[R]$ and $C_n[\tilde{R}]$ are biorthogonal if and only if R and \tilde{R} are biorthogonal.

Let $R^o := (R - R(\cdot + \pi))/2$ and $R^e := (R + R(\cdot + \pi))/2$ be the odd and even parts of R, respectively, and let \tilde{R}^o be the odd part of \tilde{R} . Since \tilde{R} is interpolatory, the even part of \tilde{R} is the constant 1/2. It is easy to check $\forall \omega_1 \in \mathbb{T}$

$$\overline{R^o(\omega_1)}\tilde{R}^o(\omega_1) = \frac{1}{2} - \frac{1}{2}\overline{R^e(\omega_1)}$$

$$\iff R \text{ and } \tilde{R} \text{ are biorthogonal.}$$

Here, as before, the overline is used to denote the complex conjugate.

We will also need the following identities:

$$\sum_{\gamma \in \pi\Gamma} e^{-i\nu \cdot \gamma} = \begin{cases} 2^n, & \text{if } \nu = 0, \\ 0, & \text{if } \nu \in \Gamma'. \end{cases}$$
(15)

Then from the definition of the coset sum (cf. Definition 1, (6) and (7)), biorthogonal condition (1), and the above identities (15), we have

$$\begin{split} &\mathcal{C}_{n}[R] \text{ and } \mathcal{C}_{n}[\tilde{R}] \text{ are biorthogonal} \\ &\Longleftrightarrow \sum_{\gamma \in \pi\Gamma} (\overline{\mathcal{C}_{n}[R]} \mathcal{C}_{n}[\tilde{R}])(\omega + \gamma) = 1, \quad \forall \omega \in \mathbb{T}^{n} \\ &\Leftrightarrow \left(\frac{1}{2^{n-1}}\right)^{2} \sum_{\gamma \in \pi\Gamma} \left(-2^{n-1} + \sum_{\nu \in \Gamma} \overline{R((\omega + \gamma) \cdot \nu)}\right) \cdot \\ &\left(\frac{1}{2} + \sum_{\tilde{\nu} \in \Gamma'} \left(\tilde{R}((\omega + \gamma) \cdot \tilde{\nu}) - \frac{1}{2}\right)\right) = 1, \quad \forall \omega \in \mathbb{T}^{n} \\ &\Leftrightarrow \sum_{\gamma \in \pi\Gamma} \left(1 - 2^{n-1} + \sum_{\nu \in \Gamma'} e^{i\gamma \cdot \nu} \overline{R^{o}(\omega \cdot \nu)} + \sum_{\nu \in \Gamma'} \overline{R^{e}(\omega \cdot \nu)}\right) \\ &\left(\frac{1}{2} + \sum_{\tilde{\nu} \in \Gamma'} e^{-i\gamma \cdot \tilde{\nu}} \tilde{R}^{o}(\omega \cdot \tilde{\nu})\right) = (2^{n-1})^{2}, \quad \forall \omega \in \mathbb{T}^{n} \\ &\Leftrightarrow 2^{n-1}(1 - 2^{n-1}) + 2^{n-1} \sum_{\nu \in \Gamma'} \overline{R^{e}(\omega \cdot \nu)} \\ &+ 2^{n} \sum_{\nu \in \Gamma'} \overline{R^{o}(\omega \cdot \nu)} \tilde{R}^{o}(\omega \cdot \nu) = (2^{n-1})^{2}, \quad \forall \omega \in \mathbb{T}^{n} \\ &\Leftrightarrow \overline{R^{o}(\omega_{1})} \tilde{R}^{o}(\omega_{1}) = \frac{1}{2} - \frac{1}{2} \overline{R^{e}(\omega_{1})}, \quad \forall \omega_{1} \in \mathbb{T}. \end{split}$$

Therefore, $C_n[R]$ and $C_n[\tilde{R}]$ are biorthogonal if and only if R and \tilde{R} are biorthogonal.

3) Proof of part (c): Let R be a univariate interpolatory refinement mask with accuracy number m. First let us prove the accuracy number of $C_n[R]$ is at least m. Since R has accuracy number m,

$$(D^k R)(\pi) = 0, \ \forall 0 \le k \le m-1, \ \text{and} \ (D^m R)(\pi) \ne 0.$$
 (16)

Furthermore, since R is interpolatory, $1 - R(\omega) = R(\omega + \pi)$ holds for all $\omega \in \mathbb{T}$. Hence $(D^k(1-R))(0) = (D^kR)(\pi)$ for all $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Thus 1 - R has a zero of order m at the origin, i.e.

$$R(0) = 1$$

(D^kR)(0) = 0, $\forall 1 \le k \le m - 1$ (17)
(D^mR)(0) $\ne 0$.

Now consider the *n*-D refinement mask $C_n[R]$. The accuracy number of $C_n[R]$ is at least one, i.e. $C_n[R](\gamma) = 0$, for all $\gamma \in \pi \Gamma'$. To see this, we need the dual identities of (15):

$$\sum_{\nu \in \Gamma} e^{-i\nu \cdot \gamma} = \begin{cases} 2^n, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi \Gamma'. \end{cases}$$
(18)

From (18), we can read off

$$\#\{\nu \in \Gamma' : \gamma \cdot \nu \equiv \pi \,(\text{mod}\,2\pi\mathbb{Z})\} = 2^{n-1},\qquad(19)$$

for all $\gamma \in \pi\Gamma'$. In particular, the left-hand side of (19) is independent of γ . We then have for any $\gamma \in \pi\Gamma'$

$$2^{n-1}\mathcal{C}_n[R](\gamma) = -2^{n-1} + \sum_{\nu \in \Gamma} R(\gamma \cdot \nu)$$
$$= 1 - 2^{n-1} + \sum_{\{\nu \in \Gamma': \gamma \cdot \nu \equiv 0\}} R(\gamma \cdot \nu) + \sum_{\{\nu \in \Gamma': \gamma \cdot \nu \equiv \pi\}} R(\gamma \cdot \nu)$$
$$= 0,$$

where \equiv in the second line is used to denote congruence in modulo $2\pi\mathbb{Z}$, and the last equality is from the conditions R(0) = 1, $R(\pi) = 0$ and the identity (19). Furthermore, for all $\gamma \in \pi\Gamma'$ and for all $\mu \in \mathbb{N}_0^n$ with $1 \leq |\mu| \leq m - 1$ $(|\mu| := \mu_1 + \cdots + \mu_n)$

$$(D^{\mu}\mathcal{C}_{n}[R])(\gamma) = \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} (D^{\mu}[R(\omega \cdot \nu)])_{|\omega=\gamma}$$
$$= \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} \left(\prod_{j=1}^{n} \nu_{j}^{\mu_{j}}\right) (D^{|\mu|}R)(\gamma \cdot \nu) = 0,$$

where the last equality is from the identities (16) and (17). Therefore the accuracy number of $C_n[R]$ is at least m.

Next we prove the accuracy number of $C_n[R]$ is exactly m by contradiction. Suppose the accuracy number of $C_n[R]$ is m + l with $l \ge 1$. Then

$$(D^{\mu}\mathcal{C}_{n}[R])(\gamma) = 0,$$

 $\forall \gamma \in \pi\Gamma' \text{ and } \forall \mu \in \mathbb{N}_{0}^{n} \text{ with } 0 \leq |\mu| \leq m + l - 1.$

Since the univariate interpolatory R and the multivariate interpolatory $C_n[R]$ are connected as follows:

$$R(\omega) = \mathcal{C}_n[R](\omega, 0, \cdots, 0), \quad \forall \omega \in \mathbb{T},$$

we have $(D^k R)(\pi) = D^{(k,0,\cdots,0)} C_n[R](\pi,0,\cdots,0) = 0$ for all $0 \le k \le m+l-1$. Hence the accuracy number of Ris at least m+l, which contradicts to the given assumption. Therefore the accuracy number of $C_n[R]$ has to be m.

B. Proof of Theorem 2

In this subsection we prove Theorem 2. In the proof we use the concepts of Compression-Alignment-Prediction (CAP) and Compression-Alignment-Modified-Prediction (CAMP) [54]. CAMP is a variant of CAP, and CAP is a generalization of the Laplacian pyramid [43]. In particular, CAP without alignment operator is the same as Laplacian pyramid. It is well known that Laplacian pyramid has a trivial reconstruction algorithm of reversing the steps in its decomposition algorithm. Both CAP and CAMP are originally designed for the redundant wavelet construction, and CAMP is introduced in order to achieve a better space localization than CAP.

Given $\tau := C_n[S], \tau^d := C_n[U]$ with interpolatory U, and $t_{\nu}(\omega) := e^{-i\nu\cdot\omega}\overline{U}(\omega\cdot\nu+\pi), \ \omega \in \mathbb{T}^n, \ \nu \in \Gamma'$, we want to show that there exist dual wavelet masks $t_{\nu}^{d}(\omega)$ such that $(\tau, (t_{\nu})_{\nu\in\Gamma'})$ and $(\tau^d, (t_{\nu}^d)_{\nu\in\Gamma'})$ satisfy the MUEP conditions in (3).

To show this, first let us construct another pair of wavelet masks $(\tau_{\nu})_{\nu\in\Gamma}$ and dual wavelet masks $(\tau_{\nu}^{d})_{\nu\in\Gamma}$, which we know for sure satisfy the MUEP conditions with τ and τ^{d} .

First extend the definition of t_{ν} by defining t_0 :

$$t_{\nu}(\omega) := \begin{cases} \frac{1}{2}(1 - \tau(\omega)), & \text{if } \nu = 0, \\ e^{-i\nu \cdot \omega} \overline{U(\omega \cdot \nu + \pi)}, & \text{if } \nu \in \Gamma'. \end{cases}$$

Then from [54] we know that

$$t_{\nu}(\omega) = 2^{\frac{n}{2}-1} \cdot t_{-\nu}^{CAMP}(\omega), \quad \nu \in \Gamma,$$
(20)

where t_{ν}^{CAMP} is the CAMPlet mask in Section 2.3 of [54].

Furthermore by comparing the CAPlet masks in Lemma 2.2 of [54] with the CAMPlet masks, it is easy to see that they are related as

$$t_{\nu}^{CAP}(\omega) - t_{\nu}^{CAMP}(\omega) = \begin{cases} 0, & \text{if } \nu = 0, \\ f_{-\nu}(\omega)t_0^{CAMP}(\omega), & \text{if } \nu \in \Gamma', \end{cases}$$
(21)

where $f_{\nu}(\omega) = e^{-i\nu\cdot\omega} \sum_{\gamma \in \pi\Gamma} e^{-i\nu\cdot\gamma} \overline{\tau^{\mathrm{d}}(\omega+\gamma)}$. Here it is necessary to point out that f_{ν} is π -periodic, i.e. $f_{\nu}(\omega+\gamma) = f_{\nu}(\omega)$, for any $\gamma \in \pi\Gamma$.

Now define $(\tau_{\nu})_{\nu \in \Gamma}$

$$\tau_{\nu}(\omega) := 2^{\frac{n}{2}-1} t_{-\nu}^{CAP}(\omega), \quad \nu \in \Gamma.$$
(22)

Then since CAP without alignment operator is the same as Laplacian pyramid, and Laplacian pyramid has a trivial reconstruction, we know that with

$$\tau^{\mathrm{d}}_{\nu}(\omega) := \begin{cases} 2^{1-n}, & \text{if } \nu = 0, \\ 2^{1-n} e^{-i\nu \cdot \omega}, & \text{if } \nu \in \Gamma', \end{cases}$$

 $(\tau, (\tau_{\nu})_{\nu \in \Gamma})$ and $(\tau^{d}, (\tau^{d}_{\nu})_{\nu \in \Gamma})$ satisfy the MUEP conditions. Next, we start from the MUEP conditions of $(\tau, (\tau_{\nu})_{\nu \in \Gamma})$ and $(\tau^{d}, (\tau^{d}_{\nu})_{\nu \in \Gamma})$ to find our dual wavelet masks t^{d}_{ν} . To do that, we need three more identities. The first one is a simple observation that can be obtained from (20), (21) and (22):

$$\tau_{\nu}(\omega) - t_{\nu}(\omega) = \begin{cases} 0, & \text{if } \nu = 0, \\ f_{\nu}(\omega)t_0(\omega), & \text{if } \nu \in \Gamma'. \end{cases}$$
(23)

The second one can be derived from the interpolatory property of τ^{d} and the identities (18):

$$\tau_0^{\rm d}(\omega) + \sum_{\nu \in \Gamma'} \overline{f_{\nu}(\omega)} \tau_{\nu}^{\rm d}(\omega) = 2\tau^{\rm d}(\omega).$$
(24)

After defining $g_{\nu}(\omega) := e^{-i\nu \cdot \omega} \sum_{\gamma \in \pi\Gamma} e^{-i\nu \cdot \gamma} \overline{\tau(\omega + \gamma)}$, the third identity:

$$t_0(\omega) + 2^{-n} \sum_{\nu \in \Gamma'} t_{\nu}(\omega) \overline{g_{\nu}(\omega)} = 0$$
(25)

can be shown from the biorthogonality between τ and τ^{d} and the identities (18). Finally from the above identities (23), (24) and (25), with

$$\delta_{\gamma 0} := \begin{cases} 1, & \text{if } \gamma = 0, \\ 0, & \text{if } \gamma \in \pi \Gamma', \end{cases}$$

we get

$$\begin{split} & \frac{\partial_{\gamma 0}}{=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega)+\overline{\tau_{0}(\omega+\gamma)}\tau_{0}^{d}(\omega)+\sum_{\nu\in\Gamma'}\overline{\tau_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega)} \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega)+\overline{t_{0}(\omega+\gamma)}\tau_{0}^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{f_{\nu}(\omega+\gamma)}\tau^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &+\overline{t_{0}(\omega+\gamma)}\left(\tau_{0}^{d}(\omega)+\sum_{\nu\in\Gamma'}\overline{f_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega)\right) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega)+\overline{t_{0}(\omega+\gamma)}2\tau^{d}(\omega)+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &+\left(\overline{t_{0}(\omega+\gamma)}+2^{-n}\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}g_{\nu}(\omega+\gamma)\right)2\tau^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega)-2^{1-n}\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}g_{\nu}(\omega+\gamma)\tau^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) \\ &+\sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) \\ &=\overline{\tau(\omega+\gamma)}\tau^{d}(\omega) + \sum_{\nu\in\Gamma'}\overline{t_{\nu}(\omega+\gamma)}\tau_{\nu}^{d}(\omega) + \tau_{\nu}^{d}(\omega)) \,. \end{split}$$

Therefore, by letting $t_{\nu}^{d} := -2^{1-n}g_{\nu}\tau^{d} + \tau_{\nu}^{d}$, we find the dual wavelet masks t_{ν}^{d} , $\nu \in \Gamma'$, such that $(\tau, (t_{\nu})_{\nu \in \Gamma'})$ and $(\tau^{d}, (t_{\nu}^{d})_{\nu \in \Gamma'})$ satisfy the MUEP conditions.

C. Proof of the identity in Step (iii) of the coset sum algorithm in $\IV-B$

In this subsection we verify the identity in Step (iii) of Reconstruction Algorithm in \S IV-B. We use the same notation as in the algorithm. In particular, G and H are univariate refinement filters associated with biorthogonal refinement masks S and U, respectively, and H is interpolatory.

From Step (i) of the algorithm in §IV-B, we know that, with $a_G = 2^n - (2^n - 1)(2 - G(0))$,

$$a_{G} y_{j}(2k) = 2^{n} y_{j-1}(k) - \sum_{\nu \in \Gamma'} \sum_{L \in \mathbb{Z} \setminus 0} G(L) y_{j}(2k + L\nu)$$

$$= 2^{n} y_{j-1}(k) - \sum_{\nu \in \Gamma'} \sum_{L \equiv 1} G(L) y_{j}(2k + L\nu)$$

$$- \sum_{\nu \in \Gamma'} \sum_{L \equiv 0, L \neq 0} G(L) y_{j}(2k + L\nu)$$
(26)

where \equiv is used to denote congruence in modulo $2\mathbb{Z}$. Since the masks S and U are biorthogonal, from (1) and the connection between the filter and the mask, it is easy to see that the associated filters G and H satisfy the following condition:

$$\sum_{m \in \mathbb{Z}} G(L+m)H(m) = \begin{cases} 0, & \text{if } L \equiv 0, L \neq 0, \\ 2, & \text{if } L = 0. \end{cases}$$

Combining this with the fact that H is interpolatory leads to

$$\sum_{m \equiv 1} G(L+m)H(m) = \begin{cases} 0 - G(L), & \text{if } L \equiv 0, L \neq 0, \\ 2 - G(0), & \text{if } L = 0. \end{cases}$$

From this and the change of variables, we see that

$$\begin{split} &\sum_{\nu \in \Gamma'} \sum_{L \equiv 0, L \neq 0} G(L) y_j(2k + L\nu) \\ &= \sum_{\nu \in \Gamma'} \sum_{L \equiv 0, L \neq 0} \left(0 - \sum_{m \equiv 1} G(L + m) H(m) \right) y_j(2k + L\nu) \\ &= -\sum_{\nu \in \Gamma'} \sum_{m \equiv 1} \sum_{L \equiv 0} G(L + m) H(m) y_j(2k + L\nu) \\ &+ \sum_{\nu \in \Gamma'} \sum_{m \equiv 1} G(m) H(m) y_j(2k) \\ &= -\sum_{\nu \in \Gamma'} \sum_{m \equiv 1} \sum_{n \equiv 1} G(n) H(m) y_j(2k + (n - m)\nu) \\ &+ (2^n - 1)(2 - G(0)) y_j(2k) \\ &= -\sum_{\nu \in \Gamma'} \sum_{L \equiv 1} \sum_{m \equiv 1} G(L) H(m) y_j(2k + (L - m)\nu) \\ &+ (2^n - 1)(2 - G(0)) y_j(2k) \end{split}$$

By substituting this result to (26) and solving for $y_j(2k)$, we obtain

$$y_j(2k) = y_{j-1}(k) - \frac{1}{2^{n-1}} \sum_{\nu \in \Gamma'} \sum_{L \in \mathbb{Z}} G(2L+1) w_{\nu,j-1}(k+L\nu)$$

as desired.

ACKNOWLEDGMENT

The authors thank the editor and referees for suggestions that improved the clarity and accessibility of this article.

REFERENCES

- S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, 1989.
- [2] S. D. Riemenschneider and Z. Shen, "Construction of compactly supported biorthogonal wavelets in L₂(ℝ^d) II," in Wavelet Applications Signal and Image Processing VII, 1999, pp. 264–272.

- [3] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for ℝⁿ," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 533–555, 1992.
- [4] A. Cohen and J.-M. Schlenker, "Compactly supported bidimensional wavelet bases with hexagonal symmetry," *Constr. Approx.*, vol. 9, no. 2-3, pp. 209–236, 1993.
- [5] J. Kovačević and M. Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Trans. Signal Processing*, vol. 43, no. 5, pp. 1269–1273, 1995.
- [6] C. A. Micchelli and Y. Xu, "Reconstruction and decomposition algorithms for biorthogonal multiwavelets," *Multidimensional Systems and Signal Processing*, vol. 8, pp. 31–69, 1997.
- [7] S. D. Riemenschneider and Z. Shen, "Multidimensional interpolatory subdivision schemes," *SIAM J. Numer. Anal.*, vol. 34, pp. 2357–2381, 1997.
- [8] W. He and M.-J. Lai, "Construction of bivariate compactly supported biorthogonal box spline wavelets with arbitrarily high regularities," *Appl. Comput. Harmon. Anal.*, vol. 6, pp. 53–74, 1999.
- [9] H. Ji, S. D. Riemenschneider, and Z. Shen, "Multivariate compactly supported fundamental refinable functions, duals and biorthogonal wavelets," *Stud. Appl. Math.*, vol. 102, pp. 173–204, 1999.
- [10] M. Salvatori and P. M. Soardi, "Multivariate compactly supported biorthogonal spline wavelets," *Annali di Matematica*, vol. 181, no. 2, pp. 161–179, 2002.
- [11] Q. Chen, C. A. Micchelli, S. Peng, and Y. Xu, "Multivariate filter banks having matrix factorizations," *SIAM J. Matrix Anal. Appl.*, vol. 25, pp. 517–531, 2003.
- [12] Q. Chen, C. A. Micchelli, and Y. Xu, "Biorthogonal multivariate filter banks from centrally symmetric matrices," *Linear Algebra and its Applications*, vol. 402, pp. 111–125, 2005.
- [13] Z. He, X. You, and Y. Yuan, "Texture image retrieval based on nontensor product wavelet filter banks," *Signal Process.*, vol. 89, pp. 1501– 1510, 2009.
- [14] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, "Adaptive nonseparable wavelet transform via lifting and its application to content-based image retrieval," *IEEE Trans. Image Processing*, vol. 19, no. 1, pp. 25–35, 2010.
- [15] X. You, L. Du, Y. Cheung, and Q. Chen, "A blind watermarking scheme using new nontensor product wavelet filter banks," *IEEE Trans. Image Processing*, vol. 19, no. 12, pp. 3271–3284, 2010.
- [16] Z. Zhang, "A new method of constructions of non-tensor product wavelets," *Acta Appl. Math.*, vol. 111, pp. 153–169, 2010.
- [17] D. B. H. Tay and N. G. Kingsbury, "Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 466–480, 1993.
- [18] S.-M. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Trans. Signal Processing*, vol. 43, no. 3, pp. 649–665, 1995.
- [19] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 882–893, 1992.
- [20] I. A. Shah and A. A. C. Kalker, "Theory and design of multidimensional QMF sub-band filters from 1-D filters and polynomials using transforms," *Communications, Speech and Vision, IEE Proceedings I*, vol. 140, no. 1, pp. 67–71, 1993.
- [21] R. Ansari and C.-L. Lau, "Two-dimensional IIR filters for exact reconstruction in tree-structured sub-band decomposition," *Electronics Letters*, vol. 23, no. 12, pp. 633–634, 1987.
- [22] T. A. C. M. Kalker and I. A. Shah, "A group theoretic approach to multidimensional filter banks: theory and applications," *IEEE Trans. Signal Processing*, vol. 44, no. 6, pp. 1392–1405, 1996.
- [23] J. M. Shapiro, "Adaptive McClellan transformations for quincunx filter banks," *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 642–648, 1994.
- [24] Y. Hur and A. Ron, "New constructions of piecewise-constant wavelets," *Electronic Transactions on Numerical Analysis*, vol. 25, pp. 138–157, 2006.
- [25] —, "L-CAMP: Extremely local high-performance wavelet representations in high spatial dimension," *IEEE Trans. Inform. Theory*, vol. 54, pp. 2196–2209, 2008.
- [26] Rajan, P. and Reddy, H. and Swamy, M., "Fourfold rotational symmetry in two-dimensional functions," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, no. 3, pp. 488–499, 1982.
- [27] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli, *Stationary subdivision*. Memoirs of Amer. Math. Soc., 1991, vol. 93.

- [28] B. Han and R.-Q. Jia, "Optimal interpolatory subdivision schemes in multidimensional spaces," *SIAM J. Numer. Anal.*, vol. 36, pp. 105–124, 1999.
- [29] G. Strang and T. Nguyen, Wavelets and Filter Banks. Wellesley: Wellesley-Cambridge Press, 1997.
- [30] A. Ron and Z. Shen, "Affine systems in $L_2(\mathbb{R}^d)$ II: dual systems," J. Fourier Anal. Appl., vol. 3, no. 5, pp. 617–637, 1997.
- [31] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.
- [32] Y. Hur and A. Ron, "High-performance very local Riesz wavelet bases of L₂(ℝⁿ)," SIAM Journal on Mathematical Analysis, vol. 44, pp. 2237– 2265, 2012.
- [33] D.-R. Chen, B. Han, and S. D. Riemenschneider, "Construction of multivariate biorthogonal wavelets with arbitrary vanishing moments," *Adv. Comput. Math.*, vol. 13, no. 2, pp. 131–165, 2000.
- [34] Y. Meyer, Wavelets and Operators. Cambridge: Cambridge University Press, 1992.
- [35] S. G. Mallat, A Wavelet Tour of Signal Processing. San Diego, CA: Academic Press, 1999.
- [36] B. Han, "Analysis and construction of optimal multivariate biorthogonal wavelets with compact support," *SIAM Journal on Math. Analysis*, vol. 31, pp. 274–304, 2000.
- [37] M. Nielsen, "On polynomial symbols for subdivision schemes," Adv. Comput. Math, vol. 27, pp. 195–209, 2007.
- [38] C. De Boor, K. Höllig, and S. Riemenschneider, *Box Splines*. New York, NY: Springer-Verlag, 1993.
- [39] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Soc. Ind. Appl. Math., 1992.
- [40] G. Deslauriers and S. Dubuc, "Interpolation dyadique," in *Fractals, Dimensions Non Entières et Applications*, Masson, Paris, 1987, pp. 44–55.
- [41] H. Ji and Z. Shen, "Compactly supported (bi)orthogonal wavelets generated by interpolatory refinable functions," *Adv. Comput. Math*, vol. 11, pp. 81–104, 1999.
- [42] Y. Hur, "Effortless critical representation of Laplacian pyramid," *IEEE Trans. Signal Processing*, vol. 58, pp. 5584–5596, 2010.
- [43] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.
- [44] W. Lawton, S. L. Lee, and Z. Shen, "An algorithm for matrix extension and wavelet construction," *Math. Comp.*, vol. 65, pp. 723–737, 1996.
- [45] H. Park, "Optimal design of synthesis filters in multidimensional perfect reconstruction FIR filter banks using Gröbner bases," *IEEE Trans. Circuits Syst.*, vol. 49, pp. 843–851, 2002.
- [46] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.
- [47] J. Kovačević and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," *IEEE Trans. Signal Processing*, vol. 9, no. 3, pp. 480–496, 2000.
- [48] José Salvador Oliver Gil, "On the design of fast and efficient wavelet image coders with reduced memory usage," Ph.D. dissertation, Universitat Politécnica de Valéncia, 2006.
- [49] Wavelet Toolbox, Version 4.7 (R2011a). Natick, Massachusetts: The MathWorks Inc., 2011.
- [50] Rabbani, M. and Joshi, R.L. and Jones, P.W., "JPEG 2000 core coding system (part 1)," *The JPEG 2000 Suite*, pp. 1–69, 2009.
- [51] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale modeling and simulation*, vol. 5, no. 3, pp. 861– 899, 2006.
- [52] CurveLab Toolbox, Version 2.1.3. Candès, E. and Demanet, L. and Ying, L., 2008.
- [53] E. Candès and D. Donoho, "Curvelets a surprisingly effective nonadaptive representation for objects with edges," in *Curves and Surfaces*, 2000, pp. 105–120.
- [54] Y. Hur and A. Ron, "CAPlets: wavelet representations without wavelets I - the algorithms," 2009, preprint.

Youngmi Hur (M'11) received the B.S. and M.S. degrees in mathematics from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in mathematics from University of Wisconsin at Madison in 2006.

From 2006 to 2008, she was a C.L.E. Moore instructor at the Department of Mathematics of the Massachusetts Institute of Technology, Cambridge, MA. Since 2008, she is an Assistant Professor at the Department of Applied Mathematics and Statistics of the Johns Hopkins University, Baltimore, MD. Her research interests are in the field of applied and computational harmonic analysis including wavelets and their applications.

Fang Zheng is currently a Ph.D candidate in the Department of Applied Mathematics and Statistics at Johns Hopkins University, under the supervision of Prof. Youngmi Hur. Her research interests include multivariate wavelet system and its applications. Fang Zheng received a Bachelors degree in Mathematics from Beijing Forestry University, China in 2008, and a Masters degree in Applied Mathematics and Statistics from Johns Hopkins University in 2010.