Smooth Affine Shear Tight Frames: Digitization and Applications

Xiaosheng Zhuang[a]

[a] Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong

ABSTRACT

In this paper, we mainly discuss one of the recent developed directional multiscale representation systems: smooth affine shear tight frames. A directional wavelet tight frame is generated by isotropic dilations and translations of directional wavelet generators, while an affine shear tight frame is generated by anisotropic dilations, shears, and translations of shearlet generators. These two tight frames are actually connected in the sense that the affine shear tight frame can be obtained from a directional wavelet tight frame through subsampling. Consequently, an affine shear tight frame indeed has an underlying filter bank from the MRA structure of its associated directional wavelet tight frame. We call such filter banks affine shear filter banks, which can be designed completely in the frequency domain. We discuss the digitization of affine shear filter banks and their implementations: the forward and backward digital affine shear transforms. Redundancy rate and computational complexity of digital affine shear transforms are also investigated in this paper. Numerical experiments and comparisons in image/video processing show the advantages of digital affine shear transforms over many other stat-of-art directional multiscale representation systems.

Keywords: directional multiscale representation systems, smooth affine shear tight frames, directional framelets, directional filter banks, digital affine shear transforms, image processing, video processing, denoising, dual-tree complex wavelets, tensor product complex tight framelets, shearlets, curvelets, contourlets, surfacelets

1. INTRODUCTION

Since the pioneer work by Meyer, Mallat, Daubechies, etc., wavelet analysis [5] has had a tremendous impact on many fields such as electric engineering, image/signal processing, computer graphics, numerical solutions of PDE, and so on. On the other hand, applications of wavelets in turn require the development of wavelet systems with more and more desirable properties, e.g., high order of vanishing moments, symmetry, and regularity. One of the lines of development is to design wavelet systems with directionality due to the fact that tensor product real-value wavelets lack the directional sensitivity and sparse approximation for anisotropic features in dimension higher than one. Along this line of development are the emergence of many directional multiscale representation systems, such as dual-tree complex wavelets [18], tensor product complex tight framelets [11,12], curvelets [2,3], contourlets [4], surfacelets [17], shearlets [1,6–9,14–16], and so on. In this paper, we shall mainly discuss one of the directional multiscale representation systems recently developed by the author and his collaborator: smooth affine shear tight frames [13].

Using the framework of frequency-based affine systems [10], smooth affine shear tight frames have been study systematically in dimension two in [13]. It has been shown that smooth affine shear tight frames include all known shearlet tight frames as special cases. Characterization in [13] for an affine shear tight frame greatly simplified the construction of affine shear tight frames and smooth affine shear tight frames with all their generators in the Schwartz class have been constructed for $L_2(\mathbb{R}^2)$ in [13]. One of the key results in [13] also shows that an affine shear tight frame can be regarded as a subsampled system from a directional affine wavelet tight frame that has an MRA structure thereby associating an affine shear tight frame with a underlying filter bank. This key result indicates that affine shear tight frames are less 'redundant' than directional affine wavelet tight frames. More importantly, digitization of affine shear tight frames can be efficiently implemented using their underlying filter banks and are very similar to the standard fast wavelet transforms.

Send correspondence to Email: xzhuang7@city.edu.hk, Phone: +852 3442 5942.

In this paper, we further investigate d-dimensional smooth affine shear tight frames for any $d \geqslant 2$ and provide the characterization and construction of d-dimensional smooth affine shear tight frames (Sec. 2). We also discuss the digitization of d-dimensional smooth affine shear tight frames: the forward and backward of digital affine shear transforms, as well as the redundancy rate and computational complexity of d-dimensional digital affine shear transforms (Sec. 3). Numerical experiments and comparisons on image/video denoising shall be given (Sec. 4).

2. SMOOTH AFFINE SHEAR TIGHT FRAMES IN $\mathbb{R}^d$

In this section, we extend the definition of smooth affine shear tight frames in [13] in dimension two to any dimension $d \geqslant 2$. We first introduce the notation of a sequence of d-dimensional affine shear systems and then provide the characterization for an affine shear system to be an affine shear tight frame in $L_2(\mathbb{R}^d)$ as well as the construction of both d-dimensional affine shear tight frames and d-dimensional affine wavelet tight frames.

2.1 Characterizaiton of smooth affine shear tight frames

Let U be a $d \times d$ invertible matrix. Throughout the paper, we shall assume $d \geqslant 2$ and use the compact notation $f_{U;\mathsf{k},\mathsf{n}}(x) := |\det U|^{1/2} f(Ux - \mathsf{k}) e^{-i\mathsf{n}\cdot Ux}, \mathsf{k}, \mathsf{n}, x \in \mathbb{R}^d$ to encode dilation U, translation k, and modulation n for a function f in $\mathbb{R}^d$. The shear operator $S^{\vec{\tau}}$ with $\vec{\tau} = (\tau_2, \ldots, \tau_d) \in \mathbb{R}^{d-1}$, anisotropic dilation matrix A_λ, and isotropic dilation matrix M_λ with $\lambda > 1$ are of the form:

$$S^{\vec{\tau}} = \begin{bmatrix} 1 & \tau_2 & \ldots & \tau_d \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}, \quad \mathsf{A}_\lambda = \begin{bmatrix} \lambda^2 & 0 & \ldots & 0 \\ 0 & \lambda & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda \end{bmatrix}, \quad \text{and} \quad \mathsf{M}_\lambda = \begin{bmatrix} \lambda^2 & 0 & \ldots & 0 \\ 0 & \lambda^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda^2 \end{bmatrix}.$$

We shall use $\mathsf{N}_\lambda := \mathsf{M}_\lambda^{-\mathsf{T}}$ and $\mathsf{B}_\lambda := \mathsf{A}_\lambda^{-\mathsf{T}}$ to denote the transpose of the inverse of M_λ and A_λ, respectively. Note that $\mathsf{M}_\lambda = \mathsf{A}_\lambda \mathsf{D}_\lambda$ with $\mathsf{D}_\lambda := \mathrm{diag}(1, \lambda \mathsf{I}_{d-1})$, where I_n denotes the $n \times n$ identity matrix. Define $S_{\vec{\tau}} := (S^{\vec{\tau}})^{\mathsf{T}}$ and denote E_n to be the elementary matrix corresponding to the coordinate exchange between the first axis and the nth one. For example, $\mathsf{E}_1 = \mathsf{I}_d$, $\mathsf{E}_2 = \mathrm{diag}(\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right], \mathsf{I}_{d-2})$, $\mathsf{E}_3 = \mathrm{diag}(\left[\begin{smallmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{smallmatrix}\right], \mathsf{I}_{d-3})$, and so on.

Let Ψ_j be a set of generators for high frequency part at scale j be given by $\Psi_j := \{\psi^{j,\vec{\ell}}(S^{-\vec{\ell}}\cdot) : |\vec{\ell}| \leqslant \vec{r}_j\}$ with $\vec{\ell} := (\ell_2, \ldots, \ell_d) \in \mathbb{Z}^{d-1}$, $\vec{r}_j := (r_{j,2}, \ldots, r_{j,d}) \in \mathbb{Z}^{d-1}$, and $\psi^{j,\vec{\ell}}$ being functions in $L_2(\mathbb{R}^d)$. Here and after we shall use the compact notation $\sum_{\vec{\ell}=-\vec{r}_j}^{\vec{r}_j}$ to mean $\sum_{\ell_2=-r_{j,2}}^{r_{j,2}} \cdots \sum_{\ell_d=-r_{j,d}}^{r_{j,d}}$ and $|\vec{\ell}| \leqslant \vec{r}_j$ to mean $|\ell_2| \leqslant r_{j,2}, \ldots, |\ell_d| \leqslant r_{j,d}$. Let $\varphi^j \in L_2(\mathbb{R}^d)$ be the scaling function for the low frequency part at scale j. Then a *d-dimensional affine shear system* (starting at scale J) is defined to be

$$\mathsf{AS}_J(\varphi^J; \{\Psi_j\}_{j=J}^\infty) = \{\varphi^J_{\mathsf{M}_\lambda^J;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d\} \cup \{h_{\mathsf{A}_\lambda^j \mathsf{E}_n;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d, n = 1, \ldots, d,\ h \in \Psi_j\}_{j=J}^\infty. \tag{2.1}$$

The Fourier transform $\widehat{f}$ of a function $f \in L_1(\mathbb{R}^d)$ is defined to be $\widehat{f}(\xi) := \int_{\mathbb{R}^d} f(x) e^{-ix\cdot\xi} dx$ for $\xi \in \mathbb{R}^d$ and can be naturally extended to functions in $L_2(\mathbb{R}^d)$. We denote by $\mathscr{D}(\mathbb{R}^d)$ the linear space of all compactly supported C^∞ (test) functions with the usual topology. Following the same lines of proof in Theorem 2 of [13]. We have the following simple characterization for a sequence of affine shear systems $\mathsf{AS}_J(\varphi^J; \{\Psi_j\}_{j=J}^\infty)$, $J \geqslant J_0$ to be a sequence of affine shear tight frames for $L_2(\mathbb{R}^d)$ when all elements $\widehat{h}$ are nonnegative for $h \in \{\{\varphi^j\} \cup \Psi_j\}_{j=J_0}^\infty$ (also see [10, Corollary 18]):

THEOREM 2.1. *Let J_0 be an integer and $\mathsf{AS}_J(\varphi^J; \{\Psi_j\}_{j=J}^\infty)$ be defined as in (2.1). Suppose that $\widehat{h} \geqslant 0$ for all $h \in \{\{\varphi^j\} \cup \Psi_j\}_{j=J}^\infty$. Then, for all integers $J \geqslant J_0$, $\mathsf{AS}_J(\varphi^J; \{\Psi_j\}_{j=J}^\infty)$ is an affine shear tight frame for $L_2(\mathbb{R}^d)$; that is, all generators are from $L_2(\mathbb{R}^d)$ and*

$$\|f\|_2^2 = \sum_{\mathsf{k}\in\mathbb{Z}^d} |\langle f, \varphi_{\mathsf{M}_\lambda^J;\mathsf{k}}\rangle|^2 + \sum_{j=J}^\infty \sum_{n=1}^d \sum_{h\in\Psi_j} \sum_{\mathsf{k}\in\mathbb{Z}^d} |\langle f, h_{\mathsf{A}_\lambda^j \mathsf{E}_n;\mathsf{k}}\rangle|^2, \quad \forall f \in L_2(\mathbb{R}^d), \tag{2.2}$$

if and only if the following holds:

$$\begin{aligned}
&\widehat{h}(\xi)\widehat{h}(\xi+2\pi\mathsf{k}) =0, \quad a.e., \xi\in\mathbb{R}^d,\ \mathsf{k}\in\mathbb{Z}^d\backslash\{0\},\ \forall h\in\{\{\varphi^j\}\cup\Psi_j\}_{j=J}^{\infty},\\
&|\widehat{\varphi^{j+1}}(\mathsf{N}_\lambda^{j+1}\xi)|^2 =|\widehat{\varphi^j}(\mathsf{N}_\lambda^j\xi)|^2+\sum_{n=1}^{d}\sum_{\vec{\ell}=-\vec{r}_j}^{\vec{r}_j}|\widehat{\psi^{j,\vec{\ell}}}(S_{\vec{\ell}}\mathsf{B}_\lambda^j\mathsf{E}_n\xi)|^2, \quad a.e., \xi\in\mathbb{R}^d,\ j\geqslant J_0,\\
&\lim_{j\to\infty}\langle|\widehat{\varphi^j}(\mathsf{N}_\lambda^j\cdot)|^2,\widehat{h}\rangle =\langle 1,\widehat{h}\rangle, \quad \forall\widehat{h}\in\mathscr{D}(\mathbb{R}^2).
\end{aligned}$$

2.2 Auxiliary functions for the construction of smooth affine shear tight frames

Before we proceed to the construction of d-dimensional smooth affine shear tight frames, we briefly introduce some necessary auxiliary functions. We shall use a function $\nu\in C_c^\infty(\mathbb{R})$ (compactly supported C^∞ function) such that $\nu(x)=0$ for $x\leqslant -1$, $\nu(x)=1$ for $x\geqslant 1$, and $|\nu(x)|^2+|\nu(-x)|^2=1$ for all $x\in\mathbb{R}$. There are many choices of such functions. For example, let $f(x):=e^{-1/x^2}$ for $x>0$ and $f(x):=0$ for $x\leqslant 0$, and $g(x):=\int_{-1}^x f(1+t)f(1-t)dt$. Define

$$\nu(x):=\frac{g(x)}{\sqrt{|g(x)|^2+|g(-x)|^2}}, \quad x\in\mathbb{R}. \tag{2.3}$$

Then $\nu\in C_c^\infty(\mathbb{R})$ is a desired function. Using such a function ν, we now construct our building blocks $\boldsymbol{\alpha}_{\lambda,t,\rho},\boldsymbol{\beta}_{\lambda,t,\rho}$ of Meyer-type scaling and wavelet functions with $\lambda>1$, $0<t\leqslant 1$, and $0<\rho\leqslant\lambda^2$ as follows:

$$\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi):=\begin{cases}\nu(\frac{\xi-a}{\varepsilon_1}) & \text{if } \xi<a+\varepsilon_1,\\ 1 & \text{if } a+\varepsilon_1\leqslant\xi\leqslant b-\varepsilon_2,\\ \nu(\frac{-\xi+b}{\varepsilon_2}) & \text{if } \xi>b-\varepsilon_2,\end{cases} \quad \boldsymbol{\beta}_{\lambda,t,\rho}(\xi):=(|\boldsymbol{\alpha}_{\lambda,t,\rho}(\lambda^{-2}\xi)|^2-|\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)|^2)^{1/2}, \tag{2.4}$$

where $[a,b]=[-\lambda^{-2}(1-t/2)\rho\pi,\lambda^{-2}(1-t/2)\rho\pi]$ and $\varepsilon_1=\varepsilon_2=\lambda^{-2}t\rho\pi/2$. Then $\boldsymbol{\alpha}_{\lambda,t,\rho},\boldsymbol{\beta}_{\lambda,t,\rho}\in C_c^\infty(\mathbb{R})$. Moreover, $\operatorname{supp}\boldsymbol{\alpha}_{\lambda,t,\rho}=[-\lambda^{-2}\rho\pi,\lambda^{-2}\rho\pi]$ and $\operatorname{supp}\boldsymbol{\beta}_{\lambda,t,\rho}=[-\rho\pi,-\lambda^{-2}(1-t)\rho\pi]\cup[\lambda^{-2}(1-t)\rho\pi,\rho\pi]$. Furthermore, define a 2π-periodic function $\boldsymbol{\mu}_{\lambda,t,\rho}$ and $\boldsymbol{\upsilon}_{\lambda,t,\rho}$ as follows:

$$\boldsymbol{\mu}_{\lambda,t,\rho}(\xi):=\begin{cases}\frac{\boldsymbol{\alpha}_{\lambda,t,\rho}(\lambda^2\xi)}{\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)} & \text{if } |\xi|\leqslant\lambda^{-2}\rho\pi,\\ 0 & \text{if } \lambda^{-2}\rho\pi<|\xi|\leqslant\pi,\end{cases} \quad \boldsymbol{\upsilon}_{\lambda,t,\rho}(\xi):=\begin{cases}\frac{\boldsymbol{\beta}_{\lambda,t,\rho}(\lambda^2\xi)}{\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)} & \text{if } \lambda^{-4}(1-t)\rho\pi\leqslant|\xi|\leqslant\lambda^{-2}\rho\pi,\\ \mathbf{g}_{\lambda,t,\rho}(\xi) & \text{if } \xi\in[-\pi,\pi]\backslash\operatorname{supp}\boldsymbol{\beta}_{\lambda,t,\rho}(\lambda^2\cdot),\end{cases} \tag{2.5}$$

where $\mathbf{g}_{\lambda,t,\rho}$ is a function in $C^\infty(\mathbb{T})$ such that $\left[\frac{d^n}{d\xi^n}\mathbf{g}_{\lambda,t,\rho}(\xi)\right]\Big|_{\xi=\pm\lambda^{-2}\rho\pi}=\delta(n)$ for all $n\in\mathbb{N}_0$. The purpose of $\mathbf{g}_{\lambda,t,\rho}$ is to make the function $\boldsymbol{\upsilon}_{\lambda,t,\rho}$ smooth. We have the following result.

PROPOSITION 2.1. *Let $\lambda>1$, $0<t\leqslant 1$, and $0<\rho\leqslant\lambda^2$. Let $\boldsymbol{\alpha}_{\lambda,t,\rho}$, $\boldsymbol{\beta}_{\lambda,t,\rho}$, and $\boldsymbol{\mu}_{\lambda,t,\rho},\boldsymbol{\upsilon}_{\lambda,t,\rho}$ be defined as in* (2.4) *and* (2.5)*, respectively. Then $\boldsymbol{\alpha}_{\lambda,t,\rho},\boldsymbol{\beta}_{\lambda,t,\rho}\in C_c^\infty(\mathbb{R})$ and $\boldsymbol{\mu}_{\lambda,t,\rho},\boldsymbol{\upsilon}_{\lambda,t,\rho}\in C^\infty(\mathbb{T})$. Moreover, $|\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)|^2+|\boldsymbol{\beta}_{\lambda,t,\rho}(\xi)|^2=|\boldsymbol{\alpha}_{\lambda,t,\rho}(\lambda^{-2}\xi)|^2$, $\boldsymbol{\alpha}_{\lambda,t,\rho}(\lambda^2\xi)=\boldsymbol{\mu}_{\lambda,t,\rho}(\xi)\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)$, and $\boldsymbol{\beta}_{\lambda,t,\rho}(\lambda^2\xi)=\boldsymbol{\upsilon}_{\lambda,t,\rho}(\xi)\boldsymbol{\alpha}_{\lambda,t,\rho}(\xi)$ for $\xi\in\mathbb{R}$.*

The functions $\boldsymbol{\alpha}_{\lambda,t,\rho}$ and $\boldsymbol{\beta}_{\lambda,t,\rho}$ shall be used for the ξ_1-axis. We next define 'bump' function $\boldsymbol{\gamma}_\varepsilon$ for splitting pieces along the other axies. Roughly speaking, the core generator for our affine shear systems in the frequency domain looks like $\boldsymbol{\beta}_{\lambda,t,\rho}(\xi_1)\prod_{n=2}^d\boldsymbol{\gamma}_\varepsilon(\xi_n/\xi_1)$, which is a pyramid shape generator. Application of parabolic scaling, shear, and translation operations to such a generator induces our affine shear systems. Further technical treatments are then applied on such systems to achieve tightness; see next subsections for details.

Let ε be a constant such that $0<\varepsilon\leqslant 1/2$. For $\lambda>1$, define $\ell_\lambda:=\lfloor\lambda-(1/2+\varepsilon)\rfloor+1=\lfloor\lambda+(1/2-\varepsilon)\rfloor$. Define a function $\boldsymbol{\gamma}_\varepsilon$ to be

$$\boldsymbol{\gamma}_\varepsilon(x)=\begin{cases}1 & \text{if } |x|\leqslant 1/2-\varepsilon,\\ \nu(\frac{-|x|+1/2}{\varepsilon}) & \text{if } 1/2-\varepsilon\leqslant|x|\leqslant 1/2+\varepsilon,\\ 0 & \text{otherwise.}\end{cases} \tag{2.6}$$

Then it is easy to check that $\gamma_\varepsilon \in C_c^\infty(\mathbb{R})$ and $\sum_{\ell\in\mathbb{Z}} |\boldsymbol{\gamma}_\varepsilon(\cdot+\ell)|^2 = 1$. Moreover,

$$\sum_{\ell=-\ell_\lambda}^{\ell_\lambda} |\boldsymbol{\gamma}_\varepsilon(\lambda x+\ell)|^2 = 1 \quad \forall |x| \leqslant \frac{\ell_\lambda + 1/2 - \varepsilon}{\lambda}. \tag{2.7}$$

We next define $\boldsymbol{\Gamma}_j$, which will be used for normalization of frequency splitting along the shear directions. We have the following result.

PROPOSITION 2.2. *Let $j \in \mathbb{N}_0$. Define $\boldsymbol{\gamma}(\xi) := \prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\xi_n/\xi_1)$ and*

$$\boldsymbol{\gamma}^{j,\vec{\ell}}(\xi) := \boldsymbol{\gamma}(S_{\vec{\ell}}\mathsf{B}_\lambda^j\xi) = \prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\lambda^j\xi_n/\xi_1 + \ell_n) \quad \text{and} \quad \boldsymbol{\Gamma}_j(\xi) := \sum_{n=1}^d \sum_{\vec{\ell}=-\vec{\ell}_{\lambda^j}}^{\vec{\ell}_{\lambda^j}} \boldsymbol{\gamma}^{j,\vec{\ell}}(\mathsf{E}_n\xi), \tag{2.8}$$

where $\vec{\ell} := (\ell_2,\dots,\ell_d) \in \mathbb{Z}^{d-1}$ and $\vec{\ell}_{\lambda^j} := (\ell_{\lambda^j},\dots,\ell_{\lambda^j}) \in \mathbb{Z}^{d-1}$. Then $\boldsymbol{\Gamma}_j \in C^\infty(\mathbb{R}^d\backslash\{0\})$ has the following properties.

(i) $0 < \boldsymbol{\Gamma}_j(\xi) \leqslant 2$, $\boldsymbol{\Gamma}_j(\mathsf{E}_n\xi) = \boldsymbol{\Gamma}_j(\xi)$ *for all* $n = 1,\dots,d$, *and* $\boldsymbol{\Gamma}_j(t\xi) = \boldsymbol{\Gamma}_j(\xi)$ *for all* $t \neq 0$ *and* $\xi \neq 0$.

(ii) $\boldsymbol{\Gamma}_j$ *satisfies* $\boldsymbol{\Gamma}_j(\xi) \equiv 1$ *for* $\xi \in \Big\{\xi = (\xi_1,\dots,\xi_d) \in \mathbb{R}^d : \max\{|\xi_m/\xi_n| : m \neq n; m,n = 1,\dots,d\} \leqslant \frac{\lambda^j}{\ell_{\lambda^j}+1/2+\varepsilon}\Big\}$.

2.3 Construction of smooth affine shear tight frames for $L_2(\mathbb{R}^d)$

Let us next discuss the quasi-stationary construction of smooth affine shear tight frames, in which case, $\varphi^j \equiv \varphi$ for all j. One can also construct non-stationary smooth affine shear tight frames similar to those in [13]. For simplicity, we only consider the quasi-stationary construction. The idea is to use the tensor product of functions in 1D to obtain rectangular bands for different scales, and then a frequency splitting using $\boldsymbol{\gamma}^{j,\vec{\ell}}$ is applied to produce generators with respect to different shears. More precisely, let $\lambda > 1$, $0 < t \leqslant 1$, and $0 < \rho \leqslant 1$. Consider $\widehat{\varphi}(\xi) := [\otimes\boldsymbol{\alpha}_{\lambda,t,\rho}](\xi) = \prod_{n=1}^d \boldsymbol{\alpha}_{\lambda,t,\rho}(\xi_n)$, $\xi = (\xi_1,\dots,\xi_d) \in \mathbb{R}^d$ and define

$$\boldsymbol{\omega}_{\lambda,t,\rho}(\xi) := \sqrt{|\widehat{\varphi}(\lambda^{-2}\xi)|^2 - |\widehat{\varphi}(\xi)|^2}, \quad \xi \in \mathbb{R}^d. \tag{2.9}$$

Then $\boldsymbol{\omega}_{\lambda,t,\rho} \in C^\infty(\mathbb{R}^d)$. In view of the construction of $\boldsymbol{\alpha}_{\lambda,t,\rho}$, the refinable structure is clear. We have $\widehat{\varphi}(\lambda^2\xi) = \widehat{a}(\xi)\widehat{\varphi}(\xi)$, $\xi \in \mathbb{R}^d$ with $\widehat{a} = \otimes^d\boldsymbol{\mu}_{\lambda,t,\rho}$ being the tensor product of the 1D mask $\boldsymbol{\mu}_{\lambda,t,\rho}$ given in (2.5). Moreover, we have $\boldsymbol{\omega}(\lambda^2\xi) = \widehat{b}(\xi)\widehat{\varphi}(\xi)$ with $\widehat{b} \in C^\infty(\mathbb{T}^d)$ being given by $\widehat{b}(\xi) = (\mathbf{g}(\xi) - |\widehat{a}(\xi)|^2)^{1/2}$ for any smooth function $\mathbf{g} \in C^\infty(\mathbb{T}^d)$ such that $\mathbf{g} = 1$ on the support of $\widehat{\varphi}$. Note that for simplicity of presentation, we omit the dependency of $\varphi, \psi^{j,\vec{\ell}}, \boldsymbol{\gamma}^{j,\vec{\ell}}, \widehat{a}, \widehat{b}, \boldsymbol{\Gamma}_j$, etc., on the parameters $\lambda, t, \rho, \varepsilon$.

Since $0 < \boldsymbol{\Gamma}_j \leqslant 2$ and $\boldsymbol{\Gamma}_j$ is in $C^\infty(\mathbb{R}^d\backslash\{0\})$, we have that $\sqrt{\boldsymbol{\Gamma}_j}$ is infinitely differentiable for all $\xi \in \mathbb{R}^d\backslash\{0\}$. Let $\mathsf{A}_\lambda, \mathsf{B}_\lambda, \mathsf{M}_\lambda, \mathsf{N}_\lambda, \mathsf{D}_\lambda$ with $\lambda > 1$ be defined as before. Let $\Psi_j := \{\psi^{j,\vec{\ell}}(S^{-\vec{\ell}}\cdot) : |\vec{\ell}| \leqslant \vec{\ell}_{\lambda^j}\}$ with

$$\begin{aligned}\widehat{\psi^{j,\vec{\ell}}}(\xi) &:= \boldsymbol{\omega}_{\lambda,t,\rho}(\mathsf{D}_\lambda^{-j}S_{-\vec{\ell}}\xi)\frac{\boldsymbol{\gamma}^{j,\vec{\ell}}((S_{\vec{\ell}}\mathsf{B}_\lambda^j)^{-1}\xi)}{\sqrt{\boldsymbol{\Gamma}_j((S_{\vec{\ell}}\mathsf{B}_\lambda^j)^{-1}\xi)}} \\ &= \boldsymbol{\omega}_{\lambda,t,\rho}(\xi_1, \lambda^{-j}(-\xi_1\ell_2+\xi_2),\dots,\lambda^{-j}(-\xi_1\ell_d\ell_d+\xi_d))\frac{\prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\xi_n/\xi_1)}{\sqrt{\boldsymbol{\Gamma}_j((S_{\vec{\ell}}\mathsf{B}_\lambda^j)^{-1}\xi)}}, \quad \xi \in \mathbb{R}^d\backslash\{0\},\end{aligned} \tag{2.10}$$

and $\widehat{\psi^{j,\vec{\ell}}}(0) := 0$, which gives $\widehat{\psi^{j,\vec{\ell}}}(S_{\vec{\ell}}\mathsf{B}_\lambda^j\xi) = \boldsymbol{\omega}_{\lambda,t,\rho}(\mathsf{N}_\lambda^j\xi)\frac{\boldsymbol{\gamma}^{j,\vec{\ell}}(\xi)}{\sqrt{\boldsymbol{\Gamma}_j(\xi)}}$. By the properties of $\boldsymbol{\Gamma}_j$ and that the support of $\boldsymbol{\omega}_{\lambda,t,\rho}$ is away from the origin, we see that $\psi^{j,\vec{\ell}}$ are functions in $C^\infty(\mathbb{R}^d)$. We have the following (quasi-stationary) *d-dimensional affine shear system*:

$$\begin{aligned}\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty) :=& \{\varphi_{\mathsf{M}_\lambda^J;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d\} \cup \{h_{\mathsf{A}_\lambda^j\mathsf{E}_n;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d, n = 1,\dots,d, h \in \Psi_j\}_{j=J}^\infty \\ =& \{\varphi_{\mathsf{M}_\lambda^J;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d\} \cup \{\psi^{j,\vec{\ell}}_{S^{-\vec{\ell}}\mathsf{A}_\lambda^j\mathsf{E}_n;\mathsf{k}} : \mathsf{k} \in \mathbb{Z}^d, n = 1,\dots,d, |\vec{\ell}| \leqslant \vec{\ell}_{\lambda^j}\}_{j=J}^\infty.\end{aligned} \tag{2.11}$$

At first glance, such a system does not have shear structure at all due to that the function $\boldsymbol{\omega}_{\lambda,t,\rho}$ is not shear-invariant. However, we show that such a system do have certain affine and shear structure in the sense that a subsystem of this system is from shear and dilation of one single generator.

THEOREM 2.2. *Let $\lambda > 1$, $0 < t \leqslant 1$, and $0 < \rho \leqslant 1$. Let $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ be defined as in (2.11) with $\widehat{\varphi} = \otimes^d \boldsymbol{\alpha}_{\lambda,t,\rho}$ and $\psi^{j,\ell}$ being given by (2.10). Then $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ is an affine shear tight frame for $L_2(\mathbb{R}^d)$ for all $J \geqslant 0$. All elements in $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ are functions in $C^\infty(\mathbb{R}^d)$. Moreover, we have $\{\psi(S^{-\vec{\ell}}\cdot) : |\vec{\ell}| \leqslant \vec{r}_j\} \subseteq \Psi_j$, $j \geqslant J$, where $\vec{r}_j := (r_j,\dots,r_j) \in \mathbb{Z}^{d-1}$ with $r_j := \lfloor \lambda^{j-2}(1-t)\rho - (1/2+\varepsilon)\rfloor$ and $\widehat{\psi}(\xi) := \boldsymbol{\beta}_{\lambda,t,\rho}(\xi_1)\prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\xi_n/\xi_1)$, $\xi \in \mathbb{R}^d$. In other words, $\{\psi_{S^{-\vec{\ell}}\mathsf{A}_\lambda^j\mathsf{E}_n;\mathsf{k}} : \mathsf{k}\in\mathbb{Z}^d, n = 1,\dots,d, |\vec{\ell}| \leqslant \vec{r}_j\}_{j=J}^\infty \subseteq \mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$.*

2.4 Smooth affine wavelet tight frames and their subsampling systems

We now discuss the connection of affine shear tight frames and affine wavelet tight frames. To this end, we first introduce affine wavelet systems. Considering the system $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ defined in (2.11), we have

$$\widehat{\varphi} = \otimes^d \boldsymbol{\alpha}_{\lambda,t,\rho} \quad \text{and} \quad \widehat{\psi^{j,\vec{\ell}}}(\xi) = \boldsymbol{\omega}_{\lambda,t,\rho}(\mathsf{D}_\lambda^{-j} S_{-\vec{\ell}}\xi)\frac{\prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\xi_n/\xi_1)}{\sqrt{\boldsymbol{\Gamma}_j((S_{\vec{\ell}}\mathsf{B}_\lambda^j)^{-1}\xi)}}, \quad \xi\in\mathbb{R}^d,$$

which gives

$$\widehat{(\psi^{j,\vec{\ell}})}_{S_{\vec{\ell}}\mathsf{B}_\lambda^j;0,\mathsf{k}}(\xi) = \lambda^{-(d+1)j/2}\boldsymbol{\omega}_{\lambda,t,\rho}(\mathsf{N}_\lambda^j\xi)\frac{\prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\lambda^j\xi_n/\xi_1+\ell_n)}{\sqrt{\boldsymbol{\Gamma}_j(\xi)}} e^{-i\mathsf{k}\cdot S_{\vec{\ell}}\mathsf{B}_\lambda^j\xi}.$$

Now define a new set

$$\mathring{\Psi}_j := \{\mathring{\psi}^{j,\vec{\ell}} : |\vec{\ell}| \leqslant \vec{\ell}_{\lambda^j}\} \text{ with } \widehat{\mathring{\psi}^{j,\ell}}(\xi) := \boldsymbol{\omega}_{\lambda,t,\rho}(\xi)\frac{\prod_{n=2}^d \boldsymbol{\gamma}_\varepsilon(\lambda^j\xi_n/\xi_1+\ell_n)}{\sqrt{\boldsymbol{\Gamma}_j(\xi)}},\ \xi\in\mathbb{R}^d. \tag{2.12}$$

Then

$$\widehat{(\mathring{\psi}^{j,\ell})}_{\mathsf{N}^j;0,\mathsf{k}}(\xi) = \lambda^{-dj}\boldsymbol{\omega}_{\lambda,t,\rho}(\mathsf{N}_\lambda^j\xi)\frac{\boldsymbol{\gamma}^{j,\vec{\ell}}(\xi)}{\sqrt{\boldsymbol{\Gamma}_j(\xi)}} e^{-i\mathsf{k}\cdot\mathsf{N}_\lambda^j\xi}.$$

We use a fixed dilation matrix M_λ for all generators φ and $\mathring{\psi}^{j,\ell}$. The (quasi-stationary) *d-dimensional affine wavelet system* is then defined to be

$$\begin{aligned}\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty) =& \{\varphi_{\mathsf{M}_\lambda^J;\mathsf{k}} : \mathsf{k}\in\mathbb{Z}^d\} \cup \{h_{\mathsf{M}_\lambda^j\mathsf{E}_n;\mathsf{k}} : \mathsf{k}\in\mathbb{Z}^d, n=1,\dots,d, h\in\mathring{\Psi}_j\}_{j=J}^\infty \\ =& \{\varphi_{\mathsf{M}_\lambda^J;\mathsf{k}} : \mathsf{k}\in\mathbb{Z}^d\} \cup \{\mathring{\psi}^{j,\vec{\ell}}_{\mathsf{M}_\lambda^j\mathsf{E}_n;\mathsf{k}} : \mathsf{k}\in\mathbb{Z}^d, n=1,\dots,d, |\vec{\ell}| \leqslant \vec{\ell}_{\lambda^j}\}_{j=J}^\infty.\end{aligned} \tag{2.13}$$

The connections of (2.13) and (2.11) is given by the following result.

THEOREM 2.3. *Retaining all the conditions for $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ in Theorem 2.2. Let $\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$ be defined as in (2.13). Then*

(i) *$\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$ is an affine wavelet tight frame for $L_2(\mathbb{R}^d)$ for all $J \geqslant 0$; that is, $\{\varphi\}\cup\{\mathring{\Psi}_j\}_{j=J}^\infty \subseteq L_2(\mathbb{R}^d)$ and*

$$\|f\|_2^2 = \sum_{\mathsf{k}\in\mathbb{Z}^d} |\langle f, \varphi_{\mathsf{M}_\lambda^J;\mathsf{k}}\rangle|^2 + \sum_{j=J}^\infty\sum_{n=1}^d\sum_{h\in\mathring{\Psi}_j}\sum_{\mathsf{k}\in\mathbb{Z}^d} |\langle f, h_{\mathsf{M}_\lambda^j\mathsf{E}_n;\mathsf{k}}\rangle|^2, \quad \forall f\in L_2(\mathbb{R}^d). \tag{2.14}$$

(ii) *All elements $\widehat{h}$ for $h\in\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$ are compactly supported functions in $C^\infty(\mathbb{R}^d)$.*

(iii) *Both $\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$ and $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ in Theorem 2.2 are connected to each other by the following relations:*

$$\psi^{j,\vec{\ell}}_{S^{-\vec{\ell}}\mathsf{A}_\lambda^j\mathsf{E}_n;\mathsf{k}} = \lambda^{(d-1)j/2}\mathring{\psi}^{j,\vec{\ell}}_{\mathsf{M}_\lambda^j\mathsf{E}_n;\mathsf{D}_\lambda^j S^{\vec{\ell}}\mathsf{k}}, \quad n=1,\dots,d. \tag{2.15}$$

When λ is an integer, we have $\mathsf{D}_\lambda^j S^{\vec{\ell}}\mathbb{Z}^d \subseteq \mathbb{Z}^d$. Equation (2.15) shows that when λ is an integer, the affine shear tight frame $\mathsf{AS}_J(\varphi;\{\Psi_j\}_{j=J}^\infty)$ is indeed a subsystem of the affine wavelet tight frame $\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$ through subsampling.

2.5 The underlying filter banks of smooth affine shear tight frames

Let us next study the underlying filter bank structure of the affine shear tight frames associated with its affine wavelet tight frame. For a filter $u=\{u(\mathsf{k})\}_{\mathsf{k}\in\mathbb{Z}^d}$; that is $u:\mathbb{Z}^d\to\mathbb{C}$, we define its *Fourier series* $\widehat{u}:\mathbb{T}^d\to\mathbb{C}$ to be $\widehat{u}(\xi)=\sum_{\mathsf{k}\in\mathbb{Z}^d}u(\mathsf{k})e^{-i\mathsf{k}\cdot\xi}$, $\xi\in\mathbb{T}^d$. Obviously, $\widehat{u}$ are $2\pi\mathbb{Z}^d$-periodic. For the (quasi-stationary) $\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$, we can define a sequence of filter banks. In this case, the low-pass filter $\widehat{a}:\mathbb{T}^d\to\mathbb{C}$ for φ is fixed as follows

$$\widehat{a}(\xi)=[\otimes^d\boldsymbol{\mu}_{\lambda,t,\rho}](\xi),\quad \xi\in\mathbb{T}^d \tag{2.16}$$

with $\boldsymbol{\mu}_{\lambda,t,\rho}$ being given in (2.5). Note that $\operatorname{supp}\widehat{\mathring{\psi}^{j,\vec{\ell}}}(\mathsf{M}_\lambda\cdot)\subseteq\operatorname{supp}\widehat{\varphi}$. Define $2\pi\mathbb{Z}^d$-periodic functions $\widehat{b^{j,\vec{\ell}}}$ for $\widehat{\mathring{\psi}^{j,\vec{\ell}}}$, $j\geqslant 0$ as follows.

$$\widehat{b^{j,\vec{\ell}}}(\xi):=\widehat{b}(\xi)\frac{\boldsymbol{\gamma}^{j,\vec{\ell}}(\xi)}{\sqrt{\boldsymbol{\Gamma}_j(\xi)}},\quad |\ell|\leqslant\ell_{\lambda^j},\quad \xi\in\mathbb{T}^d, \tag{2.17}$$

where $\widehat{b}(\xi):=\sqrt{\mathbf{g}(\xi)-|\widehat{a}(\xi)|^2}$ for some smooth function $\mathbf{g}$ defined on $\mathbb{T}^d$ satisfying $\mathbf{g}\equiv 1$ on the support of $\widehat{\varphi}$. Then, we have the following result.

THEOREM 2.4. *Let $\lambda>1$ be an integer. Choose $0<\varepsilon\leqslant 1/2$, $0<t\leqslant 1$, and $0<\rho<1$. Let $\mathsf{WS}_J(\varphi;\{\mathring{\Psi}_j\}_{j=J}^\infty)$, $J\geqslant 0$ be defined as in (2.13) and let $a,b^{j,\ell}$ be defined as in (2.16) and (2.17), respectively. Then there exists $\mathbf{g}\in C^\infty(\mathbb{T}^d)$ such that $\widehat{a},\widehat{b^{j,\vec{\ell}}}\in C^\infty(\mathbb{T}^d)$ for all $j\geqslant 0,|\vec{\ell}|\leqslant\vec{\ell}_{\lambda^j}$ and*

$$\widehat{\varphi}(\mathsf{M}_\lambda\xi)=\widehat{a}(\xi)\widehat{\varphi}(\xi)\quad\text{and}\quad\widehat{\mathring{\psi}^{j,\ell}}(\mathsf{M}_\lambda\xi)=\widehat{b^{j,\ell}}(\xi)\widehat{\varphi}(\xi) \tag{2.18}$$

for a.e. $\xi\in\mathbb{R}^d$. Moreover, for any $j\geqslant 0$, the set $\{a,b^{j,\vec{\ell}}(\mathsf{E}_n\cdot):|\ell|\leqslant\vec{\ell}_{\lambda^j},n=1,\dots,d\}$ is an affine wavelet filter bank having the perfect reconstruction property, i.e.,

$$\begin{aligned}
&|\widehat{a}(\xi)|^2+\sum_{n=1}^d\sum_{\vec{\ell}=-\vec{\ell}_{\lambda^j}}^{\vec{\ell}_{\lambda^j}}|\widehat{b^{j,\vec{\ell}}}(\mathsf{E}_n\xi)|^2=1,\quad \text{a.e. } \xi\in\sigma_\varphi,\\
&\overline{\widehat{a}(\xi)}\widehat{a}(\xi+2\pi\omega)+\sum_{n=1}^d\sum_{\vec{\ell}=-\vec{\ell}_{\lambda^j}}^{\vec{\ell}_{\lambda^j}}\overline{\widehat{b^{j,\vec{\ell}}}(\mathsf{E}_n\xi)}\widehat{b^{j,\ell}}(\mathsf{E}_n\xi+2\pi\omega)=0,
\end{aligned} \tag{2.19}$$

for a.e. $\xi\in\sigma_\varphi\cap(\sigma_\varphi-2\pi\omega)$ and for $\omega\in\Omega_{\mathsf{M}_\lambda}\backslash\{0\}$ with $\Omega_{\mathsf{M}_\lambda}=[\mathsf{M}_\lambda^{-1}\mathbb{Z}^d]\cap[0,1)^d$ and $\sigma_\varphi:=\{\xi\in\mathbb{R}^d:\sum_{\mathsf{k}\in\mathbb{Z}^d}|\widehat{\varphi}(\xi+2\pi\mathsf{k})|^2\neq 0\}$.

3. DIGITIZATION OF SMOOTH AFFINE SHEAR TIGHT FRAMES

In this section, we briefly discuss the numerical implementation of our smooth affine shear systems. Our implementation is based on the underlying filter bank of our quasi-stationary smooth affine shear tight frames: the affine wavelet filter banks in (2.19). We first construct digital affine shear filter banks induced from affine wavelet filter banks and then detail the implementation of the forward and backward digital affine shear transforms based on the digital affine shear filter banks.

3.1 Digital affine shear filter banks

For some parameters $c_0 > 0$ and $\epsilon_0 > 0$ satisfying $c_0 + \epsilon_0 \leqslant \pi/2$ (for downsampling by 2), we can define an inner function $\widehat{a} \in L_2(\mathbb{T}^d)$ (as a low-pass filter) by

$$\widehat{a}(\xi) := [\otimes^d \boldsymbol{\mu}_{[c_0,\epsilon_0]}](\xi) = \prod_{n=1} \boldsymbol{\mu}_{[c_0,\epsilon_0]}(\xi_n), \quad \xi = (\xi_1, \ldots, \xi_n) \in \mathbb{T}^d \tag{3.1}$$

with $\boldsymbol{\mu}_{[c,\epsilon]}$ being defined as

$$\boldsymbol{\mu}_{[c,\epsilon]}(t) = \begin{cases} \nu(\frac{t+c}{\epsilon}) & \text{if } t < -c+\epsilon, \\ 1 & \text{if } -c+\epsilon <= t <= c-\epsilon, \\ \nu(\frac{-t+c}{\epsilon}) & \text{if } t > c-\epsilon. \end{cases}$$

Similarly, we define an outer function $\widehat{a_1} \in L_2(\mathbb{R}^d)$ by $\widehat{a_1} := \otimes^d \boldsymbol{\mu}_{[c_1,\epsilon_1]}$, where $c_1 = \pi$ and ϵ_1 satisfies $c_1 + \epsilon_1 - (c_0 - \epsilon_0) \leqslant \pi$ for the purpose of downsampling at least by 2 for high-pass filter coefficients. Thanks to the property of ν, one can show that

$$\sum_{\mathsf{k} \in \mathbb{Z}^d} |\widehat{a_1}(\xi + 2\pi\mathsf{k})|^2 = 1 \qquad \forall \xi \in \mathbb{R}^d. \tag{3.2}$$

Now we can define an 'isotropic' function $b \in L_2(\mathbb{R}^d)$ by

$$\widehat{b}(\xi) := \begin{cases} \sqrt{|\widehat{a_1}(\xi)|^2 - |\widehat{a}(\xi)|^2} & \text{if } \xi \in \operatorname{supp} \widehat{a_1}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Note that both a_1 and b are not filters and both $\widehat{a_1}$ and $\widehat{b}$ are supported on $[-\pi - \epsilon_1, \pi + \epsilon_1]^d$.

Now, we apply the splitting technique to $\widehat{b}$ for the construction of high-pass filters $b^{j,\vec{\ell}}$. At scale $j \geqslant 0$ and a nonnegative integers k_j, define $\vec{r}_j := (2^{k_j}, \ldots, 2^{k_j}) \in \mathbb{Z}^{d-1}$. The number k_j controls the total number of shear directions at scale j. Similar to the definition of normalization function $\mathbf{\Gamma}_j$ in (2.8), we define

$$\mathbf{\Gamma}_{k_j}(\xi) = \sum_{n=1}^{d} \sum_{\vec{\ell}=-\vec{r}_j}^{r_j} |\boldsymbol{\gamma}^{k_j,\vec{\ell}}(\mathsf{E}_n \xi)|^2, \quad \xi \neq 0 \quad \text{and} \quad \mathbf{\Gamma}_{k_j}(0) := 0, \tag{3.4}$$

where $\boldsymbol{\gamma}^{k_j,\vec{\ell}}(\xi) := \prod_{n=2}^{d} \boldsymbol{\gamma}_\varepsilon(2^{k_j}\xi_n/\xi_1 + \ell_n)$ for some $0 \leqslant \varepsilon \leqslant \frac{1}{1+\epsilon_1} - 1/2$. To guarantee smoothness of boundary, we need to further split $\boldsymbol{\gamma}^{k_j,\vec{\ell}}(\xi)$ to positive part and negative part of ξ_1-axis. Define $\boldsymbol{\gamma}^{k_j,\vec{\ell},\pm}(\xi) := \boldsymbol{\gamma}^{k_j,\vec{\ell}}(\xi)\chi_{\{\pm\xi_1 > 0\}}$. Note that $\widehat{b}(\xi)\frac{\boldsymbol{\gamma}^{k_j,\vec{\ell},\pm}(\xi)}{\sqrt{\mathbf{\Gamma}_{k_j}(\xi)}}$ are not $2\pi\mathbb{Z}^d$-periodic functions. We define $b^{j,\vec{\ell},\pm}$ to be the $2\pi\mathbb{Z}^d$-periodization of $\widehat{b}(\xi)\frac{\boldsymbol{\gamma}^{k_j,\vec{\ell},\pm}(\xi))}{\sqrt{\mathbf{\Gamma}_{k_j}(\xi)}}$ as follows.

$$\widehat{b^{j,\vec{\ell},\pm}}(\xi) := \sum_{\mathsf{k} \in \mathbb{Z}^d} \widehat{b}(\xi + 2\pi\mathsf{k}) \frac{\boldsymbol{\gamma}^{k_j,\vec{\ell},\pm}(\xi + 2\pi\mathsf{k})}{\sqrt{\mathbf{\Gamma}_{k_j}(\xi + 2\pi\mathsf{k})}}, \quad \xi \in \mathbb{T}^d. \tag{3.5}$$

The total number of high-pass filters $b^{j,\vec{\ell},+}$ and $b^{j,\vec{\ell},-}$ at this scale j is $2(2^{k_j+1}+1)^{d-1}$. Each filter of $\widehat{b^{j,\vec{\ell},\pm}}$ is $2\pi\mathbb{Z}^d$-periodic function on $\mathbb{T}^d$.

Now, in view of (3.2) and (3.4), it is easy to prove the following result.

THEOREM 3.1. *Let $a, b^{j,\vec{\ell},\pm}$ be defined as in (3.1), (3.5), respectively. Then $\{a, b^{j,\vec{\ell},\pm}(\mathsf{E}_n\cdot) : n = 1, \ldots, d, |\vec{\ell}| \leqslant \vec{r}_j\}$ forms a perfect reconstruction (PR) filter bank satisfying:*

$$\begin{aligned} &|\widehat{a}(\xi)|^2 + \sum_{n=1}^{d} \sum_{\vec{\ell}=-\vec{r}_j}^{\vec{r}_j} \big(|\widehat{b^{j,\vec{\ell},+}}(\mathsf{E}_n\xi)|^2 + |\widehat{b^{j,\vec{\ell},-}}(\mathsf{E}_n\xi)|^2\big) = 1, \\ &\overline{\widehat{a}(\xi)}\widehat{a}(\xi + 2\pi\omega) = 0, \quad \overline{\widehat{b^{j,\vec{\ell},\pm}}(\mathsf{E}_n\xi)}\widehat{b^{j,\vec{\ell},\pm}}(\mathsf{E}_n(\xi + 2\pi\omega_1)) = 0, \quad n = 1, \ldots, d, \quad \xi \in \mathbb{T}^d, \end{aligned} \tag{3.6}$$

for all $\omega \in [\frac{1}{2}\mathbb{Z}^d] \cap [0,1)^d \backslash \{0\}$ *and all* $\omega_1 \in [\mathrm{diag}(\frac{1}{2}, 2^{-k_j}\mathsf{I}_{d-1})\mathbb{Z}^d] \cap [0,1)^d \backslash \{0\}$.

Given a sequence of nonnegative integers $k_j : j = 0, \dots, J-1$ for some fixed integer $J \geqslant 0$ with respect to the finest scale. Let $\mathsf{M} := 2\mathsf{I}_d$ and $\mathsf{A}_{j,n} := \mathsf{E}_n \,\mathrm{diag}(2, 2^{k_j}\mathsf{I}_{d-1})$ for $n = 1, \dots, d$. We can then obtain a sequence of PR filter banks

$$\mathcal{B}_j := \{a \downarrow \mathsf{M}, b^{j,\vec{\ell},\pm}(\mathsf{E}_n \cdot) \downarrow \mathsf{A}_{j,n} : |\vec{\ell}| \leqslant \vec{r}_j, n = 1, \dots, d\} \tag{3.7}$$

for $j = 0, \dots, J-1$. Here M in $a \downarrow \mathsf{M}$ indicates downsampling matrix for filtered coefficients with respect to the low-pass filter a and $\mathsf{A}_{j,n}$ in $b^{j,\vec{\ell},\pm}(\mathsf{E}_n \cdot) \downarrow \mathsf{A}_{j,n}$ indicates downsampling matrix for filtered coefficients with respect to the high-pass filter $b^{j,\vec{\ell}}(\mathsf{E}_n \cdot)$. We shall call such a sequence of PR filter banks as a sequence of (quasi-stationary) *d-dimensional digital affine shear filter banks* and denote it as $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^{J-1})$. Note that the total number of high-pass filters in $\mathcal{B}_j$ is $2d(2^{k_j+1}+1)^{d-1}$.

3.2 Digital affine shear transforms

We next discuss the implementation of the forward transform (decomposition) and backward transform (reconstruction) of our digital affine shear filter banks. We shall implement our transforms based on discrete Fourier transform (DFT) for fast digitization. We first discuss three main operations for a filter bank decomposition and reconstruction: convolution, downsampling, and upsampling.

Without loss of generality and for the simplicity of presentation, we shall assume our data live on the dyadic grids $\Lambda(K)$ for $K = (K_1, \dots, K_d) \in \mathbb{N}_0^d = (\mathbb{N} \cup \{0\})^d$ define by

$$\Lambda(K) = \Lambda(K_1, \dots, K_d) := ([0, \dots, 2^{K_1} - 1] \times \cdots \times [0, \dots, 2^{K_d} - 1]) \cap \mathbb{Z}^d.$$

and

$$\widehat{\Lambda}(K) = \widehat{\Lambda}(K_1, \dots, K_d) := \frac{2\pi}{2^{K_1+\cdots+K_d}} \big(([-2^{K_1-1}, \dots, 2^{K_1-1} - 1] \times \cdots \times [-2^{K_d-1}, \dots, 2^{K_d-1} - 1]) \cap \mathbb{Z}^d\big).$$

The (centred) *discrete Fourier transform* (DFT) $\mathscr{F}$ maps a time-domain data $v : \Lambda(K) \to \mathbb{C}$ to a $2\pi\mathbb{Z}^d$-periodic frequency domain data $\widehat{v}|_{\widehat{\Lambda}(K)}$, which is defined to be

$$\widehat{v}(\mathsf{k}) = [\mathscr{F}v](\mathsf{k}) := \sum_{\mathsf{n} \in \Lambda(K)} v(\mathsf{n}) e^{-i\mathsf{n}\cdot\mathsf{k}}, \quad \mathsf{k} \in \widehat{\Lambda}(K).$$

The inverse (centred) DFT is given by

$$[\mathscr{F}^{-1}\widehat{v}](\mathsf{n}) = \frac{1}{2^{K_1+\cdots+K_d}} \sum_{\mathsf{k} \in \widehat{\Lambda}(K)} v(\mathsf{n}) e^{i\mathsf{n}\cdot\mathsf{k}}, \quad \mathsf{n} \in \Lambda(K).$$

The centred DFT and its inverse can be implemented by `fftn`, `ifftn`, and `fftshift` in MATLAB.

Given a filter u defined by its Fourier series $\widehat{u} : \mathbb{T}^d \to \mathbb{C}$, and a data $v : \Lambda(K) \to \mathbb{C}$, the *circular convolution* $v \circledast u : \Lambda(K) \to \mathbb{C}$ is given by

$$v \circledast u := \mathscr{F}^{-1}\big[\mathscr{F}(v) \cdot \widehat{u}|_{\widehat{\Lambda}(K)}\big],$$

where $\widehat{u}|_{\widehat{\Lambda}(K)}$ is the sampling of $\widehat{u}$ on the lattice $\widehat{\Lambda}(K)$. We shall omit such dependence and simply write $v \circledast u = \mathscr{F}^{-1}[\widehat{v} \cdot \widehat{u}]$ since it can be easily told from the expression. For a downsampling matrix $\mathsf{A} := \mathrm{diag}(2^{m_1}, \dots, 2^{m_d})$ for $\mathsf{m} := (m_1, \dots, m_d) \in \mathbb{N}_0^d$, the *downsampling operation* $v \downarrow \mathsf{A} : \Lambda(K - \mathsf{m}) \to \mathbb{C}$ and *upsampling operation* $v \uparrow \mathsf{A} : \Lambda(K + \mathsf{m}) \to \mathbb{C}$ is defined by

$$[v \downarrow \mathsf{A}](\mathsf{k}) = v(\mathsf{A}\mathsf{k}) \text{ for } \mathsf{k} \in \Lambda(K - \mathsf{m}) \quad \text{and} \quad [v \uparrow \mathsf{A}](\mathsf{k}) = \begin{cases} v(\mathsf{A}^{-1}\mathsf{k}) & \text{if } \mathsf{A}^{-1}\mathsf{k} \in \Lambda(K), \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to show that $\mathscr{F}(v \downarrow \mathsf{A}) = \Big(\frac{1}{|\det \mathsf{A}|} \sum_{\omega \in \Omega_\mathsf{A}} \widehat{v}(\mathsf{A}^{-\mathsf{T}}\xi + 2\pi\omega)\Big)\Big|_{\widehat{\Lambda}(K-\mathsf{m})}$, where $\Omega_\mathsf{A} := [\mathsf{A}^{-\mathsf{T}}\mathbb{Z}^d] \cap [0,1)^d$. Note that if $\widehat{v}(\cdot)\widehat{v}(\cdot + 2\pi\omega) = 0$ for all $\omega \in \Omega_\mathsf{A}$, then $\mathscr{F}(v \downarrow \mathsf{A})$ can be implemented efficiently by lattice modulation

$[\mathscr{F}(v \downarrow \mathsf{A})](\mathsf{k}) := \widehat{v}(\mathsf{A}^{-\mathsf{T}}\mathsf{k} + 2\pi\omega_{\mathsf{k}})$ for those k such that $\widehat{v}(\mathsf{A}^{-\mathsf{T}}\mathsf{k} + 2\pi\omega_{\mathsf{k}}) \neq 0$ for some $\omega_{\mathsf{k}} \in \Omega_{\mathsf{A}}$. For the upsampling operation, we have $\mathscr{F}(v \uparrow \mathsf{A}) = \widehat{v}(\mathsf{A}^{\mathsf{T}}\xi)\big|_{\widehat{\Lambda}(K+\mathsf{m})}$, which can be obtained by the periodic extension of $\mathscr{F}(v)$ in practice. The transition operator $\mathcal{T}_{u,\mathsf{A}}v$ and subdivision operator $\mathcal{S}_{u,\mathsf{A}}v$ combine the operations of circular convolution and down(up)-sampling together, which are defined to be

$$\mathcal{T}_{u,\mathsf{A}}v := (v \circledast u^\star) \downarrow \mathsf{A} \quad \text{and} \quad \mathcal{S}_{u,\mathsf{A}}v := |\det \mathsf{A}|[(v \uparrow \mathsf{A}) \circledast u],$$

where $u^\star$ is defined by $\widehat{u^\star}(\xi) = \overline{\widehat{u}(\xi)}$. It is easy to show that

$$\mathscr{F}(\mathcal{T}_{u,\mathsf{A}}v) = \left(\frac{1}{|\det \mathsf{A}|}\sum_{\omega\in\Omega_{\mathsf{A}}}[\widehat{v}\cdot\overline{\widehat{u}}](\mathsf{A}^{-\mathsf{T}}\xi + 2\pi\omega)\right)\Bigg|_{\widehat{\Lambda}(K-\mathsf{m})} \quad \text{and} \quad \mathscr{F}(\mathcal{S}_{u,\mathsf{A}}v) = \left(|\det \mathsf{A}|\widehat{v}(\mathsf{A}^{\mathsf{T}}\xi)\widehat{u}(\xi)\right)\Big|_{\widehat{\Lambda}(K+\mathsf{m})}.$$

Consequently, $\mathscr{F}(\mathcal{S}_{u,\mathsf{A}}\mathcal{T}_{u,\mathsf{A}}v) = \left(\sum_{\omega\in\Omega_{\mathsf{A}}}\widehat{v}(\xi+2\pi\omega)\cdot\overline{\widehat{u}(\xi+2\pi\omega)}\widehat{u}(\xi)\right)\Big|_{\widehat{\Lambda}(K)}$. Due to (3.6), it is easy to see the PR property of decomposition and reconstruction using (3.7):

$$\begin{aligned}
\mathscr{F}\left(\sum_{(b\downarrow\mathsf{A})\in\mathcal{B}_j}\mathcal{S}_{b,\mathsf{A}}\mathcal{T}_{b,\mathsf{A}}v\right) &= \sum_{(b\downarrow\mathsf{A})\in\mathcal{B}_j}\mathscr{F}\left(\mathcal{S}_{b,\mathsf{A}}\mathcal{T}_{b,\mathsf{A}}v\right) = \sum_{(b\downarrow\mathsf{A})\in\mathcal{B}_j}\left(\sum_{\omega\in\Omega_{\mathsf{A}}}\widehat{v}(\xi+2\pi\omega)\cdot\overline{\widehat{b}(\xi+2\pi\omega)}\widehat{b}(\xi)\right)\Bigg|_{\widehat{\Lambda}(K)} \\
&= \sum_{(b\downarrow\mathsf{A})\in\mathcal{B}_j}\left(\widehat{v}(\xi)\cdot|\widehat{b}(\xi)|^2\right)\Bigg|_{\widehat{\Lambda}(K)} = \left(\widehat{v}(\xi)\sum_{(b\downarrow\mathsf{A})\in\mathcal{B}_j}|\widehat{b}(\xi)|^2\right)\Bigg|_{\widehat{\Lambda}(K)} \\
&= \mathscr{F}(v).
\end{aligned}$$

We are now ready to present the forward and backward transforms of a digital affine shear filter bank. We first present the forward transform in Algorithm 1.

Algorithm 1. Forward Digital Affine Shear Transform (Decomposition)

(a) **Input:** d-dimensional real-value data v_0 on $\Lambda(K)$ for some $K = (K_1, \ldots, K_d) \in \mathbb{N}_0^d$ and a digital affine shear filter bank $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^{J-1})$ defined as in (3.7) with $J \leqslant \min\{K_1-1, \ldots, K_d-1\}$ and $k_0, \ldots, k_{J-1}$ of nonnegative integers determining the number of shear directions in $\mathcal{B}_j := \{a \downarrow \mathsf{M}, b^{j,\vec{\ell},\pm}(\mathsf{E}_n\cdot) \downarrow \mathsf{A}_{j,n} : |\vec{\ell}| \leqslant \vec{r}_j, n = 1, \ldots, d\}$ with $\vec{r}_j := (2^{k_j}, \ldots, 2^{k_j}) \in \mathbb{Z}^{d-1}$, $\mathsf{A}_{j,n} := \mathsf{E}_n \operatorname{diag}(2, 2^{k_j}\mathsf{I}_{d-1})$, and $a, b^{j,\vec{\ell},\pm}$ being defined as in (3.1) and (3.5), respectively.

(b) **Output:** Digital affine shear coefficients: $c^J \cup \{c^{j,\vec{\ell},n,+} : n = 1, \ldots, d, |\vec{\ell}| \leqslant \vec{r}_j, j = 0, \ldots, J-1\}$

(c) **Main steps:**

1: Initialization: $\widehat{v} \leftarrow \mathscr{F}(v_0)$ and $j \leftarrow 0$.
2: **while** $j < J$ **do**
3: Low-pass filtering $\mathcal{T}_{a,\mathsf{M}}v$: Initialize zero matrix $\widehat{u}$ on $\widehat{\Lambda}(K_1 - 1, \ldots, K_d - 1)$. Compute $\widehat{u}$ by $\widehat{u}(\mathsf{k}) \leftarrow \widehat{v}(\mathsf{M}^{-\mathsf{T}}\mathsf{k}) \cdot \widehat{a}(\mathsf{M}^{-\mathsf{T}}\mathsf{k})$ for $\mathsf{M}^{-\mathsf{T}}\mathsf{k} \in \operatorname{supp}(\widehat{a})$.
4: **for** each $b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot) \downarrow \mathsf{A}_{j,n} \in \mathcal{B}_j$ **do**
5: $b \leftarrow b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot)$ and $\mathsf{A} \leftarrow \mathsf{A}_{j,n}$. Initialize a zero matrix c on the lattice $\Lambda(K_1 - k_j, \ldots, K_{n-1} - k_j, K_n - 1, K_{n+1} - k_j, \ldots, K_d - k_j)$.
6: Compute $\mathscr{F}(c)$ by $\widehat{c}(\mathsf{k}) \leftarrow \widehat{v}(\mathsf{A}^{\mathsf{T}}\mathsf{k} + 2\pi\omega_{\mathsf{k}}) \cdot \widehat{b}(\mathsf{A}^{\mathsf{T}}\mathsf{k} + 2\pi\omega_{\mathsf{k}})$ for $\mathsf{A}^{\mathsf{T}}\mathsf{k} + 2\pi\omega_{\mathsf{k}} \in \operatorname{supp}(\widehat{b})$.
7: High-pass filtering $\mathcal{T}_{b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot),\mathsf{A}^{j,n}}v$: $c^{j,\vec{\ell},n,+} \leftarrow \mathscr{F}^{-1}(\widehat{c})$.
8: **end for**
9: $j \leftarrow j + 1$, $(K_1, \ldots, K_d) \leftarrow (K_1 - 1, \ldots, K_d - 1)$, and $\widehat{v} \leftarrow \widehat{u}$.
10: **end while**
11: $c^J \leftarrow \mathscr{F}^{-1}(\widehat{v})$.

We remark that filters in $\mathcal{B}_j$ can be precomputed and stored before doing decomposition and reconstruction. Since by our design the $\operatorname{supp}(\widehat{b})$ can be precomputed for each $b \in \mathcal{B}_j$, we only need to store data of filters

on its support. Moreover, we only need to compute $c^{j,\vec{\ell},n,+}$ due to the fact that $\widehat{b^{j,\vec{\ell},-}}(\xi) = \widehat{b^{j,\vec{\ell},+}}(-\xi)$ and $\widehat{v}(-\xi) = \overline{\widehat{v}(\xi)}$ for real-valued data, $\xi \in \mathbb{R}^d$. These make our transforms extremely fast. We shall give the computation complexity estimate in next subsection.

We next present the backward transform for the digital affine shear filter bank, which is simply the adjoint transform of the forward transform due to the tightness of the filter bank.

Algorithm 2. Backward Digital Affine Shear Transform (Reconstruction)

(a) **Input:** Digital affine shear coefficients: $\{c^J\} \cup \{c^{j,\vec{\ell},n,+} : n = 1, \dots, d, |\vec{\ell}| \leqslant \vec{r}_j, j = 0, \dots, J-1\}$ and a digital affine shear filter bank $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^{J-1})$ defined as in (3.7) with $J \leqslant \min\{K_1-1, \dots, K_d-1\}$ and $k_0, \dots, k_{J-1}$ of nonnegative integers determining the number of shear directions in $\mathcal{B}_j := \{a \downarrow \mathsf{M}, b^{j,\vec{\ell},\pm}(\mathsf{E}_n\cdot) \downarrow \mathsf{A}_{j,n} : |\vec{\ell}| \leqslant \vec{r}_j\}$ with $\vec{r}_j := (2^{k_j}, \dots, 2^{k_j}) \in \mathbb{Z}^{d-1}$, $\mathsf{A}_{j,n} := \mathsf{E}_n \operatorname{diag}(2, 2^{k_j}\mathsf{I}_{d-1})$, and $a, b^{j,\vec{\ell},\pm}$ being defined as in (3.1) and (3.5), respectively. c^J is on the lattice $\Lambda(K_1^0, \dots, K_d^0)$.

(b) **Output:** d-dimensional real-value data v_0 on $\Lambda(K_1, \dots, K_d)$ with $K_n = K_n^0 + J$ for $n = 1, \dots, d$.

(c) **Main steps:**

1: Initialization: $\widehat{v} \leftarrow \mathscr{F}(c^J)$, $j \leftarrow J-1$, and $(K_1, \dots, K_d) \leftarrow (K_1^0+1, \dots, K_d^0+1)$.
2: **while** $j >= 0$ **do**
3: Low-pass subdivision $\mathcal{S}_{a,\mathsf{M}}v$: Initialize zero matrix $\widehat{v}_{lo}$ on $\widehat{\Lambda}(K_1, \dots, K_d)$. Compute $\widehat{v}_{lo}$ by $\widehat{v}_{lo}(\mathsf{k}) \leftarrow \widehat{v}(\mathsf{M}^\mathsf{T}\mathsf{k}) \cdot \widehat{a}(\mathsf{k})$ for $\mathsf{k} \in \operatorname{supp}(\widehat{a})$.
4: Initialize a zero matrix $\widehat{v}_{hi}$ on $\widehat{\Lambda}(K_1, \dots, K_d)$.
5: **for** each $b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot) \downarrow \mathsf{A}_{j,n} \in \mathcal{B}_j$ **do**
6: $b \leftarrow b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot)$, $\mathsf{A} \leftarrow \mathsf{A}_{j,n}$, and $\widehat{c} \leftarrow \mathscr{F}(c^{j,\vec{\ell},n,+})$.
7: High-pass subdivision for all $\mathcal{S}_{b^{j,\vec{\ell},+}(\mathsf{E}_n\cdot),\mathsf{A}^{j,n}}v$: $\widehat{v}_{hi}(\mathsf{A}^\mathsf{T}\mathsf{k} + 2\pi\omega_\mathsf{k}) \leftarrow \widehat{v}_{hi}(\mathsf{A}^\mathsf{T}\mathsf{k} + 2\pi\omega\mathsf{k}) + \widehat{c}(\mathsf{k})\widehat{b}(\mathsf{A}^\mathsf{T}\mathsf{k} + 2\pi\omega_\mathsf{k})$ for $\mathsf{A}^\mathsf{T}\mathsf{k} + 2\pi\omega_\mathsf{k} \in \operatorname{supp}(\widehat{b})$.
8: **end for**
9: High-pass subdivision $\mathcal{S}_{b^{j,\vec{\ell},-}(\mathsf{E}_n\cdot),\mathsf{A}^{j,n}}v$: $\widehat{v}_{hi} \leftarrow \widehat{v}_{hi} + \overline{\widehat{v}_{hi}(-\cdot)}$.
10: $\widehat{v} \leftarrow \widehat{v}_{lo} + \widehat{v}_{hi}$, $(K_1, \dots, K_d) \leftarrow (K_1+1, \dots, K_d+1)$, $j \leftarrow j-1$.
11: **end while**
12: $v_0 \leftarrow \mathscr{F}^{-1}(\widehat{v})$.

3.3 Redundancy rate and computational complexity

The redundancy rate measures the storage complexity of a filter bank transform, which is usually given by the ratio of size of the output coefficients and the size of the input data. Let $N = 2^{K_1+\cdots+K_d}$ be the size of the input data. For our digital affine shear transforms, at scale j, the output coefficients are $c^{j,\vec{\ell},n,+}$ for $n = 1, \dots, d$, $|\vec{\ell}| \leqslant \vec{r}_j$ with $\vec{r}_j = (2^{k_j}, \dots, 2^{k_j}) \in \mathbb{Z}^{d-1}$. The coefficient matrix $c^{j,\vec{\ell},n,+}$ is on the lattice $\Lambda(K_1-j-k_j, \dots, K_{n-1}-j-k_j, K_n-j-1, K_{n+1}-j-k_j, \dots, K_d-j-k_j)$, which is of size $\frac{N}{2^{dj}} \cdot \frac{1}{2^{1+(d-1)k_j}}$. The total number of output coefficients at scale j is $d(2^{k_j+1}+1)^{d-1}$. Consequently, the size of the total output coefficients at scale j is:

$$\frac{N}{2^{dj}} \cdot \frac{1}{2^{1+(d-1)k_j}} \times d(2^{k_j+1}+1)^{d-1} \times 2.$$

The '$\times 2$' is due to complex-value coefficients. The low-pass coefficient c^J is of size $N/2^{dJ}$. Therefore, the total size of output coefficients is

$$N\left(\sum_{j=0}^{J-1} \frac{2d(2^{k_j+1}+1)^{d-1}}{2^{dj} \cdot 2^{1+(d-1)k_j}} + \frac{1}{2^{dJ}}\right) = N\left(\sum_{j=0}^{J-1} \frac{d(2^{-k_j}+2)^{d-1}}{2^{dj}} + \frac{1}{2^{dJ}}\right) \leqslant Nd(2^{-k_{min}}+2)^{d-1}\left(\sum_{j=0}^{\infty} 2^{-dj}\right)$$
$$= Nd(2^{-k_{min}}+2)^{d-1}\frac{2^d}{2^d-1},$$

where $k_{min} := \min\{k_j : j = 0, \dots, J-1\}$. The redundancy rate is given by

$$r = \left(\sum_{j=0}^{J-1} \frac{d(2^{-k_j}+2)^{d-1}}{2^{dj}} + \frac{1}{2^{dJ}} \right) \leqslant d(2^{-k_{min}}+2)^{d-1} \frac{2^d}{2^d-1}.$$

The following table gives the bound of the redundancy rate with respect to a fixed dimension d and a fixed k_{min}.

$d \backslash k_{min}$	0	1	2	3	4	5	6	2^d	TP-$\mathbb{C}$TF$_6$ [12]
2	8.00	6.67	6.00	5.67	5.50	5.42	5.38	4	10.66
3	30.86	21.43	17.36	15.48	14.58	14.15	13.93	8	29.71
4	115.20	66.67	48.60	40.94	37.43	35.76	34.94	16	85.33
5	418.06	201.61	132.28	105.24	93.40	87.86	85.19	32	249.80

Table 1. Theoretical redundancy bound of r for digital affine shear transforms.

Interestingly, instead of increasing, the redundancy rate decreases as k_{min} increases, which determines the number of shear directions $\geqslant d(2^{k_{min}+1}+1)^{d-1}$. In dimension two, $k_j = 2$ corresponds to 18 directions at scale j. In dimension three, $k_j = 2$ corresponds to 243 directions at scale j. In Table 1, we also compare with 2^d and the redundancy rate of TP-$\mathbb{C}$TF$_6$ in [12], which has redundancy rate $2^d \times \frac{3^d-1}{2^d-1}$. One can see from the table that the redundancy rate increases slower than TP-$\mathbb{C}$TF$_6$ with respect to dimension d and $k_{min} \geqslant 1$. Actually, the ratio between these two is bounded by $\frac{d(2^{-k_{min}}+2)^{d-1}}{3^d-1} \to 0$ as $d \to \infty$ for $k_{min} \geqslant 1$.

For fast computational realization, with a fixed size of data and level J of decomposition, the digital affine shear filter bank $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^{J-1})$ can be precomputed. By the compact property of $\widehat{b}$ for $b \in \mathcal{B}_j$, we only need to store the support indices and values $\{(\mathsf{k}, \widehat{b}(\mathsf{k})) : \widehat{b}(\mathsf{k}) \neq 0, \mathsf{k} \in \widehat{\Lambda}(K_1 - j, \dots, K_d - j)\}$. Again, by the symmetry property, we only need to store half of such information. Similar to the above analysis, we can conclude that the total size is again bounded by rN for N the size of input data and r the redundancy rate the transform.

We next discuss the computational complexity of the digital affine shear transforms. In Algorithm 1 (decomposition), the computational complexity of Line 1 (Initialization, DFT) is $N\log(N)$. Let us next analyze Lines 3-7 In the `while` loop with respect to scale j. At Line 3, we need to perform $\frac{N}{2^{d(j+1)}}$ (complex) multiplications for computing $\widehat{v}|_{\widehat{\Lambda}(K_1-j-1,\dots,K_d-j-1)}$. Since the support indices are precomputed, the main computational cost in the `for` loop is Line 6 for multiplication and Line 7 for inverse DFT. The total number of (complex) multiplications in the `for` loop is

$$\left(\frac{N}{2^{dj}} \cdot \frac{1 + \log(N) - dj - 1 - (d-1)k_j}{2^{1+(d-1)k_j}} \right) \times d(2^{k_j+1}+1)^{d-1}.$$

The last step Line 13 (inverse DFT) requires $\frac{N}{2^{dJ}}(\log(N) - dJ)$ numbers of (complex) multiplications. Consequently, the total computational cost is

$$\begin{aligned}
&N\log N + \sum_{j=0}^{J-1} \left[\frac{N}{2^{dj}} \cdot \frac{1 + \log N - dj - 1 - (d-1)k_j}{2^{1+(d-1)k_j}} \times d(2^{k_j+1}+1)^{d-1} + \frac{N}{2^{d(j+1)}} \right] \\
&\qquad + \frac{N}{2^{dJ}}(\log N - dJ) \\
\leqslant &N\log N + N(1+\log N) \left(\sum_{j=0}^{J-1} \left[\frac{d(2^{k_j+1}+1)^{d-1}}{2^{dj} \cdot 2^{1+(d-1)k_j}} + \frac{1}{2^{d(j+1)}} \right] + \frac{1}{2^{dJ}} \right) \\
\leqslant &N\log N + N(1+\log N)(r + (2^d-1)^{-1}) \\
\leqslant &N[(1+\log N)(r + (2^d-1)^{-1} + 1) - 1] \\
\leqslant &(r+2) \cdot (N + N\log N).
\end{aligned}$$

Roughly speaking, N mainly comes from the point-wise multiplications of filters and input data while $N \log N$ comes from the performance of DFT for output coefficients. Our computational cost is proportional to the computational complexity of DFT for input data of size N. The ratio is controlled by the redundancy rate r. Since the backward transform Algorithm 2 basically reverses the steps in Algorithm 1, the computational cost of Algorithm 2 is the same as Algorithm 1. See Figure 1 for the computational time comparison between the forward digital affine shear transform and the standard FFT in dimension three. J is set to be 4 and $k_0 = 2$, $k_1 = 1$, $k_2 = 1$, $k_3 = 0$. For each $N = n^3$ with n ranging from 80 to 256 (step size 16), the computational times (in second) for a standard FFT (FFT Time) and our forward digital affine shear transform (DAS Time) are obtained. The line of DAS Time vs. FFT time is plotted in Figure 1 together with the line $r \times$ FFT Time with $r \approx 17.87$ being the redundancy rate. We can see that the computational complexity of our digital affine shear transform is indeed controlled by the FFT Time.

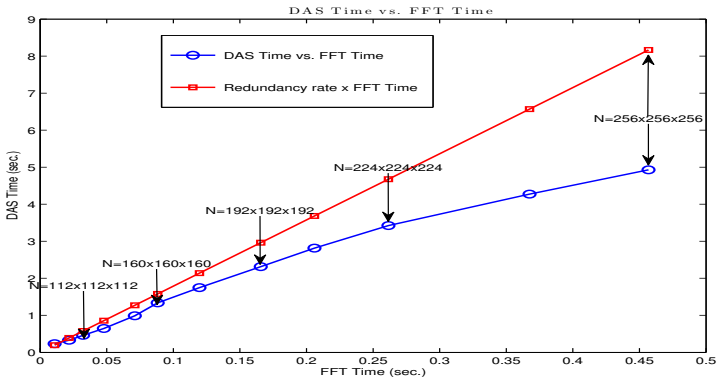

Figure 1. Comparison of computational costs of the forward digital affine shear transform and the standard FFT in MATLAB for $N = n^3$, where n ranges from 80 to 256 with step size 16.

4. NUMERICAL EXPERIMENTS ON IMAGE AND VIDEO DENOISING

In this section, we apply our digital affine shear transforms in the tasks of image/video denoising. We compare the performance of our systems to several other stat-of-art directional multiscale representation systems.

The peak-signal-noise-ratio (PSNR) index is used to measure the performance of different systems, which is defined to be

$$\mathrm{PSNR}(u, \widetilde{u}) = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(u, \widetilde{u})}, \tag{4.1}$$

where $u : \Lambda \to \mathbb{C}$ is the original data defined on a lattice Λ, $\widetilde{u}$ is the denoised data of u, and $\mathrm{MSE}(u, \widetilde{u})$ is the mean square error $\frac{1}{|\Lambda|} \sum_{\mathsf{k} \in \Lambda} |u(\mathsf{k}) - \widetilde{u}(\mathsf{k})|^2$ with $|\Lambda|$ the cardinality of the lattice Λ. The unit of PSNR is dB.

For the thresholding technique in the denoising task in our digital affine shear transforms, we shall employ the local-soft (LS) thresholding method. For each high-pass coefficient matrix $c \in \{c^J\} \cup \{c^{j,\vec{\ell},n,+} : n = 1, \dots, d, |\vec{\ell}| \leqslant \vec{r}_j, j = 0, \dots, J-1\}$, let b be the filter that induces c, that is, $c = \mathcal{T}_{b,\mathsf{A}} v_0$ for the input data v_0 and some downsampling matrix A, we first normalize it with respect to the norm $\|b\|_2$ of b to obtain $c_b := \frac{c}{\|b\|_2}$. The filter b can be computed by applying the backward transform to a delta data on the support of c and $\|b\|_2$ is the Frobenius norm of the reconstructed data. Let σ be the variance of a noise obeying normal distribution $N(0, \sigma^2)$ and Λ be the lattice for c. For each $\mathsf{k} \in \Lambda$, compute local coefficient variance $\sigma_c : \Lambda \to [0, \infty)$ by $\sigma_c(\mathsf{k}) := \sqrt{\left(\frac{1}{\#\Lambda_\mathsf{k}} \sum_{\mathsf{n} \in \Lambda_\mathsf{k}} |c_b(\mathsf{n})|^2 - \sigma^2\right)_+}$, where Λ_k is the lattice of the the same size as $[-L, L]^d$ centering at position k for some integer $L \geqslant 0$. Note that σ_c can be computed by convolution of c_b with a normalized window $\frac{[-L,L]^d \cap \mathbb{Z}^d}{(2L+1)^d}$. The threshold $t_c : \Lambda \to [0, \infty]$ is then define by $t_c(\mathsf{k}) = \frac{\sigma^2}{\sigma_c(\mathsf{k})}$, $\mathsf{k} \in \Lambda$. The soft-thresholding operator

$\eta_t^{soft}(x)$ and hard-thresholding operator $\eta_t^{hard}(x)$ for $t \geqslant 0, x \in \mathbb{C}$ is defined to be

$$\eta_t^{soft}(x) = \begin{cases} x - t\frac{x}{|x|} & \text{if } |x| > t, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \eta_t^{hard}(x) = \begin{cases} x & \text{if } |x| > t, \\ 0 & \text{otherwise.} \end{cases}$$

The local-soft thresholding $\eta_{t_c}^{ls}(c_b) : \Lambda \to \mathbb{C}$ applying to c_b with threshold t_c is then defined to be $[\eta_{t_c}^{ls}(c_b)](\mathsf{k}) := \eta_{t_c(\mathsf{k})}^{soft}(c_b(\mathsf{k}))$, $\mathsf{k} \in \Lambda$. The threshold coefficient $\widetilde{c}$ from c is then renomralized by $\widetilde{c} := \|b\|_2 \cdot \eta_{t_c}^{ls}(c_b)$.

Figure 2. 2D images `Lena` and `Barbara` (left two) and the first frame of 3D data `Mobile` and `Coastguard` (right two).

4.1 Comparisons on image denoising and inpainting

We first apply our system to the task of denoising in image processing. The parameters $c_0, \epsilon_0, \epsilon_1$ of a_1, a are given by $c_0 = \frac{33}{32}$, $c_1 = \pi$, $\epsilon_0 = \frac{69}{128}$, $\epsilon_1 = \frac{51}{512}$ and $\varepsilon = \frac{1}{2}$ for $\boldsymbol{\gamma}_\varepsilon$. We choose $J = 5$ for $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^J)$ as in (3.7); that is, we decompose to 5 scales. The shear parameters $(k_0, k_1, k_2, k_3, k_4)$ is set to be $(2, 1, 1, 1, 0)$. That is, for the finest scale $j = 0$, we use totally $2(2^{k_0+1} + 1) = 18$ shear directions (9 on horizontal cone and 9 on vertical cone). For next three scales $j = 1, 2, 3$, we use 10 shear directions, and for the coarsest scale, we use 6 shear directions. The redundancy of our system $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^4)$ is 6.165. The convolution window size L to compute local coefficient variance σ_c is set to be 4, i.e., we are using 9×9 window.

We test two standard images: `Lena` and `Barbara`; see Figure 2. Both are of size 512×512. We first employ symmetric boundary extension (with 32 pixels) on the noisy image to avoid boundary effect. We then apply our forward transform to obtained the coefficients. After performing the local-soft threshold procedure, we then apply the backward transform to the thresholded coefficients and throw away the extended boundary to obtained the final denoised image.

We compare our denoising performance to several other stat-of-art directional multiscale representation systems: dual-tree complex wavelets [18], tensor product complex tight framelets [11, 12] , curvelets [2, 3], compact support shearlets [16], contourlets [4]. We download each of their available packages and run their denoising codes for both `Lena` and `Barbara`. The 2D dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) in [18] has redundancy rate 4. The number of directional filters of DT-$\mathbb{C}$WT at each scale is 6 covering $\pm 15°, \pm 45°, \pm 75°$. The total number of scales is 6. Bivariate shrinkage thresholding technique is employed for denoising. The TP-$\mathbb{C}$TF$_6$ is detailed in [11, 12], which has redundancy rate 10.67 and 14 directional filters for each scale. The total number of scales is 5 and it also use bivariate shrinkage thresholding technique. The curvelet transform (FDCT at `http://www.curvelab.org`) has two implementations: one uses un-equispace FFT, another uses frequency wrapping. Here we use the wrapping package; detailed information can be found at [3]. The total number of scales is 5. At the finest scale, the FDCT uses isotropic wavelet transform to avoid checkerboard effect. At scale 4, 32 (angular) directions are used. At scales 3 and 2, 16 (angular) directions are used. At the coarsest scale, 8 (angular) directions are used. The redundancy of the FDCT is about 2.8. The shearlet transform at `http://www.shearlab.org` also has many implementations and we choose the compactly supported shearlet implementation DNST as in [16], which has the best performance so far in the ShearLab package. For DNST, the total number of scales is 4. 16 shear directions are used for finest scale 4 and 3; while 8 shear directions are used for the other two scales. The redundancy of DNST is 49. The contourlet transform [4] (NSCT package at `http://minhdo.ece.illinois.edu/software`) has redundancy rate 53. It uses 4, 8, 8, 16, 16 directions in the scales from coarser to finer. The three transforms FDCT, DNST, NSCT use hard thresholding for denoising.

We compare the denoising performance over different Gaussian noise $N(0, \sigma^2)$ with noise level σ in {5, 10, 30, 50, 80, 100}. The comparison results are presented in Table 2. The values in the brackets are gain (+) or loss

(-) of our method comparing to other methods. From Table 2, we see significant improvement over FDCT and NSCT for both `Lena` and `Barbara`. Comparing our method with DNST, when noise level is small $\sigma \leqslant 30$, DNST performs better than our method for the image `Lena` while our method perform better than DNST when noise level is high $\sigma > 30$ for both `Lena` and `Barbara`. The performance of DT-$\mathbb{C}$WT and our method is comparable when noise level is small and our method outperforms DT$\mathbb{C}$WT when noise level is high. TP-$\mathbb{C}$TF$_6$ outperforms our method for `Lena` when noise level $\sigma < 80$ and for `Barbara` when $\sigma \leqslant 30$. However, when noise level is high, our method outperforms TP-$\mathbb{C}$TF$_6$. In summary, we conclude that our method is in general better than DT-$\mathbb{C}$WT, TP-$\mathbb{C}$TF$_6$, FDCT, DNST, NSCT, especially for texture-rich image Barbara and for high noise level ($\sigma > 50$).

512 × 512 Lena						
σ	DAS	DT-$\mathbb{C}$WT	TP-$\mathbb{C}$TF$_6$	DNST	FDCT	NSCT
5	38.15	38.25(-0.10)	38.37(-0.22)	38.01(0.14)	35.77(2.38)	37.71(0.44)
10	35.13	35.19(-0.06)	35.48(-0.35)	35.35(-0.22)	33.37(1.76)	34.92(0.21)
30	30.62	30.50(0.12)	30.80(-0.18)	30.68(-0.06)	29.34(1.28)	30.32(0.30)
50	28.49	28.22(0.27)	28.54(-0.05)	28.21(0.28)	27.19(1.30)	28.02(0.47)
80	26.55	26.15(0.40)	26.47(0.08)	25.78(0.77)	25.16(1.39)	25.80(0.75)
100	25.64	25.20(0.44)	25.52(0.12)	24.58(1.06)	24.22(1.42)	24.71(0.93)
512 × 512 Barbara						
σ	DAS	DT-$\mathbb{C}$WT	TP-$\mathbb{C}$TF$_6$	DNST	FDCT	NSCT
5	37.33	37.37(-0.04)	37.84(-0.51)	37.17(0.16)	33.83(3.50)	36.96(0.37)
10	33.65	33.54(0.11)	34.18(-0.53)	33.62(0.03)	29.17(4.48)	33.35(0.30)
30	28.33	27.89(0.44)	28.38(-0.05)	27.97(0.36)	24.44(3.89)	27.28(1.05)
50	26.01	25.36(0.65)	25.71(0.30)	25.31(0.70)	23.38(2.63)	24.57(1.44)
80	23.99	23.27(0.72)	23.53(0.46)	22.96(1.03)	22.22(1.77)	22.65(1.34)
100	23.08	22.42(0.66)	22.64(0.44)	22.06(1.02)	21.61(1.47)	21.90(1.18)

Table 2. PSNR of denoised `Lena` and `Barbara`.

4.2 Comparisons on video denoising and inpainting

We next apply our system to the task of denoising in video processing. The parameters $c_0, c_1, \epsilon_0, \epsilon_1$ of a_1, a are same as before except $\varepsilon = 0.469$ for $\boldsymbol{\gamma}_\varepsilon$. We choose $J = 4$ for $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^J)$ as in (3.7); that is, we decompose to 4 scales. The shear parameters (k_0, k_1, k_2, k_3) is set to be $(2, 1, 1, 0)$. That is, for the finest scale $j = 0$, we use totally $3(2^{k_0+1} + 1)^2 = 243$ shear directions (81 for each cone). For next two scales $j = 1, 2$, we use 75 shear directions, and for the coarsest scale, we use 27 shear directions. The redundancy of our system $\mathsf{DAS}(\{\mathcal{B}_j\}_{j=0}^3)$ is 17.88. The convolution window size L to compute local coefficient variance σ_c is set to be 2, i.e., we are using $5 \times 5 \times 5$ window.

We test two videos: `Mobile` and `Coastguard`, which can be downloaded from `http://www.shearlab.org`; see Figure 2 for the first frame of each video. Both videos are of size $192 \times 192 \times 192$. We first employ symmetric boundary extension (with 16 pixels) on the noisy image to avoid boundary effect. We then apply our forward transform to obtained the coefficients. After performing the local-soft threshold procedure, we then apply the backward transform to the thresholded coefficients and throw away the extended boundary to obtained the final denoised image.

We compare our denoising performance to 3D dual-tree complex wavelets [18], 3D tensor product complex tight framelets [11, 12] , surfacelets [17], 3D compact support shearlets [16]. The 3D dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) in [18] has redundancy rate 8. The number of directional filters of DT-$\mathbb{C}$WT at each scale is 56. The number of scales is 5. The 3D TP-$\mathbb{C}$TF$_6$ is detailed in [12], which has redundancy rate 29.71 and 208 directional filters for each scale. The number of scales is 4. Bivariate shrinkage thresholding technique is employed for both DT$\mathbb{C}$WT and TP-$\mathbb{C}$TF$_6$. For 3D DNST from the ShearLab package, we choose the one with redundancy rate 154 (3 scales). The surfacelet transform (SURF) from SurfBox at `http://minhdo.ece.illinois.edu/software` has redundancy rate 6.4. The 3D DNST and sufacelet transform use hard thresholding for denoising.

We compare the denoising performance over different noise level $\sigma \in \{10, 20, 30, 50, 80, 100\}$. The comparison results are presented in Table 3. From Table 3, we see that our method outperforms both two low-redundant 3D transforms DT$\mathbb{C}$WT and surfacelets. For the other two high-redundant 3D transforms TP-$\mathbb{C}$TF$_6$ and DNST, our performance is worse while noise level is small ($\sigma < 50$)but significant better when noise level is high ($\sigma \geqslant 50$).

	192 × 192 × 192 Mobile					192 × 192 × 192 Coastguard				
σ	DAS	DT-$\mathbb{C}$WT	TP-$\mathbb{C}$TF$_6$	SURF	DNST	DAS	DT-$\mathbb{C}$WT	TP-$\mathbb{C}$TF$_6$	SURF	DNST
10	34.99	34.72(0.27)	35.52(-0.53)	32.79(2.20)	35.91(-0.92)	33.70	33.21(0.49)	34.15(-0.45)	30.86(2.84)	33.81(-0.11)
20	31.50	30.86(0.64)	31.77(-0.27)	29.95(1.55)	32.18(-0.68)	30.27	29.61(0.66)	30.62(-0.35)	28.26(2.01)	30.28(-0.01)
30	29.57	28.67(0.90)	29.66(-0.09)	28.26(1.31)	29.99(-0.42)	28.47	27.71(0.76)	28.73(-0.26)	26.87(1.60)	28.40(0.07)
50	27.26	26.06(1.20)	27.08(0.18)	26.11(1.15)	27.22(0.04)	26.40	25.56(0.84)	26.48(-0.08)	25.17(1.23)	26.17(0.23)
80	25.20	24.00(1.20)	24.82(0.38)	24.25(0.95)	24.75(0.45)	24.65	23.83(0.82)	24.53(0.12)	23.61(1.04)	24.17(0.48)
100	24.23	23.17(1.06)	23.82(0.41)	23.40(0.83)	23.62(0.61)	23.86	23.08(0.78)	23.65(0.21)	22.87(0.99)	23.22(0.64)

Table 3. PSNR of denoised `Mobile` and `Coastguard`.

ACKNOWLEDGMENTS

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 108913).

REFERENCES

[1] B. G. Bodmann, G. Kutyniok, and X. Zhuang, Gabor shearlets, *Appl. Comput. Harmon. Anal.* **38** (1) (2015), 87–114.

[2] E. J. Candès and D. L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, *Comm. Pure Appl. Math.* **57** (2) (2004), 219–266.

[3] E. J. Candès, L. Demannet, D. Donoho, and L. Ying, Fast discrete curvelet transforms, *Multiscale Model. Simul.* **5** (3) (2006), 861–899.

[4] A. L. Cunha, J. Zhou, and M. N. Do, The nonsubsampled contourlet transform: theory, design, and applications, *IEEE Trans. Image Proc.* **15** (10) (2006), 3089–3101.

[5] I. Daubechies, Ten Lectures on Wavelets, CBMS-NSF Regional Conference Series in Applied Mathematics, **61**, SIAM, Philadelphia, PA, 1992.

[6] K. Guo, D. Labate, W. Lim, G. Weiss, and E. Wilson, Wavelets with composite dilations, *Electr. Res. Ann. AMS.* **10** (2004), 78–87.

[7] K. Guo, G. Kutyniok, and D. Labate, Sparse multidimensional representations using anisotropic dilation and shear operators, Wavelets and Splines (Athens, GA, 2005), Nashboro Press, Nashville, TN (2006), 189-201.

[8] K. Guo and D. Labate, Optimal sparse multidimensional representation using shearlets, *SIAM J. Math. Anal.* **9** (2007), 298–318.

[9] K. Guo and D. Labate, The construction of smooth Parseval frames of shearlets, *Math. Model. Nat. Phenom.* **8** (2013), 82–105.

[10] B. Han, Nonhomogeneous wavelet systems in high dimensions. *Appl. Comput. Harmon. Anal.* **32** (2012), 169–196.

[11] B. Han and Z. Zhao, Tensor product complex tight framelets with increasing directionality, *SIAM J. Imaging Sci.* **7** (2014), 997–1034.

[12] B. Han, Z. Zhao, and X. Zhuang, Directional tensor product complex tight framelets with low redundancy, *Appl. Comput. Harmon. Anal.*, to appear.

[13] B. Han and X. Zhuang, Smooth affine shear tight frames with MRA structures, *Appl. Comput. Harmon. Anal.* **39** (2) (2015), 300–338.

[14] E. J. King, G. Kutyniok, and X. Zhuang, Analysis of data separation and recovery problems using clustered sparsity, *J. Math. Imaging Vis.* **48** (2014), 205–234.

[15] G. Kutyniok, M. Sharam, and X. Zhuang, ShearLab: A rational design of a digital parabolic scaling algorithm, *SIAM J. Imaging Sci.* **5** (4) (2012), 1291–1332.

[16] W.-Q. Lim, Nonseparable shearlet transform. *IEEE Trans. Image Proc.* **22** (2013), 2056 - 2065.

[17] Y. M. Lu and M. N. Do, Multidimensional directional filter banks and surfacelets, *IEEE Trans. Image Process.* **16** (2007), 918–931.

[18] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, The dual-tree complex wavelet transform, *IEEE Signal Process. Mag.* **22** (6) (2005), 123–151.