Proceedings of the Project Review, Geo-Mathematical Imaging Group (Purdue University, West Lafayette IN), Vol. 1 (2010) pp. 25-50.

ON THE REPRESENTATION OF FUNCTIONS WITH GAUSSIAN WAVE PACKETS

FREDRIK ANDERSSON*, MARCUS CARLSSON†, AND LUIS TENORIO‡

1. Introduction. The main purpose of this paper is to develop algorithms to obtain sparse representations of functions of several variables using Gaussian wave packets. Numerically, we will primarily focus on the two-dimensional case; our main interest is the representation of *waves* related to the wave equation. Sparse representations are useful, in some cases essential, for large-scale problems that arise in, for example, seismic imaging [8, 23, 22, 33].

To analyze functions whose features vary at different resolution, one may use *wavelet transforms*, which employ pairs of translation and dilation operators [12, 24]. One can also consider a *time–frequency analysis* [19] using a combination of translation and modulation in the *short–time Fourier transform*. On the other hand, the use of *Gaussian wave packets* allows us to incorporate all three of the operations translation, dilation and modulation in the one-dimensional case. A classic one-dimensional example is music; it is often used as a motivation for the usage of both wavelets and time-frequency analysis. It also serves to motivate the three-parameter transform that we propose as musical notes contain (at least) three characteristic features: the tone, the time it is played and its duration. In higher dimensions Gaussian wave packets can also be used to include rotation invariance and anisotropic dilation parameters.

A simple way to represent functions in two dimensions, either by the wavelet or time-frequency methods, is to employ tensor products of the chosen one-dimensional functions. This has the disadvantage of representing horizontal and vertical features/waves well but not those that are not aligned with the coordinate axes. Therefore, a lot of work has been targeted into obtaining more rotation-invariant representations. Proposed methods include two-dimensional wavelets [3], steerable pyramids [27], brushlets [2], curvelets [9], shearlets [21] and beamlets [33]. The anisotropic dilation parameter has been shown to play a crucial theoretical role for solutions of wave equations, cf. [28]. From a practical perspective, it is useful when working with waves that have (locally) largely varying curvature. For waves with small curvature, a locally plane wave approximation works well, and it is preferable to work with representations where the extent in the non-oscillatory direction is much larger than in the oscillatory direction, whereas for waves with large curvature the extent in the two directions should be similar.

The one-dimensional continuous wavelet — or — continuous short–time Fourier transform are redundant in the sense that they map a function of one variables to a function of two variables coming from the use of two invariance properties (translation/scale or translation/modulation). In the continuous Gaussian wave packet representations we propose, the one-dimensional representation requires three variables while the two-dimensional uses up to six variables. Redundancy is often considered problematic as there is no unique way to represent a function. Although it is possible to construct an orthogonal basis through the discrete wavelet transform [12], the redundancy of the continuous representations often shines through in discrete representations in the sense that many approaches yields representations with more elements than necessary. On common such approach is to construct an inversion formula for the redundant representation, for instance by means of a *frame*, which provides *one* unique representation. Given that the function is reasonable well described by the frame elements, a *sparse* representation is sought for by using the unique representation provided by the frame, and simply threshold small elements. If the frame is redundant, then the redundancy will limit the effectiveness the approach, and the obtained representation will in fact

*Centre for Mathematical Sciences, Lund University, Box 118, SE-22100, Lund, Sweden and Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907, USA

†Centre for Mathematical Sciences, Lund University, Box 118, SE-22100, Lund, Sweden and Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907, USA

‡Mathematical and Computer Sciences, Colorado School of Mines, Chauvenet 232, Golden, CO 80401, USA

not be particularly sparse. Obtaining representations with as low redundancy as possible has been desirable, e.g., for critical sampling for Gabor analysis [17].

In recent years, variable selection methods to obtain sparse representations have proved successful, see for example [31, 16, 11, 26]. A large degree of redundancy should then be regarded as having a large dictionary of functions to select from in order to find an optimal approximation with few terms. In contrast to the thresholded frame approach mentioned above, the main difficultly with such an approach is to find a good representation in reasonable time.

In this paper, we will start in a continuous setting and review invariance properties of Gaussian wave packets. We will construct inversion formulas by means of isometry arguments. Moreover, we will study some group properties and provide analytic formulas that prove useful in the analysis.

We will then focus on two-dimensional Gaussian wave packets. At the numerical implementation stage, the representation of a function f as a linear combination of Gaussian wave packets reduces to the computation of a vector of coefficients c such that $\mathcal{V}c \approx f$, where $\mathcal{V}$ the synthesis operator defined by the wave packet representation. We will show how to construct a fast numerical algorithm for the analysis operator $\mathcal{V}^*$ corresponding to a particular sampling. This algorithm relies on the usage of FFT and USFFT [6, 15].

To determine the coefficients c we use an ℓ^1 regularization approach; c is chosen to minimize $\|\mathcal{V}c - f\|_2^2 + \mu\|c\|_1$ for a fixed value of the regularization parameter μ. From the statistical point of view, this ℓ^1 approach leads to shrinkage estimators that can reduce the mean square error of the estimates. The ℓ^1 penalty leads to exact zero values for small coefficients and thus performs a type of variable selection process that helps control overfitting and yields parsimonious models that are easier to compute and interpret [7, 29].

Most large-scale algorithms for solving the ℓ^1-regularized approach discussed above, rely on fast numerical methods for applying the analysis $\mathcal{V}^*$ and synthesis $\mathcal{V}$ operators. However, for the case with large redundancy, the full application of $\mathcal{V}$ and $\mathcal{V}^*$ is unnecessarily expensive. For instance, working with redundant representations with a redundancy factor of 100 and seeking a sparse representation with a compression ration of 1:100, implies that only 1 out of 10000 entries of the coefficient vector c would actually be used.

We will explore the special structure of Gaussian wave packets to construct a new algorithm for obtaining ℓ^1-regularized solutions by taking advantage of the high sparsity of the solutions c to reduce the number of full application of the operators $\mathcal{V}$ and $\mathcal{V}^*$.

2. Continuous Gaussian wave packet transforms.

Definitions. To motivate the notation that will be used, we start by considering the product of a Gaussian function of one variable and an oscillatory exponential. We will refer to such functions as one-dimensional *Gaussian wave packets*. Assuming that the Gaussian is centered at the origin, two parameters will be used to parametrize it; a frequency parameter and a parameter describing the width of the Gaussian. It will be convenient to measure this width relative to the frequency parameter. We therefore use the following parametrization

$$\varphi_{\alpha,\eta}(x) = e^{2\pi i \eta\, x}\, e^{-(\eta/\alpha)^2 \ln(16)\, x^2}.$$

The factor ln(16) comes from the fact that we want the half-width of the Gaussian to be equal to α oscillations with frequency η. When choosing $\eta = \alpha = 1$, this means that $e^{-\ln(16)(1/2)^2} = 1/2$ as we would like it to be. The parameter α describes the number of oscillations within the half-width of the Gaussian and controls the shape of the wave packet, while the parameter η determines the frequency of the oscillations and also plays the role of scale parameter.

In two dimensions, we include a Gaussian wave packet in each direction:

$$\varphi_{\alpha,\beta,\eta}(x) = \rho_{\alpha,\beta,\eta}\, e^{2\pi i \eta^{\mathrm{t}} x}\, e^{-x^{\mathrm{t}} \Lambda(\alpha,\beta,\eta) x}, \tag{2.1}$$

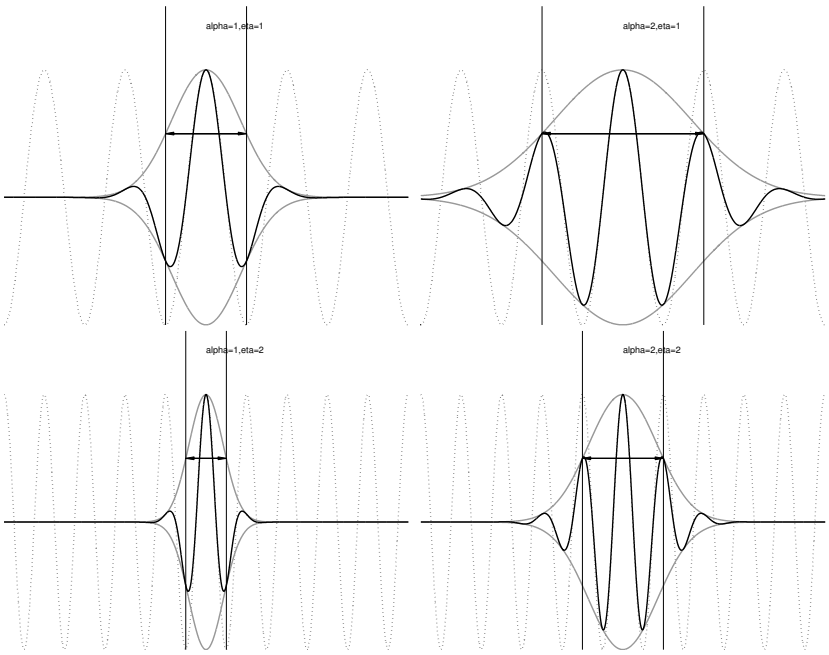

FIG. 1. *One-dimensional Gaussian wave packets. The parameter α that describes the width of the Gaussian is in terms of the number of oscillations within a half-width. The two top panels use $\eta = 1$, while the two bottom ones have $\eta = 2$. The two left panels have $\alpha = 1$, while the two right ones have $\alpha = 2$.*

where t denotes the transpose, $x = (x_1, x_2)^{\mathsf{t}}$, $\eta = (\eta_1, \eta_2)^{\mathsf{t}}$ and the matrix

$$\Lambda(\alpha, \beta, \eta) = \ln(16) \begin{pmatrix} |\eta|^2/\alpha^2 & 0 \\ 0 & |\eta|^2/\alpha^2\beta^2 \end{pmatrix}$$

describes the decay coefficients.

The normalization factor

$$\rho_{\alpha,\beta,\eta} = \sqrt{\frac{2\ln(16)}{\pi}} \frac{|\eta|}{\alpha\sqrt{\beta}}$$

is chosen so that $\varphi_{\alpha,\eta}$ has unit L^2-norm. Note that $\Lambda(\alpha, \beta, \eta)$ and $\rho_{\alpha,\beta,\eta}$ are invariant under rotations of η. The parameter β describes the ratio between the half-widths of the Gaussians in the x_1 and x_2 directions.

Finally, we define Gaussian wave packets not centered at the origin and with axes not parallel to the x_1 and x_2 axes; the sinusoids and the Gaussian functions are centered at a chosen $y \in \mathbb{R}^2$ and the axes of the Gaussian function are subject to a rotation Θ, where

$$\Theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad \theta \in [0, 2\pi).$$

Let $T_y(x) = x - y$ be the translation by y. The general Gaussian wave packet is defined as

$$\varphi_\gamma(x) = \varphi_{\alpha,\beta,\Theta\eta}(\Theta T_y(x)); \quad \gamma = (y, \eta, \alpha, \beta, \theta).$$

It is often desirable to lock η to θ as follows; $\eta = |\eta|(\cos\theta, \sin\theta)$. Then the oscillation aligns with the semi-axes, and $\Lambda(\alpha, \beta, \theta)$ describes the decay in the direction along and orthogonal to the oscillation, respectively, as indicated in Figure 2.

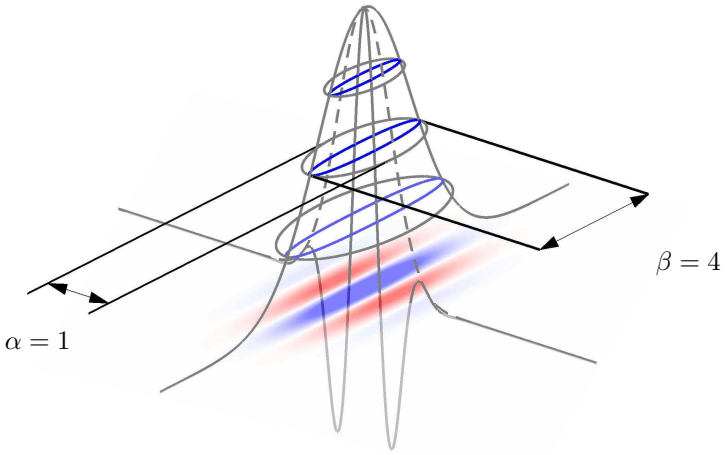

FIG. 2. *Example of a two-dimensional Gaussian wave packet with level sets at heights 0.25, 0.5 and 1. The gray level sets shows the amplitude, while the colored ones illustrate the real part of the function. The anisotropy of the packet is illustrated with $\alpha = 1$ oscillations in the oscillatory direction and $\beta = 4$ times as many oscillations in the perpendicular direction.*

To summarize, the parameters that define the Gaussian wave packets are:

- y - Central location
- η - Frequency/scale
- θ - Orientation of axes
- α - Number of oscillations within a half-width, (assuming $\eta = |\eta|(\cos\theta, \sin\theta)$)
- β - Ratio of the half-widths along and perpendicular to the direction of oscillation, (assuming $\eta = |\eta|(\cos\theta, \sin\theta)$

It is natural to think about wave packets where the oscillation is aligned with the semi-axis of the Gaussian. However, there are several examples where this alignment fails. One such is in modeling of seismic data acquisition. Let us assume that we have decomposed a wave field in two dimensions (horizontal position and depth) into gaussian wave packets, and (somewhat simplified) model the action of the wave equation on these wave packet as rigid motion. By seismic data acquisition means that this field is recorded at the surface (where the depth variable i equal to zero) over time. It then turns out that the recorded data will have the form (2.1), with $\eta/|\eta| \neq \theta$. Another example where the alignment fails is cross-correlation of wave packets. If we assume that we have decomposed two fields into wave packets, then we can approximate the action of the wave equation on these by a rigid motion corresponding to a Hamiltonian flow. The Hamiltonian flow can be quickly computed by solving systems of ordinary differential equations. Under the assumption that two wave-packets move locally along straight lines in the vicinity of each other, the cross correlation (with respect to time) of the wave fields can be analytically computed as another linear combination of Gaussian wave packets. The resulting wave packets will, however, not have the direction of oscillation aligned with the semi-axis of the Gaussian, even if the decomposed wave fields had.

It is interesting to note that the set of Gaussian wave packets is invariant (up to multiplicative factors) under:

- Multiplication
- Convolution
- Fourier transformation

We will exploit these invariance properties in Section 2.

Isometry and invariance properties. In this section we discuss how to build a continuous frame of Gaussian wave-packets for functions defined on $\mathbb{R}^n$. Our objective is two-fold: to answer questions regarding the completeness in $L^2(\mathbb{R}^n)$, and to show that it is possible to construct inversion

formulas. However, since inversion formulas of this type will not in general produce *sparse* solutions, the efficient computation of sparse solutions from a discrete set of Gaussians will be addresed in Sections 4.

We use the (unitary) n-dimensional Fourier transform defined by

$$Ff(\xi) = \widehat{f}(\xi) = \int_{\mathbb{R}^n} f(x) e^{-2\pi i x^{\mathrm{t}} \xi} dx.$$

We will also make use of the following operators:

$$\begin{aligned}
(T_y\varphi)(x) &= \varphi(x-y), && y \in \mathbb{R}^n && \text{Translation,}\\
(D_\alpha\varphi)(x) &= |\alpha|^{-n/2}\varphi\left(\frac{x}{a}\right), && \alpha \in \mathbb{R} && \text{Dilation,}\\
(M_\omega\varphi)(x) &= \varphi(x)e^{2\pi i\langle x,\omega\rangle}, && \omega \in \mathbb{R}^n && \text{Modulation,}\\
(K_\beta\varphi)(x) &= |\beta|^{-(n-1)/2}\varphi\left(x_1, \frac{x_2}{\beta}, \ldots, \frac{x_n}{\beta}\right), && \beta \in \mathbb{R} && \text{Anisotropic dilation,}\\
(P_\gamma\varphi)(x) &= \varphi(x)e^{2\pi i\gamma}, && \gamma \in \mathbb{R} && \text{Phase shift,}\\
(R_\theta\varphi)(x) &= \varphi(\Theta x), && \theta \in S^{n-1} && \text{Rotation,}
\end{aligned}$$

where Θ is the $n \times n$-matrix that rotates vectors in the plane spanned by e_1 and θ in such a way that $\Theta(e_1) = \theta$, and which acts as the identity on the orthogonal complement. Note that if $n = 2$ and $\phi(x) = e^{-x_1^2 - x_2^2}$, then

$$D_{\alpha/|\eta|}K_\beta\phi(x) = |\beta|^{\frac{n-1}{2}}|\alpha|^{\frac{n}{2}} e^{x^{\mathrm{t}}\Lambda(\alpha,\beta,\eta)x}.$$

It is easy to show that all of these operators are unitary on $L^2(\mathbb{R}^n)$ with the following adjoints

$$D_\alpha^* = D_{1/\alpha} \quad T_y^* = T_{-y} \quad M_\omega^* = M_{-\omega} \quad K_\beta^* = K_{1/\beta} \quad P_\gamma^* = P_{-\gamma} \quad R_\theta^* = R_{\widetilde{\theta}},$$

where $\widetilde{\theta}$ denotes the mirror image of θ in the line spanned by e_1, i.e., if $\theta \leftrightarrow \Theta$, then $\widetilde{\theta} \leftrightarrow \Theta^{\mathrm{t}}$. The following rules for the Fourier transform will prove useful:

$$\begin{aligned}
FT_y &= M_{-y}F,\\
FD_\alpha &= D_{1/\alpha}F,\\
FM_\omega &= T_\omega F,\\
FR_\theta &= R_\theta F,\\
FK_\beta &= K_{1/\beta}F,
\end{aligned} \tag{2.2}$$

as well as the following commutative rules for the operators:

$$\begin{aligned}
T_yD_\alpha &= D_\alpha T_{y/\alpha}, & D_\alpha T_y &= T_{y\alpha}D_\alpha,\\
M_\omega D_\alpha &= D_\alpha M_{\alpha\omega}, & D_\alpha M_\omega &= M_{\omega/\alpha}D_\alpha,\\
R_\theta D_\alpha &= D_\alpha R_\theta, & K_\beta D_\alpha &= D_\alpha K_\beta,\\
T_yM_\omega &= P_{y\omega}M_\omega T_y, & M_\omega T_y &= P_{-y\omega}T_yM_\omega.\\
T_yR_\theta &= R_\theta T_{\Theta y}, & R_\theta T_y &= T_{\Theta^{\mathrm{t}}y}R_\theta\\
T_yK_\beta &= K_\beta T_{y\beta}, & K_\beta T_y &= T_{y/\beta}K_\beta\\
M_\omega R_\theta &= R_\theta M_{\Theta\omega}, & R_\theta M_\omega &= M_{\Theta^{\mathrm{t}}\omega}R_\theta\\
M_\omega K_\beta &= K_\beta M_{\beta\omega}, & K_\beta M_\omega &= M_{\omega/\beta}K_\beta.
\end{aligned} \tag{2.3}$$

As mentioned in the introduction the wavelet transform employs a pair of translation T_y and dilation D_α operators while time–frequency analysis uses translation T_y and modulation M_ω. A

group theoretic approach has proven to be useful for understanding these operations; for instance it leads to inversion formulas. The chapter about the continuous wavelet transform in [24] and Chapter in [19] are good references in this regards, as well as [20].

For our purposes it will be illustrative to quickly recapitulate some group theoretic results for the continuous wavelet transform. Here, a function $f \in L^2(\mathbb{R})$ is mapped to a function in $L^2(\mathbb{R}^+ \times \mathbb{R})$ through the *continuous wavelet transform* $Wf(\alpha, y) = \langle T_y D_\alpha \psi, f\rangle$ for some choice of *wavelet* $\psi \in L^2$. For this transform to be invertible, ψ has to satisfy some regularity conditions that we derive below.

Since $T_{y_2} D_{\alpha_2} T_{y_1} D_{\alpha_1} = T_{(y_2+\alpha_2 y_1, \alpha_1\alpha_2)}$, we can associate the wavelet transform with the affine group with group operation

$$(\alpha_2, y_2) \circ (\alpha_1, y_1) = (\alpha_1\alpha_2, y_2 + a_2 y_1).$$

It is readily verified that the *left-invariant Haar measure* associated with the affine group can be written as a weighted Lebesgue measure, $d\mu_{_L} = \alpha^{-2} d\alpha dy$. One way to obtain an inversion formula is by constructing an isometry. We consider the wavelet transform as the mapping $W : L^2(\mathbb{R}) \to L^2(\mathbb{R}^+ \times \mathbb{R}, d\alpha dy/\alpha^2)$ induced by the (left–invariant) Haar measure. Hence, by Plancharel's theorem,

$$\begin{aligned}\|Wf\|^2 &= \int_{\mathbb{R}^+}\int_{\mathbb{R}} |\langle T_y D_\alpha \psi, f\rangle|^2 \frac{dyd\alpha}{\alpha^2} = \int_{\mathbb{R}^+}\int_{\mathbb{R}} |\big((D_\alpha\psi) * \bar{f}(-\cdot)\big)(y)|^2 \frac{dyd\alpha}{\alpha^2} \\ &= \int_{\mathbb{R}} |\widehat{f}(\xi)|^2 \int_{\mathbb{R}^+} |F D_\alpha \psi(\xi)|^2 \frac{d\alpha}{\alpha^2} d\xi.\end{aligned}$$

Using now the rules for the Fourier transform (2.3) and the fact $d\alpha/\alpha$ is the Haar measure of the multiplicative group $\mathbb{R}^+$, we obtain

$$\begin{aligned}\|Wf\|^2 &= \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+} |D_{1/\alpha}\widehat{\psi}|^2 \frac{d\alpha}{\alpha^2} d\xi = \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+} |\widehat{\psi}(\alpha\xi)|^2 \frac{d\alpha}{\alpha} d\xi \\ &= \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+} |\widehat{\psi}(\alpha)|^2 \frac{d\alpha}{\alpha} d\xi.\end{aligned}$$

This is an important step as it eliminates the dependence on ξ in the integral, and allows us to consider the two integrals separately. If follows that if ψ satisfies the condition

$$\int_{\mathbb{R}^+} |\widehat{\psi}(\xi)|^2 \frac{1}{\xi} d\xi = 1, \tag{2.4}$$

then we obtain an isometry: $\|Wf\|^2 = \|f\|^2$ and $I = W^*W$. It is then easy to derive the following formula for the inverse

$$f(x) = \int_{\mathbb{R}^+}\int_{\mathbb{R}} Wf(\alpha, \beta) T_y D_\alpha \psi(x) \frac{d\alpha dy}{\alpha^2}. \tag{2.5}$$

The preceding argument is widely used in wavelet/time-frequency analysis and can be done in a more a general manner, cf. [20, Theorem 3.1], with connections to representation theory. The condition (2.4) is usually called the *admissibility condition* on ψ. It implies that $\widehat{\psi}(0) = 0$, i.e., that ψ has zero mean. It is clear that Gaussians do not satisfy this condition.

In time-frequency analysis, (translations of) functions of the form $\psi(x) = e^{-x^2 + 2\pi i\omega x}$ are often used; they are often referred to as *Gabor wavelets* [19]. Since the admissibility condition is not typically satisfied, Gabor wavelets do not define continuous wavelet transforms according to the standard definition. However, since modulated Gaussians are natural objects for signal analysis, they can be considered wavelets in the following slighly modified version:

$$\psi_{\text{Morlet}}(x) = \rho_\omega\, e^{-x^2}(e^{2\pi i\omega x} - \kappa_\omega), \tag{2.6}$$

where the constant κ_ω is chosen so that $\widehat{\psi}_{\text{Morlet}}(0) = 0$ and ρ_ω such that $\int \psi_{\text{Morlet}}(x)\, dx = 0$. The wavelets in (2.6) are called *Morlet wavelets*. In practical applications, the constant κ_ω is often discarded as it is more natural to work with purely modulated Gaussians. It also happens that κ_ω is often rather small (at least for large values of ω).

We will now follow the general strategy outlined above using the Gaussian wave packets considered in in this paper. We write a Gaussian wave packet in the form:

$$T_y M_\eta R_\theta D_{\alpha/|\eta|} K_\beta \varphi \quad \text{with} \quad \varphi(x) = \left(\frac{2\ln(16)}{\pi}\right)^{n/4} e^{-\ln(16)\sum_j^n x_j^2}. \tag{2.7}$$

In the one-dimensional case (2.7) reduces to

$$T_y M_\eta D_{\alpha/|\eta|}\varphi(x) = \sqrt{\frac{|\eta|}{\alpha}}\left(\frac{2\ln(16)}{\pi}\right)^{1/4} e^{-\ln(16)(\frac{(x-y)}{\alpha/\eta})^2 + 2\pi i\eta(x-y)}.$$

This motivates us to study objects of the form

$$U_{\lambda,\eta,y}\phi(x) = T_y M_\eta D_\lambda \phi(x) = \frac{1}{\sqrt{|\lambda|}} e^{-(\frac{(x-y)}{\lambda})^2 + 2\pi i\eta(x-y)}, \quad \phi(x) = e^{-x^2}.$$

To construct the group structure associated with $U_{\lambda,\eta,y}$, we consider

$$\begin{aligned}
&U_{\lambda_2,\eta_2,y_2} U_{\lambda_1,\eta_1,y_1}\phi = T_{y_2} M_{\eta_2} D_{\lambda_2} T_{y_1} M_{\eta_1} D_{\lambda_1}\phi = \\
&T_{y_2} M_{\eta_2} T_{\lambda_2 y_1} D_{\lambda_2} M_{\eta_1} D_{\lambda_1}\phi = \\
&P_{-\eta_2\lambda_2 y_1} T_{y_2+\lambda_2 y_1} M_{\eta_2+\eta_1/\lambda_2} D_{\lambda_2\lambda_1}\phi = P_{-\eta_2\lambda_2 y_1} U_{\lambda_2\lambda_1,\eta_2+\eta_1/\lambda_2,y_2+\lambda_2 y_1}\phi.
\end{aligned}$$

Due to the complex phase $P_{-\eta_2\lambda_2 y_1}$ it will be necessary to extend the group to include additional parameter. The group G associated with the operator $e^{2\pi i\tau} U_{\lambda,\eta,y}$ is defined as the set of quartets $(\lambda,\eta,y,\tau) \in \mathbb{R}^3 \times [0,1]$ with the group operation

$$(\lambda_2,\eta_2,y_2,\tau_2)\circ(\lambda_1,\eta_1,y_1,\tau_1) = \big(\lambda_1\lambda_2, \eta_1/\lambda_2 + \eta_2, \lambda_2 y_1 + y_2, (\tau_1+\tau_2-\eta_2\lambda_2 y_1) \mod 1\big),$$

unit element $(1,0,0,0)$, and inverse

$$(\lambda,\eta,y,\tau)^{-1} = (\lambda^{-1}, -\lambda\eta, -\lambda^{-1}y, -\eta y - \tau).$$

We now proceed to construct a (left-invariant) Haar measure as a weighted Lebesgue measure with some weight $v(\lambda,\eta,y,\tau)$. The Haar property says that for any $H \subset G$, we have that

$$\mu_{_L}(H) = \int_{_H} v(\lambda,\eta,y,\tau)\; d\lambda d\eta dy d\tau = \int_{(\lambda',\eta',y',\tau')\circ H} v(\lambda,\eta,y,\tau)\; d\lambda d\eta dy d\tau.$$

The integration over $(\lambda',\eta',y',\tau')\circ H$ can be carried out by a change of variables induced by the group operation. The functional matrix of the transformation is given by

$$\begin{pmatrix} \lambda' & 0 & 0 & 0 \\ 0 & 1/\lambda' & 0 & 0 \\ 0 & 0 & \lambda' & 0 \\ 0 & 0 & -\eta'\lambda' & 1 \end{pmatrix}$$

with Jacobian $J = \lambda'$. Hence,

$$\int_{_H} v(\lambda,\eta,y,\tau)\; d\lambda d\eta dy d\tau = \int_{_H} v(\lambda\lambda', \eta/\lambda'+\eta', \lambda' y + y', \tau+\tau'-\eta'\lambda' y)\lambda'\; d\lambda d\eta dy d\tau,$$

which gives $w(\lambda,\eta,y,\tau)=\lambda^{-1}$ as a possible weight. Since the Haar measure is unique up to a constant factor, we conclude that $d\mu_{_L}(\lambda,\eta,y,\tau)=\lambda^{-1}\,d\lambda d\eta dy d\tau$. The general framework suggests that we consider the following integral to determine the admissibility condition:

$$\begin{aligned}
&\int_{\mathbb{R}^+\times\mathbb{R}\times\mathbb{R}} |\langle U_{\lambda,\eta,y}\phi,f\rangle|^2\lambda^{-1}\,d\lambda d\eta dy = \int_{\mathbb{R}^+\times\mathbb{R}\times\mathbb{R}} |\langle T_y M_\eta D_\lambda\phi,f\rangle|^2\lambda^{-1}\,d\lambda d\eta dy\\
&= \int_{\mathbb{R}^+\times\mathbb{R}\times\mathbb{R}} |(M_\eta D_\lambda\phi * f(-\cdot))(y)|^2\lambda^{-1}\,d\lambda d\eta dy = \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}\times\mathbb{R}^+} |(FM_\eta D_\lambda\phi)(\xi)|^2\lambda^{-1}\,d\lambda d\eta d\xi\\
&= \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+\times\mathbb{R}} |(T_\eta D_{1/\lambda}\widehat{\phi})(\xi)|^2\lambda^{-1}\,d\lambda d\eta d\xi = \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}\times\mathbb{R}^+} |\widehat{\phi}(\lambda\xi-\eta)|^2 d\eta\,d\lambda d\xi\\
&= \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+} \|\widehat{\phi}\|^2\,d\lambda d\xi = \|\phi\|^2 \int_{\mathbb{R}} |f(\xi)|^2 \int_{\mathbb{R}^+} d\lambda d\xi = \infty,
\end{aligned}$$

which shows that there are no admissible functions in L^2. As we show below, this result will not prevent us from finding isometries and reconstruction formulas similar to (2.5). One just needs to find weights that will make the above calculation finite. This will also allow us to work with purely Gaussian functions, in contrast to, for example, Morlet wavelets. We derive the weights in the following theorem, which is proved for functions in $L^2(\mathbb{R}^n)$. Let S denote the normalized surface measure on the unit sphere S^{n-1} in $\mathbb{R}^n$.

THEOREM 2.1. *Let ϕ be as in (2.7) and $u:\mathbb{R}^+\times\mathbb{R}^+\times S^{n-1}\to\mathbb{R}^+$ a function that satisfies*

$$\int_{\mathbb{R}^+\times\mathbb{R}^+\times S^{n-1}} u(\lambda,\beta,\theta)\,d\lambda d\beta dS = 1,$$

and let $w(\alpha,\beta,\eta,\theta)=u(\alpha/|\eta|,\beta,\theta)/|\eta|$. Let the operator $\mathcal{W}:L^2(\mathbb{R}^n)\to L^2_w(\mathbb{R}^+\times\mathbb{R}^+\times S^{n-1}\times\mathbb{R}^n\times\mathbb{R}^n)$ be defined by

$$\mathcal{W}(f)(\alpha,\lambda,\theta,\eta,y)=\langle T_y M_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi,f\rangle. \tag{2.8}$$

Then $\mathcal{W}$ is an isometry and a reproducing formula for $f\in L^2(\mathbb{R}^n)$ is given by

$$f(x)=\int_{\mathbb{R}^+\times\mathbb{R}^n\times S^{n-1}\times\mathbb{R}^+\times\mathbb{R}^n} \mathcal{W}f(\alpha,\beta,\theta,\eta,y)\,T_y M_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi(x)\,w(\alpha,\beta,\eta,\theta)\,d\alpha d\eta dS d\beta dy.$$

Proof. We need to show that

$$\|\mathcal{W}f\|^2_{L^2_w}=\int_{\mathbb{R}^+\times\mathbb{R}^n\times S^{n-1}\times\mathbb{R}^+\times\mathbb{R}^n} |\langle T_y M_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi,f\rangle|^2 w(\alpha,\beta,\eta,\theta)\,d\alpha d\eta dS d\beta dy=\|f\|^2.$$

By writing the scalar product above as a convolution and applying Fubini and Plancherel's theorems, it follows that

$$\begin{aligned}
&\int_{\mathbb{R}^+\times\mathbb{R}^n\times S^{n-1}\times\mathbb{R}^+\times\mathbb{R}^n} |\langle T_y M_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi,f\rangle|^2\,w(\alpha,\beta,\eta,\theta)\,d\alpha d\eta dS d\beta dy =\\
&\int_{\mathbb{R}^n\times\mathbb{R}^n\times\mathbb{R}^+\times S^{n-1}\times\mathbb{R}^+} |\langle (M_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi) * f(-\cdot)(y)\rangle|^2\,w(\alpha,\beta,\eta,\theta)\,dy d\eta d\alpha dS d\beta =\\
&\int_{\mathbb{R}^n} |f(\xi)|^2 \int_{\mathbb{R}^+\times\mathbb{R}^n\times S^{n-1}\times\mathbb{R}^+} |(FM_\eta R_\theta D_{\alpha/|\eta|}K_\beta\phi)(\xi)|^2\,w(\alpha,\beta,\eta,\theta)\,d\alpha d\eta dS d\beta d\xi =\\
&\int_{\mathbb{R}^n} |f(\xi)|^2 J(\xi)\,d\xi.
\end{aligned}$$

By the change of variables $\{\lambda = \alpha/|\eta|, d\alpha = |\eta| d\lambda\}$ and using the commutation rules for the Fourier transform we obtain

$$J(\xi) = \int_{\mathbb{R}^+ \times \mathbb{R}^n \times S^{n-1} \times \mathbb{R}^+} |(T_\eta R_\theta F D_\lambda K_\beta \phi)(\xi)|^2 \; u(\lambda, \beta, \theta) \; d\lambda d\eta dS d\beta =$$
$$\int_{\mathbb{R}^n \times \mathbb{R}^+ \times S^{n-1} \times \mathbb{R}^+} |(F D_\lambda K_\beta \phi)(\Theta(\xi - \eta))|^2 \; d\eta \; u(\lambda, \beta, \theta) d\lambda dS d\beta =$$
$$\int_{\mathbb{R}+ \times S^{n-1} \times \mathbb{R}^+} \|F D_\lambda K_\beta \phi\|^2_{L^2(\mathbb{R}^n)} u(\lambda, \beta, \theta) \; d\lambda dS d\beta.$$

But since F, D_λ and K_β are unitary, we have $\|F D_\lambda K_\beta \phi\|^2 = \|\phi\| = 1$, and therefore

$$J(\xi) = \|\phi\|^2 \int_{\mathbb{R}^+ \times S^{n-1} \times \mathbb{R}^+} u(\lambda, \beta, \theta) \; d\lambda dS d\beta = 1,$$

by the assumption on w and the choice of ϕ. Hence $\mathcal{W}$ is an isometry and the reproducing formula follows immediately. □

2.1. Decay rates. The Gaussian wave packets will exhibit superior decay rates compared to many of the competing representations. To illustrate this we provide a comparison of Gaussian wave packets and curvelets. We have chosen to compare against curvelets because it is one of the most popular methods that incorporates multi-scale and multi-orientation structure. A major drawback of curvelets, however, is that they have large tails. Hence, the scalar product between curvelets can be quite large even if the curvelets are fairly far apart. This implies that it will be necessary to use many curvelets in a vicinity of the actual wave front in order to cancel tail artifacts introduced by the curvelets. The extent of the tails is in principle caused by the comparatively sharp cut-off in the Fourier domain (although curvelets can also be defined with $\mathcal{C}^\infty$ functions). A smoother cut-off causes a more rapid decay in the spatial domain. This is illustrated in Figure 3.

The top panel in Figure 3 shows contour plots of four Gaussian wave packets (left) and four curvelet packets (right) with similar parameters. The bottom panel shows the same packets in the frequency domain. For illustrative purposes, we have normalized the wave packets so that their individual maximum absolute value is one in both domains. The window functions for the curvelets are designed to have compact support in the frequency domain. For each window function, a rectangular domain that inscribes the (frequency) support of the window function is selected, a Fourier series is applied on that to construct associated *curvelets* translations. We note that the part in frequency that causes most of the slow decay in space for curvelets, is the wedge-like structure closest to the origin, since the cut-off is sharpest there. This feature is due to the use of a tensor description in polar coordinates. While such a tensor product can yield a C^∞ window function, sharp features in the frequency domain will cause slow decay in the spatial domain. We note that Gaussian windows provide optimal decay in both domains [19].

As shown in [1], it is possible to obtain improved decay in the spatial domain, without loosing any of the required frequency coverage; that is, without increasing the rectangular domain on which a Fourier series expansion is used. Hence, it would be more appropriate to compare the performance of Gaussian wave packets to that of wave packets constructed in [1], which obey the same dyadic parabolic scaling law of the curvelets designed in [10]. We will, however, make numerical comparisons with the implementations of [10] as it is considered to be the standard implementation.

3. Discrete Gaussian wave packet transforms. In this section, we will develop numerical methods for fast implementation of a decomposition similar to the recomposition-formula following (2.8), in the two-dimensional case. Since the research is motivated by concrete applications, we need to represent functions in a finite dimensional setting. Hence we will only consider representations of functions with compact support and to simplify the presentation the support of the functions will

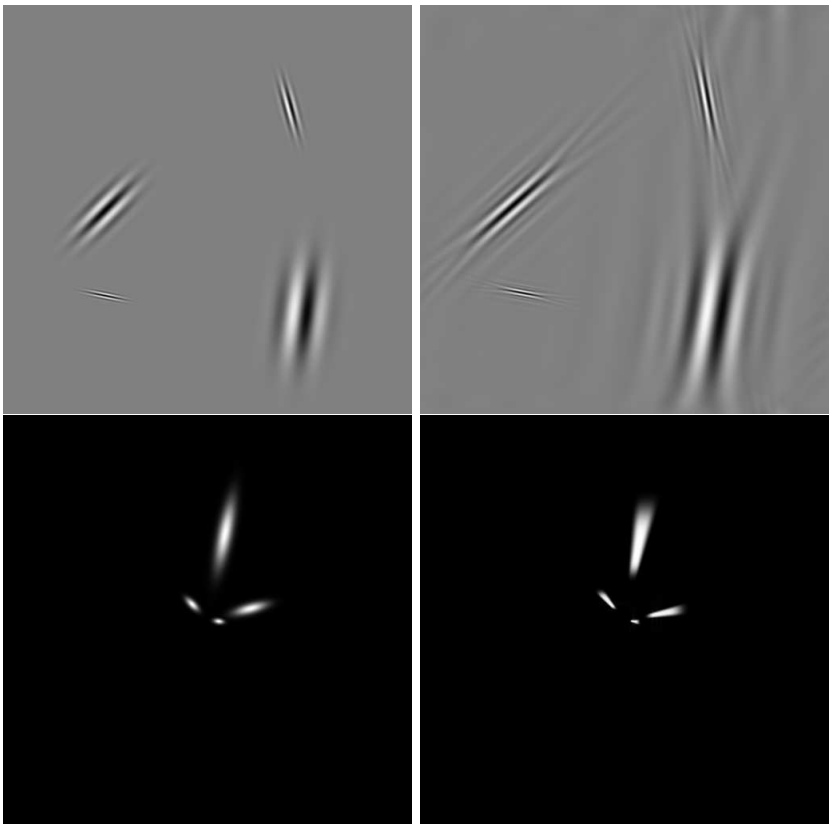

FIG. 3. *Comparison between a few curvelets and Gaussian wave packets with similar parameters. The panels to the left display Gaussian wave packet while the right panel display curvelets with a similar behaviour, with the spatial domain being depicted in the top panels and the frequency in the bottom ones.*

be assumed to be contained in the unit disc. We shall use the notation

$$L^\infty(\mathbb{D}) = \left\{ f \in L^\infty \, : \, \mathrm{supp}(f) \subset \mathbb{D} = \{x \in \mathbb{R}^2 : |x| \leq 1/2\} \right\}.$$

This section aims at giving estimates for the errors introduced by the finite dimensional representations.

We start by proving some ancillary results that will be used in the discretization of Gaussian wave packets. In the following lemmas we use the Gaussian function $\varphi_\lambda(x) = \sqrt{\lambda/\pi}\, e^{-\lambda x^2}$ for a fixed $\lambda > 0$. For a given $\epsilon > 0$, we define the following constants whose meaning will be clarified below: $K_\epsilon > 0$ will denote an even integer such that $\widehat{\varphi}_\lambda(K_\epsilon/2) \leq \epsilon$, and $\sigma_\epsilon > 0$ such that $\sigma_\epsilon \geq \sqrt{-\ln(\epsilon/K_\epsilon)/\lambda}$ and $\sigma_\epsilon K_\epsilon \in \mathbb{N}$. In particular, $K_\epsilon(1/2 + \sigma_\epsilon) \in \mathbb{N}$. These constants of course depend on λ as well.

LEMMA 3.1. *Let $f \in L^1(\mathbb{R})$ and $\epsilon \in (0, 1/2)$. Then,*

$$\sup_{|\xi| \leq K_\epsilon/2} \left| \sum_j f * \varphi_\lambda(j/K_\epsilon)\, e^{-2\pi i j \xi/K_\epsilon} - \widehat{f}(\xi)\widehat{\varphi}_\lambda(\xi) \right| \leq 4\epsilon \, \| \, \widehat{f} \, \|_\infty.$$

Proof. By the Poisson summation formula applied to $f * \varphi_\lambda$,

$$\sum_j \widehat{f}(\xi + jK_\epsilon)\widehat{\varphi}_\lambda(\xi + jK_\epsilon) = \sum_j f * \varphi_\lambda \left(j/K_\epsilon\right) e^{-2\pi i j \xi/K_\epsilon}$$

for any $\xi \in \mathbb{R}$. It follows that

$$\sup_{|\xi|\le K_\epsilon/2}\left|\sum_j f*\varphi_\lambda(j/K_\epsilon)\,e^{-2\pi ij\xi/K_\epsilon}-\widehat{f}(\xi)\widehat{\varphi}_\lambda(\xi)\right|=\sup_{|\xi|\le K_\epsilon/2}\left|\sum_{j\neq 0}\widehat{f}(\xi+jK_\epsilon)\widehat{\varphi}_\lambda(\xi+jK_\epsilon)\right|.$$

However, it is straightforward to derive the following estimate

$$\begin{aligned}\sup_{|\eta|\le K_\epsilon/2}\left|\sum_{j\neq 0}\widehat{f}(\eta+jK_\epsilon)\widehat{\varphi_\lambda}(\eta+jK_\epsilon)\right| &\le \|\,\widehat{f}\,\|_\infty \sup_{|\eta|\le K_\epsilon/2}\left|\sum_{j\neq 0}\widehat{\varphi}_\lambda(\eta+jK_\epsilon)\right|\\ &\le \|\,\widehat{f}\,\|_\infty \sum_{j\neq 0}\epsilon^{4(j+1/2)^2}\le 4\epsilon\,\|\,\widehat{f}\,\|_\infty,\end{aligned}$$

where in the second inequality we have used the definition of K_ϵ. □

LEMMA 3.2. *Let $f\in L^\infty(\mathbb{R})$ with $\mathrm{supp}(f)\in[-1/2,1/2]$. Then, for any $\epsilon\in(0,1/2)$*

$$\sup_\xi\left|\sum_{|j|>K_\epsilon(1/2+\sigma_\epsilon)} f*\varphi_\lambda(j/K_\epsilon)\,e^{-2\pi ij\xi}\right|<\epsilon\,\|f\|_\infty.$$

Proof. By separating the sum over $|j|>K_\epsilon(1/2+\sigma_\epsilon)$ into its positive and negative parts, it follows that

$$\begin{aligned}&\sup_\xi\left|\sum_{|j|>K_\epsilon(1/2+\sigma_\epsilon)} f*\varphi_\lambda(j/K_\epsilon)\,e^{-2\pi ij\xi}\right|\le 2\,\|f\|_\infty\sum_{j>K_\epsilon(1/2+\sigma_\epsilon)}\int_{-1/2}^{1/2}\varphi_\lambda(j/K_\epsilon-y)\,dy\\ &\le 2\,\|f\|_\infty\sum_{j>0}\int_{j/K_\epsilon}^{j/K_\epsilon+1}\varphi_\lambda(\sigma_\epsilon+y)\,dy\le 2\,\|f\|_\infty\,e^{-\lambda{\sigma_\epsilon}^2}\sum_{j>0}\int_{j/K_\epsilon}^{j/K_\epsilon+1}\varphi_\lambda(y)\,dy.\\ &\le 2\,\|f\|_\infty\,e^{-\lambda{\sigma_\epsilon}^2}\sum_{j>0}\varphi_\lambda(j/K_\epsilon)\le 2\,\|f\|_\infty\,e^{-\lambda{\sigma_\epsilon}^2}K_\epsilon\int_0^\infty\varphi_\lambda(y)\,dy\\ &=\|f\|_\infty\,e^{-\lambda{\sigma_\epsilon}^2}K_\epsilon\le\epsilon\,\|f\|_\infty.\end{aligned}$$

□

A consequence of Lemma 3.2 and Lemma 3.1 is:

LEMMA 3.3. *Let $f\in L^\infty(\mathbb{R})$ with $\mathrm{supp}(f)\in[-1/2,1/2]$. Then, for any $\epsilon\in(0,1/2)$*

$$\sup_{|\xi|\le K_\epsilon/2}\left|\widehat{f}(\xi)\widehat{\varphi}_\lambda(\xi)-\sum_{|j|\le K_\epsilon(1/2+\sigma_\epsilon)} f*\varphi_\lambda(j/K_\epsilon)\,e^{-2\pi ij\xi/K}\right|\le 5\epsilon\,\|f\|_\infty.$$

We define the set of integers $\mathbb{Z}_{K,\sigma}$ as

$$\mathbb{Z}_{K,\sigma}=\{\,j\in\mathbb{Z}\,:\,-K_\epsilon(1/2+\sigma_\epsilon)\le j<K_\epsilon(1/2+\sigma_\epsilon)\,\}\,. \tag{3.1}$$

Moreover, for any discrete set I we use $|I|$ to denote the number of elements of I. In particular, $|\mathbb{Z}_{K,\sigma}|=2\,K_\epsilon(1/2+\sigma_\epsilon)$.

LEMMA 3.4. *Let $f\in L^\infty(\mathbb{R})$ with $\mathrm{supp}(f)\in[-1/2,1/2]$. Then for any $\epsilon\in(0,1/2)$ and $j\in\mathbb{Z}_{K,\sigma}$*

$$\mid Q_{\lambda,\mathbb{Z}_{K,\sigma}}[f](j/K_\epsilon)\mid<5\epsilon\,\|f\|_\infty,$$

where

$$Q_{\lambda,\mathbb{Z}_{K,\sigma}}[f](j/K_\epsilon) = f * \varphi_\lambda(j/K_\epsilon) - \frac{1}{|\mathbb{Z}_{K,\sigma}|} \sum_{m\in\mathbb{Z}_{K,\sigma}} \widehat{f}(\xi_m)\widehat{\varphi}_\lambda(\xi_m)\, e^{2\pi i \xi_m j/K_\epsilon}$$

and $\xi_m = m/(2\sigma_\epsilon + 1)$.

Proof. It follows from the orthogonality of the discrete Fourier transform that

$$\begin{aligned} f * \varphi_\lambda(j/K_\epsilon) &= f * \varphi_\lambda(j/K_\epsilon)\, \frac{1}{|\mathbb{Z}_{K,\sigma}|} \sum_{m,n\in\mathbb{Z}_{K,\sigma}} e^{2\pi i m(j-n)/(K_\epsilon + 2\sigma_\epsilon K_\epsilon)} \\ &= \frac{1}{|\mathbb{Z}_{K,\sigma}|} \sum_{m,n\in\mathbb{Z}_{K,\sigma}} f * \varphi_\lambda(j/K_\epsilon) e^{2\pi i \xi_m (j-n)/K_\epsilon} \end{aligned}$$

Hence,

$$\begin{aligned} &\frac{1}{|\mathbb{Z}_{K,\sigma}|} \left| \sum_{m\in\mathbb{Z}_{K,\sigma}} \widehat{f}(\xi_m)\widehat{\varphi}_\lambda(\xi_m)\, e^{2\pi i \xi_m j/K_\epsilon} - f * \varphi_\lambda(j/K_\epsilon) \right| = \\ &\frac{1}{|\mathbb{Z}_{K,\sigma}|} \left| \sum_{m\in\mathbb{Z}_{K,\sigma}} \left(\widehat{f}(\xi_m)\widehat{\varphi}_\lambda(\xi_m) - \sum_{n\in\mathbb{Z}_{K,\sigma}} f * \varphi_\lambda(j/K_\epsilon)\, e^{-2\pi i \xi_m n/K_\epsilon} \right) e^{2\pi i \xi_m j/K_\epsilon} \right| \leq \\ &\frac{1}{|\mathbb{Z}_{K,\sigma}|} \sum_{m\in\mathbb{Z}_{K,\sigma}} \left| \widehat{f}(\xi_m)\widehat{\varphi}_\lambda(\xi_m) - \sum_{n\in\mathbb{Z}_{K,\sigma}} f * \varphi_\lambda(j/K_\epsilon)\, e^{-2\pi i \xi_m n/K_\epsilon} \right| \leq 5\epsilon\, \|f\|_\infty, \end{aligned}$$

where we have used Lemma 3.3 in the last step. □

In our discretization we will make use of both spatial and frequency lattices, generalizing (3.1). To simplify notation, for a given $\epsilon > 0$ and a pair $\lambda_1, \lambda_2 > 0$, we shall write their corresponding parameters $\{K_\epsilon, \sigma_\epsilon\}$ simply as $\{K_1, \sigma_1\}$ for φ_{λ_1} and $\{K_2, \sigma_2\}$ for φ_{λ_2}.

The spatial lattice is defined as

$$X_{\theta,K,\sigma} = \left\{ \Theta^{\mathsf{t}} \begin{pmatrix} j_1/K_1 \\ j_2/K_2 \end{pmatrix} : j_1 \in \mathbb{Z}_{K_1,\sigma_1}, \quad j_2 \in \mathbb{Z}_{K_2,\sigma_2} \right\},$$

while the frequency lattice is

$$\Xi_{\eta,\theta,K,\sigma} = \left\{ \Theta^{\mathsf{t}} \left(|\eta| e_1 + \begin{pmatrix} m_1/(2\sigma_1+1) \\ m_2/(2\sigma_2+1) \end{pmatrix} \right) : m_1 \in \mathbb{Z}_{K_1,\sigma_1}, \quad m_2 \in \mathbb{Z}_{K_2,\sigma_2} \right\}.$$

There is a bijection between the sets $X_{\theta,K,\sigma}$ or $\Xi_{\eta,\theta,K,\sigma}$ and $\mathbb{Z}_{K_1,\sigma_1} \times \mathbb{Z}_{K_2,\sigma_2}$. In our notation, it will be understood that if we write $\xi_m \in \Xi_{\eta,\theta,K,\sigma}$, then $m \in \mathbb{Z}_{K_1,\sigma_1} \times \mathbb{Z}_{K_2,\sigma_2}$ and likewise for elements in $X_{\theta,K,\sigma}$.

The following theorem shows that the inner product $f_\gamma = \langle f, \varphi_\gamma \rangle$ can be approximated by a quantity $\widetilde{f}_\gamma$ that can be efficiently computed using the FFT.

THEOREM 3.5. *Let $f \in L^\infty(\mathbb{D})$ and choose the constants $\epsilon \in (0, 1/2)$, $\alpha, \beta > 0$ and $\eta \in \mathbb{R}^2$. Define the Gaussian parameters $\lambda_1 = \ln(16)\|\eta\|^2/\alpha^2$ and $\lambda_2 = \lambda_1/\beta^2$. Then for any $y = \Theta^{\mathsf{t}} (j_1/K_1 \;\; j_2/K_2)^t \in X_{\theta,K,\sigma}$*

$$| f_\gamma - \widetilde{f}_\gamma | \leq \frac{20 \left(1 + \sqrt{\lambda_1/\pi} \right)}{\rho_{\alpha,\beta,\eta}} \epsilon \|f\|_\infty, \tag{3.2}$$

where $\gamma = (y, \eta, \alpha, \beta, \theta)$ *and*

$$\widetilde{f}_\gamma = \frac{e^{2\pi i \eta^{\mathsf{t}} y}}{|\mathbb{Z}_{1,2}|} \sum_{m \in \mathbb{Z}_{1,2}} \widehat{f}(\xi_m) \widehat{\varphi}_{\gamma_0}(\xi_m)\, e^{2\pi i \left(j_1 m_1 / |\mathbb{Z}_{K_1,\sigma_1}| + j_2 m_2 / |\mathbb{Z}_{K_2,\sigma_2}| \right)},$$

where $\mathbb{Z}_{1,2} = \mathbb{Z}_{K_1,\sigma_1} \times \mathbb{Z}_{K_2,\sigma_2}$ *and* $\gamma_0 = (\mathbf{0}, \eta, \alpha, \beta, \theta)$.

Proof. Using the change of variable $u = -\Theta(x + y)$, we can rewrite the scalar product f_γ as

$$\begin{aligned} f_\gamma &= \rho_{\alpha,\beta,\eta} \int_{\mathbb{R}^2} f(x)\, e^{-2\pi i \eta^t T_y(x)}\, e^{-T_y(x)^t \Theta^t \Lambda(\alpha,\beta,\eta) \Theta T_y(x)}\, dx \\ &= \rho_{\alpha,\beta,\eta} \int_{\mathbb{R}^2} f(y - \Theta^{\mathsf{t}} u)\, e^{2\pi i (\Theta\eta)^{\mathsf{t}} u}\, e^{-u^{\mathsf{t}} \Lambda(\alpha,\beta,\eta) u}\, du \end{aligned}$$

By introducing the function

$$g(x) = f(\Theta^{\mathsf{t}} x) e^{-2\pi i (\Theta\eta)^{\mathsf{t}} x},$$

and defining the two-dimensional Gaussian $\varphi_{\lambda_1,\lambda_2}(x) = \varphi_{\lambda_1}(x_1)\, \varphi_{\lambda_2}(x_2)$, we may now write f_γ as a modulated convolution:

$$\begin{aligned} f_\gamma &= \rho_{\alpha,\beta,\eta}\, e^{2\pi i \eta^{\mathsf{t}} y} \int_{\mathbb{R}^2} g(\Theta y - x) e^{-x^{\mathsf{t}} \Lambda(\alpha,\beta,\eta) x} dx \\ &= \frac{2}{\rho_{\alpha,\beta,\eta}}\, g * \varphi_{\lambda_1,\lambda_2}(\Theta y)\; e^{2\pi i \eta^{\mathsf{t}} y}. \end{aligned} \tag{3.3}$$

Define the vectors $z = \Theta y = (j_1/K_1\ j_2/K_2)^{\mathsf{t}}$ and $\zeta_m = (\zeta_{m_1}\ \zeta_{m_2})^{\mathsf{t}}$ with

$$\zeta_{m_1} = m_1/(2\sigma_1 + 1), \quad \zeta_{m_2} = m_2/(2\sigma_2 + 1) \tag{3.4}$$

and the auxiliary functions

$$u_{z_1}(x_2) = g(\,\cdot\,, x_2) * \varphi_{\lambda_1}(z_1) = \int_{\mathbb{R}} g(x_1, x_2) \varphi_{\lambda_1}\,(z_1 - x_1)\; dx_1,$$

and

$$v_{m_2}(x_1) = \widehat{g}(x_1, \zeta_{m_2}) = \int_{\mathbb{R}} g(x_1, x_2)\, e^{-2\pi i x_2\, \zeta_{m_2}}\; dx_2.$$

Note that

$$\widehat{u}_{z_1}(\zeta_{m_2}) = v_{m_2} * \varphi_{\lambda_1}\,(z_1) \quad \text{and} \quad g * \varphi_{\lambda_1,\lambda_2}\,(z) = u_{z_1} * \varphi_{\lambda_2}\,(z_2)\,.$$

In addition, the Fourier transforms of g and φ_{γ_0} can be written as

$$\widehat{g}(\zeta_m) = \widehat{v}_{m_2}(\zeta_{m_1}) = \widehat{f}(\xi_m) \quad \text{and} \quad \widehat{\varphi}_{\gamma_0}(\xi_m) = \frac{2}{\rho_{\alpha,\beta,\eta}}\, \widehat{\varphi}_{\lambda_1,\lambda_2}(\zeta_m)$$

for $\xi_m = \Theta^{\mathsf{t}}(|\eta| e_1 + \zeta_m) \in \Xi_{\eta,\theta,K,\sigma}$. Note also that we can write

$$\left(\frac{j_1 m_1}{|\mathbb{Z}_{K_1,\sigma_1}|}, \frac{j_2 m_2}{|\mathbb{Z}_{K_2,\sigma_2}|} \right) = \left(\frac{j_1\, \zeta_{m_1}}{K_1}, \frac{j_2\, \zeta_{m_2}}{K_2} \right) = (z_1\, \zeta_{m_1},\ z_2\, \zeta_{m_2}).$$

Hence we can rewrite $\widetilde{f}_\gamma$ as

$$\widetilde{f}_\gamma = \frac{2\, e^{2\pi i \eta^{\mathsf{t}} y}}{\rho_{\alpha,\beta,\eta}}\, \frac{1}{|\mathbb{Z}_{1,2}|} \sum_m \widehat{g}(\zeta_m)\, \widehat{\varphi}_{\lambda_1,\lambda_2}(\zeta_m)\, e^{2\pi i\, z^{\mathsf{t}} \zeta_m},$$

and since $e^{-2\pi i\eta^{\mathrm{t}}y}$ enters both f_γ and $\widetilde{f}_\gamma$ as a multiplicative factor, using the triangle inequality and $|\widehat{\varphi}_{\lambda_2}| \leq 1$ we obtain

$$\begin{aligned}
\frac{\rho_{\alpha,\beta,\eta}}{2} \,|\, f_\gamma - \widetilde{f}_\gamma \,|\, &= \left| g * \varphi_{\lambda_1,\lambda_2}(z) - \frac{1}{|\mathbb{Z}_{1,2}|} \sum_{m\in\mathbb{Z}_{1,2}} \widehat{g}(\zeta_m)\, \widehat{\varphi}_{\lambda_1,\lambda_2}(\zeta_m)\, e^{2\pi i\zeta^{\mathrm{t}}z} \right| \\
&\leq \left|Q_{\lambda_2,\mathbb{Z}_{K_2,\sigma_2}}[u_{z_1}](z_2)\right| + \left| \frac{1}{|\mathbb{Z}_{K_2,\sigma_2}|} \sum_{m_2\in\mathbb{Z}_{K_2,\sigma_2}} \widehat{\varphi}_{\lambda_2}(\zeta_{m_2})\, e^{2\pi i\zeta_{m_2}z_2}\, Q_{\lambda_1,\mathbb{Z}_{K_1,\sigma_1}}[v_{m_2}](z_1) \right| \\
&\leq \left|Q_{\lambda_2,\mathbb{Z}_{K_2,\sigma_2}}[u_{z_1}](z_2)\right| + \left|\, Q_{\lambda_1,\mathbb{Z}_{K_1,\sigma_1}}[v_{m_2}](z_1) \,\right|
\end{aligned}$$

Applying Lemma 3.4 to the two terms on the right-hand side yields the final inequality

$$\begin{aligned}
\frac{\rho_{\alpha,\beta,\eta}}{2} \,|\, f_\gamma - \widetilde{f}_\gamma \,|\, &\leq 5\epsilon \,\|u_{z_1}\|_\infty + 5\epsilon \,\|v_{m_2}\|_\infty \\
&\leq 10\epsilon \left(1 + \sqrt{\lambda_1/\pi}\right) \,\|f\|_\infty.
\end{aligned}$$

□

We now proceed with constructing a tiling of the frequency domain. Given that we want to work with approximations that are accurate within a frequency domain $\Xi \subset \mathbb{R}^2$, we choose a set

$$\Omega = \{(\eta_j, \alpha_j, \beta_j, \theta_j)\}_j \tag{3.5}$$

for a finite set of j such that for a chosen $r > 0$

$$\sum_{(\eta,\alpha,\beta,\theta)\in\Omega} |\widehat{\varphi}_{0,\eta,\alpha,\beta,\theta}(\xi)|^2 > r, \quad \forall \xi \in \Xi. \tag{3.6}$$

As a motivation for this choice, we note that one way to find a representation of f is to (formally)

Note that the set Ω may vary substantially depending on the structure of the functions whose sparse decompositions we want to find, and it may be quite redundant in the sense that there may be a wide overlap of the supports of the functions $\widehat{\varphi}_{0,\eta,\alpha,\beta,\theta}$. Given Ω, we choose a precision parameter $\epsilon > 0$ and define the sets Ω_ϵ and Γ_ϵ as follows:

$$\begin{aligned}
\Omega_\epsilon = \Big\{ &(\eta,\alpha,\beta,\theta,K,\sigma) : (\eta,\alpha,\beta,\theta) \in \Omega,\ (\lambda_1,\lambda_2) = \ln(16)|\eta|^2(1, 1/\beta^2)/\alpha^2, \\
&\widehat{\varphi}_{\lambda_1}(K_1/2) < \epsilon, \quad \widehat{\varphi}_{\lambda_2}(K_2/2) < \epsilon, \quad \varphi_{\lambda_1}(\sigma_1) < \epsilon/K_1, \quad \varphi_{\lambda_2}(\sigma_2) < \epsilon/K_2 \Big\}
\end{aligned}$$

To each element in Ω_ϵ we associate a lattice $X_{\theta,K,\sigma}$, and form the *wave packet parameter set*

$$\Gamma_\epsilon = \{\gamma = (y,\eta,\alpha,\beta,\theta) : y \in X_{\theta,K,\sigma}, \quad (\eta,\alpha,\beta,\theta,K,\sigma) \in \Omega_\epsilon\},$$

which defines the discrete set of Gaussian wave packets φ_γ, $\gamma \in \Gamma_\epsilon$ that will be used to represent functions. We define the synthesis operator $\mathcal{V} : \ell^2(\Gamma_\epsilon) \mapsto L^2(\mathbb{R}^2)$ as

$$\mathcal{V}c = \sum_{\gamma\in\Gamma_\epsilon} c_\gamma \varphi_\gamma. \tag{3.7}$$

and the analysis operator $\mathcal{V}^* : L^2(\mathbb{R}^2) \mapsto \ell^2(\Gamma_\epsilon)$

$$\mathcal{V}^* f = (\,\langle f, \varphi_\gamma \rangle)_{\gamma\in\Gamma_\epsilon}. \tag{3.8}$$

By Theorem 3.5 we know that for a fixed element $(\eta, \alpha, \beta, \theta, K, \sigma) \in \Omega_\epsilon$ and for $f \in L^\infty(\mathbb{D}) \subset L^2(\mathbb{R}^2)$, we can efficiently approximate $f_\gamma = \langle f, \varphi_{y,\eta,\alpha,\beta,\theta} \rangle$ for $y \in X_{\theta,K,\sigma}$ by means of FFT provided the values $\widehat{f}(\xi_m)$, $\xi_m \in \Xi_{\eta,\theta,K,\sigma}$, are known. If f is sufficiently densely sampled on a equally spaced lattice, then $\widehat{f}(\xi_m)$ can be rapidly evaluated at all the points $\xi_m \in \Xi_{\eta,\theta,K,\sigma}$ by employing USFFT algorithms [6, 15]. These algorithms have the same time complexity as ordinary FFT algorithms but with worse complexity constants.

In the evaluation of the approximation of f_γ for $y \in X_{\eta,\theta,K,\sigma}$ through Theorem 3.5, the computation of $\widehat{f}(\xi_m)$, $\xi_m \in \Xi_{\eta,\theta,K,\sigma}$ would constitute a dominant part of the computational time. However, it is possible to reduce such computational cost by choosing the the parameters K_i and σ_i so that the values of $\widehat{f}(\xi_m)$ can be reused. It thus makes sense to have the same set of values (η, α, β) for a fixed direction θ and restrict to sets Ω in (3.5) of the form

$$\Omega = \{(\eta_{k,\ell}, \alpha_{k,\ell}, \beta_{k,\ell}, \theta_l)\}_{k,\ell}, \tag{3.9}$$

where for each fixed ℓ we choose (common) parameters K_i^ℓ and σ_i^ℓ such that for every $(\eta_{k,\ell}, \alpha_{k,\ell}, \beta_{k,\ell}, \theta_\ell) \in \Omega$

$$\widehat{\varphi}_{\lambda_1}(K_1^\ell/2) < \epsilon, \;\; \widehat{\varphi}_{\lambda_2}(K_2^\ell/2) < \epsilon, \;\; \varphi_{\lambda_1}(\sigma_1^\ell) < \epsilon/K_1^\ell, \;\; \varphi_{\lambda_2}(\sigma_2^\ell) < \epsilon/K_2^\ell,$$

where $\lambda_1 = \ln(16)\,|\eta_{k,\ell}|^2/\alpha_{k,\ell}^2$ and $\lambda_2 = \ln(16)\,|\eta_{k,\ell}|^2/\alpha_{k,\ell}^2\,\beta_{k,\ell}^2$. Given this choice, it is natural to define

$$\Omega_\epsilon^{K^\ell,\sigma^\ell} = \left\{(\eta_{k,\ell}, \alpha_{k,\ell}, \beta_{k,\ell}, \theta_{k,\ell}, K^{k,\ell}, \sigma^\ell) : (\eta_{k,\ell}, \alpha_{k,\ell}, \beta_{k,\ell}, \theta_\ell) \in \Omega, \;\; K^{k,l} \leq K^\ell \right\}.$$

Now, given that the values $\eta_{k,l}$ has been chosen such that there exists a η_l such that $|\eta_l| - \sigma_l|\eta_{k,l}| \in \mathbb{Z}$ then we have that

$$\Xi_{\eta_{k,l},\theta,K^{k,\ell},\sigma^\ell} \subset \Xi_{\eta_l,\theta,K^\ell,\sigma^\ell}. \tag{3.10}$$

This implies that it is only necessary to evaluate $\widehat{f}(\xi_m^\ell)$, $\xi_m^\ell \in \Xi_{\eta,\theta,K^\ell,\sigma^\ell}$, to obtain all the values $\widehat{f}(\xi_m^{k,\ell})$, $\xi_m^{k,\ell} \in \Xi_{\eta,\theta,K^{k,\ell},\sigma^\ell}$ that are required for rapidly approximating the application of the analysis operator $\mathcal{V}^*$ in (3.7).

4. Sparse and redundant representations. Given a function $f \in L^\infty(\mathbb{D})$ and a collection of wave packets $\Gamma = \Gamma_\epsilon$, we want to find an element $c \in \ell^2(\Gamma)$ such that $\mathcal{V}c \approx f$ (see (3.7)) with few non-zero elements $c(\gamma)$, $\gamma \in \Gamma$. As noted in the Introduction, a popular way to obtain such a *sparse* representation is by minimizing an L^2 data-misfit with an ℓ^1 penalty. The coefficients c are thus obtained by solving the quadratic programming problem

$$\operatorname{argmin}_{c\in\ell^2(\Gamma)} \frac{1}{2}\|\mathcal{V}c - f\|_2^2 + \mu\|c\|_1 = \operatorname{argmin}_{c\in\ell^2(\Gamma)} \frac{1}{2}c^*\mathcal{V}^*\mathcal{V}c - \Re(c^*\mathcal{V}^*f) + \mu\|c\|_1. \tag{4.1}$$

The problem (4.1) is sometimes referred to as the *LASSO* problem, [31]. The wave packet parameter set Γ will typically give rise to rather redundant representations. This is useful from the point that it provides a wider selection of $c \in \ell^2(\Gamma_\epsilon)$ for which $\mathcal{V}c \approx f$, which increases our chances of finding a c with few non-zero elements such that $\mathcal{V}c \approx f$. Since Γ is a discrete set but the space $L^\infty(\mathbb{D})$ we want to approximate (accurately within some frequency domain Ξ, cf. (3.6)), is not, we will speak about redundancy not in terms of the usual ratio between the number of "basis" functions at hand and the dimension of the space to be approximated, but rather in terms of the ratio between the number of "basis" functions at hand and the (numerical) rank of the Gram matrix

$$\mathcal{G}_{\gamma,\gamma'} = \langle \varphi_\gamma, \varphi_{\gamma'} \rangle, \quad \gamma, \gamma' \in \Gamma_\epsilon.$$

We note that $\mathcal{G}$ can also be obtained by the product

$$\mathcal{G} = \mathcal{V}^*\mathcal{V}. \tag{4.2}$$

The Gram matrix will play a crucial role in our framework.

The problem of developing computationally efficient algorithms to solve quadratic programing problems such as (4.1) has received much attention in recent years [14, 16, 30]. The simplest algorithm seems to be *iterative soft thresholding*, which is defined as follows: Let $\mathrm{S}_\mu(x) = \mathrm{sign}(x)(|x| - \mu)_+$ be the soft thresholding operator. It is shown in [14], that for problems of the form (4.1), the iterative procedure (with S_μ applied component wise)

$$c^{n+1} = \mathrm{S}_\mu(c^n - \mathcal{V}^*\mathcal{V}c^n + \mathcal{V}^* f), \tag{4.3}$$

will converge to the solution $c^\dagger$ of (4.1) as $n \to \infty$ provided $\|\mathcal{V}^*\mathcal{V}\| \leq 1$. If $\|\mathcal{V}^*\mathcal{V}\| > 1$ then (4.3) can be easily modified by incorporating a normalization factor in $\mathcal{V}$. Other popular methods are based on iterative reweighted least squares, see [13] and references therein. The idea is the following: Let $c^\dagger$ be the solution to (4.1) for a fixed μ. If $c^\dagger$ would have been known, we could choose weights $w_j = \mu/|c_j^\dagger|$ (or practically as $w_j = \mu(|c_j^\dagger|^2 + \epsilon^2)^{-1/2}$, for some precision parameter ϵ) and consider the solution of the penalized weighted least square problem

$$\min_{c \in \ell^2(\Gamma_\epsilon)} \frac{1}{2}\|\mathcal{V}c - f\|_2^2 + \|c\|_w^2,$$

which by variational calculus is seen to have the solution

$$(\mathcal{G} + \mathrm{diag}(w))c = \Re(\mathcal{V}^* f).$$

Now, we do not know how to choose the weights w_j, since $c^\dagger$ is not known. But it turns out that it is possible to obtain it (under some conditions) through the following iterative procedure: This

Algorithm 1 Iteratively reweighted least squares for the LASSO problem (4.1)

1: Let $w_j^0 = \mu$.
2: **for** $k \geq 0$ **do**
3: Solve $c^k = (\mathcal{G} + \mathrm{diag}(w))^{-1}\Re(\mathcal{V}^* f)$,
4: Assign $w_j^k = \mu|c_j^k|^{-1}$.
5: **end for**
6: **return** c^∞

algorithm does not always converge, particularly if the initial guess $cb^\dagger = 1$ (i.e., $w_j^0 = \mu$) is too far away from the true $c^\dagger$. However, for practical purposes it is useful because it is simple and fast. In [13] convergence and sparsity results for an iteratively reweighted least squares algorithm for a related problem is shown, as well as generalizations to the counterpart ℓ^p problem for $p < 1$.

A common feature of with methods for solving (4.1) is the iterative application of the synthesis and analysis operators $\mathcal{V}$ and $\mathcal{V}^*$. For large-scale problems, it is crucial that these operations can be applied in a fast manner. As shown in the previous section and Theorem 3.5, we have a method to approximate $\mathcal{V}^* f$ that requires $\mathcal{O}(N \log N)$ operations. [1] An important point is that many of the algorithms for solving (4.1) only employ the combination of $\mathcal{V}$ and $\mathcal{V}^*$ as in (4.2), or at least it is possible to modify them to only employ that combination. We will take advantage of this feature. We note that, even though $\mathcal{G}$ will be very sparse, its complete computation is not computationally feasible.

[1] A similar argument can be used for the evaluation of $\mathcal{V}c(x)$ for points x on finite lattices (again by using USFFT algorithms).

An unattractive aspect of iteratively applying $\mathcal{V}$ and $\mathcal{V}^*$ when looking for a *sparse* solution $c^\dagger$ to (4.1), is that most of the elements in $c^\dagger$ will be zero, and thus since a lot of the computed values of $\mathcal{G}c = \mathcal{V}^*\mathcal{V}c$ for $c \approx c^\dagger$ will never be used, it seems unnecessary to compute them. Ideally, we would like to compute only the elements that we believe are relevant. However, to modify a fast algorithm so that it only computes relevant elements is in many cases difficult; a typical case of this is the FFT algorithm.

Real-valued formulation. In this section we describe an efficient method to find sparse representations of real-valued functions using Gaussian wave packets. We can do this if we assume that the basis functions that we work with exists in conjugated pairs. The case of complex-valued functions is in fact simpler to treat, since this assumption is not necessary. But as the principle is the same we satisfy with considering the real case.

Throughout the remainder of the paper we will assume that the set of linearly independent functions Γ is such that $(y, \eta, \alpha, \beta, \theta) \in \Gamma$ if and only if $(y, -\eta, \alpha, \beta, \theta) \in \Gamma$. Note that $\overline{\varphi_{y,-\eta,\alpha,\beta,\theta}} = \varphi_{y,\eta,\alpha,\beta,\theta}$. We introduce $\Gamma_{\mathrm{i}} = \{j : 1 \leq j \leq |\Gamma|/2\}$ and assume that the functions $\{\varphi_{\gamma_k}\}_{\gamma_k \in \Gamma}$ have been ordered so that

$$\overline{\varphi_{\gamma_{2j}}} = \varphi_{\gamma_{2j-1}}, j \in \Gamma_{\mathrm{i}}. \tag{4.4}$$

For simplicity we shall write φ_k in place of φ_{γ_k}. If the function $\mathcal{V}c$ is real-valued, then $c_{2j-1} = \overline{c_{2j}}$ and we can write

$$\mathcal{V}c = \sum_{j \in \Gamma_{\mathrm{i}}} \left(c_{2j-1}\varphi_{2j-1} + \overline{c_{2j}\varphi_{2j-1}} \right) = \sum_{j \in \Gamma_{\mathrm{i}}} 2\Re(c_{2j-1}\varphi_{2j-1}).$$

We define the support of c

$$\mathrm{supp}(c) = \{j : (c_{2j-1}, c_{2j}) \neq 0\}.$$

For convenience of the reader we include the following lemma.

LEMMA 4.1. *Let $\{\psi_j\}_{j=1}^N$, with $\psi_j \in L^2(\mathbb{R}^2)$ being complex-valued and having unit norm, and such that $\psi_{2j} = \overline{\psi_{2j-1}}$. Suppose that $f \in L^2(\mathbb{R}^2)$ is real-valued, and let*

$$G_{j,k} = \begin{pmatrix} \Re\langle\psi_{2j-1}, \psi_{2k-1}\rangle + \Re\langle\psi_{2j-1}, \psi_{2k}\rangle & -\Im\langle\psi_{2j-1}, \psi_{2k-1}\rangle + \Im\langle\psi_{2j-1}, \psi_{2k}\rangle \\ \Im\langle\psi_{2j-1}, \psi_{2k-1}\rangle + \Im\langle\psi_{2j-1}, \psi_{2k}\rangle & \Re\langle\psi_{2j-1}, \psi_{2k-1}\rangle - \Re\langle\psi_{2j-1}, \psi_{2k}\rangle \end{pmatrix}. \tag{4.5}$$

Construct the $2N \times 2N$ matrices

$$\widetilde{G} = \frac{1}{2} \begin{pmatrix} G_{1,1} & G_{1,2} & \dots & G_{1,N} \\ G_{2,1} & G_{2,1} & \dots & G_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N,1} & G_{N,2} & \dots & G_{N,N} \end{pmatrix},$$

$$G = \frac{1}{2}\left(\widetilde{G} + \widetilde{G}^{\mathrm{t}} \right), \tag{4.6}$$

and the vector $b = (b_1, b_2, \dots, b_N)^{\mathrm{t}}$,

$$b_j = (\Re\langle f, \psi_{2j-1}\rangle, \Im\langle f, \psi_{2j-1}\rangle). \tag{4.7}$$

Let $A : \ell^2_{2N} \mapsto L^2(\mathbb{R}^2)$ be defined by $Ax = \sum_{j=1}^{2N} x_j\varphi_j$, and let $c^\dagger$ be the solution to

$$\mathrm{argmin}_{c \in \mathbb{R}^{2N}} \frac{1}{2} c^{\mathrm{t}} G c - c^{\mathrm{t}} b + \mu \sum_{j=1}^{N} \sqrt{c_{2j-1}^2 + c_{2j}^2}. \tag{4.8}$$

If $x^\dagger$ is the solution to

$$\operatorname{argmin}_{x\in\ell^1_N} \frac{1}{2}\|Ax-f\|_2^2+\mu\|x\|_1,$$

then

$$x_j^\dagger = \begin{cases} c_j^\dagger + i\, c_{j+1}^\dagger, & \text{if } j \text{ is odd;} \\ c_{j-1}^\dagger - i\, c_j^\dagger, & \text{if } j \text{ is even.} \end{cases}$$

Proof. Since f is real valued and $\psi_{2j}=\overline{\psi_{2j-1}}$, $j=1,\dots,N$, it follows that $(A^*f)_{2j}=\overline{(A^*f)_{2j-1}}$. The assertion of the lemma then follows immediately from expanding $\|Ax-f\|_2^2$ and by decomposing x and the elements of A^*A into real and imaginary parts. ∎

LEMMA 4.2. *Given the same assumptions as in Lemma 4.1, let c be a solution to (4.8). To simplify the notation, introduce $d_j=\big(c_{2j-1},c_{2j}\big)$. Then, for $d_j\neq 0$ it holds that*

$$\Big((Gc-b)_{2j-1},(Gc-b)_{2j}\Big) = -\mu\frac{d_j}{\|d_j\|},$$

while for $d_j=0$ we have

$$\Big\|\Big((Gc-b)_{2j-1},(Gc-b)_{2j}\Big)\Big\| \le \mu,$$

and conversely, if the conditions above hold, then c is the solution to (4.8).

Proof. The proof will be done in a similar fashion as in [18]. We will need the notion of subdifferentials, cf. [25] for details. Let

$$h=\frac{1}{2}c^{\mathrm{t}}Gc-c^{\mathrm{t}}b+\mu\sum_{j=1}^N\sqrt{c_{2j-1}^2+c_{2j}^2}=\frac{1}{2}c^{\mathrm{t}}Gc-c^{\mathrm{t}}b+\mu\sum_{j=1}^N\|d_j\|.$$

Since h is convex, it follows that (4.8) has a unique solution. Since h is locally Lipschitz, a necessary and sufficient condition for c to be the unique solution is that $0\in\partial(h)$, where ∂h denotes the subdifferential of h, cf. [25, Theorem 3.2.5]. The subdifferential is

$$\partial\left(\sum_{j=1}^N\|d_j\|\right)=\Bigg\{u=(u_1,\dots,u_N)\,:u_j=d_j/\|d_j\|,\quad \text{if } d_j\neq 0,$$

$$u_j(1)^2+u_j(2)^2\le 1,\quad \text{if } d_j=0\Bigg\}. \tag{4.9}$$

The condition $0\in\partial h$ can therefore be written as

$$\Big((Gc-b)_{2j-1},(Gc-b)_{2j}\Big)+\mu\,u_j=0, \tag{4.10}$$

for all $1\le j\le N$ (with u_j as in (4.9)). For j such that $d_j\neq 0$ we thus obtain from (4.10) that

$$\Big((Gc-b)_{2j-1},(Gc-b)_{2j}\Big)=-\mu\,d_j/\|d_j\|,$$

and for j such that $d_j=0$ we have that $\|\big((Gc-b)_{2j-1},(Gc-b)_{2j}\big)\|\le\mu$, since $\|u_j\|<1$ in this case. ∎

To circumvent the computationally expensive application of the full synthesis ($\mathcal{V}$) and analysis ($\mathcal{V}^*$) in each iteration, we propose an alternative method described in Algorithm 2. The idea is

to work with *active sets* and exploit the fact that it is much easier to evaluate submatrices of the Gram matrix $\mathcal{G}$ (the same holds for its real counterpart G). That is, we wish to iteratively solve the minimization problem

$$\min_{\mathrm{supp}(c)\subset P} \frac{1}{2}c^{\mathrm{t}}Gc - c^{\mathrm{t}}b + \mu\sum_{j=1}^{|\Gamma_{\mathrm{i}}|}\sqrt{c_{2j-1}^2 + c_{2j}^2}, \quad P \subset \Gamma_{\mathrm{i}}, \tag{4.11}$$

where $P \in \{1, \ldots, |\Gamma_{\mathrm{i}}|\}$ is chosen to suit the data b. Note that the above can be formulated as a minimization over $\mathbb{R}^{2N'}$ for some $N' << N$, (compare with (4.8)).

Algorithm 2 ℓ^1-optimization using active sets and Gram matrices for a real-valued function f.

1: Let $Q^0 = \emptyset, \quad c^0 = 0, \quad m = 0$, use $\mathcal{V}^* f$ and compute b through (4.7).
and $R^0 = \left\{ j : \left\| \left(b_{2j-1}, b_{2j} \right) \right\| > \mu \right\}$.
2: **while** $R^m \neq \emptyset$ **do**
3: Let $m = m + 1$ and select a non-empty subset $S^m \subset R^{m-1}$ of new candidates for the representation.
4: Let $P^m = Q^{m-1} \cup S^m$.
5: Let c^m be the solution of

$$\min_{\mathrm{supp}(c)\subset P^m} \frac{1}{2}c^{\mathrm{t}}Gc - c^{\mathrm{t}}b + \mu\sum_{j=1}^{|\Gamma_{\mathrm{i}}|}\sqrt{c_{2j-1}^2 + c_{2j}^2}. \tag{4.12}$$

6: Let $Q^m = \mathrm{supp}(c^m)$.
7: Define

$$R^m = \left\{ j : \left\| \left((Gc^m - b)_{2j-1}, (Gc^m - b)_{2j} \right) \right\| > \mu \right\}.$$

8: **end while**
9: **return** c^m

The main advantage of Algorithm 2 is that the full synthesis $\mathcal{V}$ and analysis $\mathcal{V}^*$ operators are needed only comparatively few times. The analysis operation $\mathcal{V}^*$ is needed once during the initialization to compute b. After that, each of the operations is applied only once per iteration over m. For functions f that are approximated well by a sparse representation, it is possible to keep comparatively few terms in each of the sets P^m, and (in combination that G and the corresponding submatrices will be sparse) which makes it possible to find solutions to (4.12) much faster compared to methods that depend on applying the full operators $\mathcal{V}$ and $\mathcal{V}^*$ in each iterative step for solving (4.12).

LEMMA 4.3. *Algorithm 2 converges in finitely many steps to the solution of the full minimization problem*

$$\min_{c} \frac{1}{2}c^{\mathrm{t}}Gc - c^{\mathrm{t}}b + \mu\sum_{j=1}^{|\Gamma_{\mathrm{i}}|}\sqrt{c_{2j-1}^2 + c_{2j}^2}. \tag{4.13}$$

Proof. Define the function

$$h(c) = \frac{1}{2}c^{\mathrm{t}}Gc - c^{\mathrm{t}}b + \mu\sum_{j=1}^{|\Gamma_{\mathrm{i}}|}\sqrt{c_{2j-1}^2 + c_{2j}^2}. \tag{4.14}$$

We will show that $(\, h(q^m)\,)_m$ is a strictly decreasing sequence. By construction, it is clear that

$$\min_{\mathrm{supp}(c)\subset P^m} h(c) = \min_{\mathrm{supp}(c)\subset Q^m} h(c).$$

For $j \in S^{m+1}$, let

$$\tau_j = \left\| \Big((Gc^m - b)_{2j-1}, (Gc^m - b)_{2j} \Big) \right\| - \mu > 0, \tag{4.15}$$

where the positivity of τ_j follows from the definition of R^m. It is easily seen from the proof of Lemma 4.2 that

$$\min_{\mathrm{supp}(c)\subset Q^m} h(c) = \min_{\mathrm{supp}(c)\subset P^{m+1}} h(c) + \sum_{j\in S^{m+1}} \tau_j \sqrt{c_{2j-1}^2 + c_{2j}^2},$$

because c^m is the unique minimizer of both expressions and $c_j^m = 0$ for $j \in S^{m+1}$. In adition, it is clear that

$$\min_{\mathrm{supp}(c)\subset P^{m+1}} h(c) + \sum_{j\in S^m} \tau_j \sqrt{c_{2j-1}^2 + c_{2j}^2} \;\geq\; \min_{\mathrm{supp}(c)\subset P^{m+1}} h(c). \tag{4.16}$$

Now, the only way to get equality in (4.16) is if the solution $\widetilde{c}^{m+1}$ of the left-hand side of (4.16) is such that $\widetilde{c}_j^{m+1} = 0$ for all $j \in S^m$. But in this case $\widetilde{c}^{m+1} = c^m$, and Lemma 4.2 then implies that $\tau_j = 0$ for all $j \in S^{m+1}$, which contradicts the construction. Hence, we must have

$$\min_{\mathrm{supp}(c)\subset P^m} h(c) \;>\; \min_{\mathrm{supp}(c)\subset P^{m+1}} h(c).$$

The algorithm converges in finitely many steps because there are only finitely many subsets of Γ_i and $h(c^m)$ is decreasing with m. □

One way of solving (4.12) is to use Algorithm 1 with the weight update on line 4 replaced by $w_j^k = \mu(c_{2j-1}^2 + c_{2j}^2)^{-1/2}$.

ℓ_p-formulation. Just as in [13], it is interesting to change the ℓ^1 penalty in (4.8) to an ℓ^p-type of penalty with $p < 1$, since this may lead to a sparser solution. This change however leads to nonconvex versions of (4.8) for which there is no guarantee that the algorithms will not get stuck at local minima. It is still interesting to look for formulations of (4.8) that may promote sparser solutions. The following lemma provides as example of such formulation.

LEMMA 4.4. *Given the same assumptions as in Lemma 4.1, define*

$$\Upsilon_{p,\mu}(t) = \frac{\mu|t|}{1 + (1-p)(|t|/\mu)^{1-p}}, \tag{4.17}$$

and let c be a solution to

$$\min_c \frac{1}{2} c^{\mathrm{t}} G c - c^{\mathrm{t}} b + \sum_{j=1}^{N} \Upsilon_{p,\mu}\left(\sqrt{c_{2j-1}^2 + c_{2j}^2} \right). \tag{4.18}$$

Then

$$\left\| \Big((Gc - b)_{2j-1}, (Gc - b)_{2j} \Big) \right\| \leq \mu.$$

Proof. The proof follows as in the proof of Lemma 4.2 by noting that

$$\left| \frac{d}{dt} \Upsilon_{p,\mu}(t) \right| < \mu, \qquad t \neq 0,$$

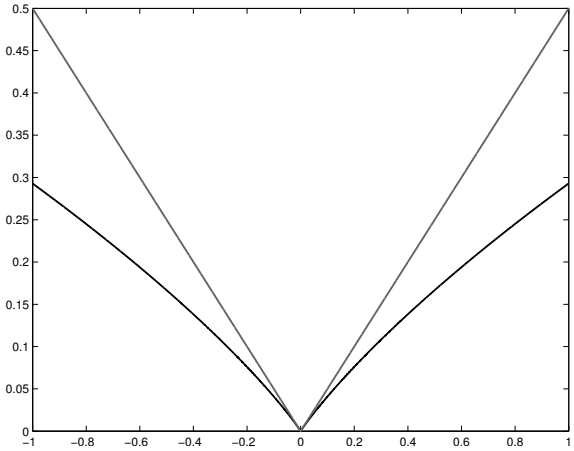

FIG. 4. *The function* $\Upsilon_{p,\mu}(t)$ *for* $p = 0.5$, $q = 2$, *and* $\mu = 0.5$ *in black, and the function* $t \to |t|$ *in gray.*

and that for $t = 0$ the sub-differential of $\Upsilon_{p,\mu}$ coincides with the sub-differential of the function $t \mapsto |t|$. □

Again, one way of solving (4.18) is to use Algorithm 1 with the weight update on line 4 replaced by

$$w_j^k = \left(\Upsilon_{p,\mu} \left(\sqrt{c_{2j-1}^2 + c_{2j}^2} \right) \right)^{-1}. \tag{4.19}$$

In the sequel we will show the results of numerical simulations, similar to those done for (4.8), that show good sparsity and computational properties of the modified framework.

5. Numerical examples.

5.1. Missing data. As an example of how Gaussian wave packets perform we use the following missing traces example from the *SPARCO toolbox* [4]. The data consists of synthetic seismic data traces, given on a rectangular lattice, but where some of the data columns are missing. The missing data can be modeled by introducing the multiplication with a characteristic function χ. The proposed ℓ_1 model in the SPARCO toolbox for this problem is slighly different from the one in (4.1);

$$\text{argmin}_{c \in \ell^2(\Gamma_\epsilon)} \|\chi(\mathcal{V}c - f)\|_1 \quad \text{subject to } \|c\|_1 \leq \sigma, \tag{5.1}$$

for some choice of regularization parameter σ. Above, χ is one where the data is available and zero where it is missing. The factor χ will destroy the analytic expressions for the element of the Gram matrix, since the Gram matrix in the case (5.1) will be of the form $\mathcal{G}_\chi = \mathcal{V}^* \chi A$.

To solve (5.1) we make use of the software package *SPGL1*, [32, 5], which employs iterative applications of $\mathcal{V}$ and $\mathcal{V}^*$. For illustrative reasons, we want to compare our results with Gaussian wave packet representations with competing representations. We compare against a curvelet representation, i.e., the *USFFT-based version*, as it according to the authors [10, p 10] yields "the most faithful discretization of the continuous definition". For reasons of simplicity, we have also used the *MATLAB* implementation, although it probably is not as optimized with regards to computational speed as the *C++* version.

The results after applying *SPGL1* to the Gaussian wave packet, and curvelet[2], representions, respectively, are depicted in Figure 5, with the Gaussian wave packet result to the left and the

[2]we have included weight so that the coefficients in the right term of (5.1) corresponds to basis functions with unit norm

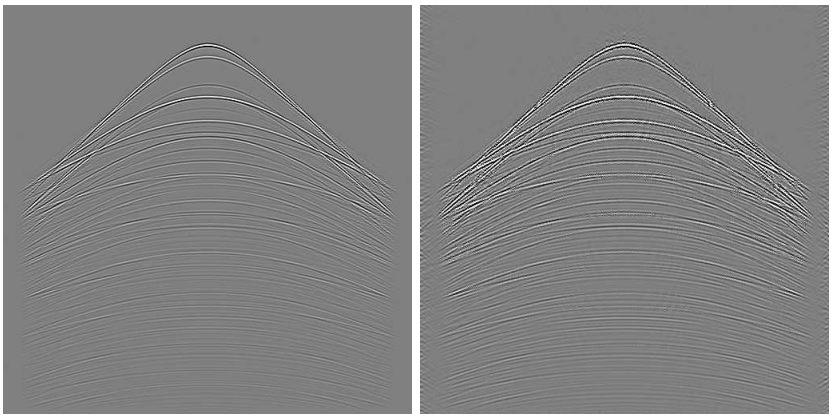

FIG. 5. *Results for the missing traces example from the Sparco toolbox. To left the result from using Gaussian wave packets, and to the right the result from using curvelets.*

TABLE 5.1
Table of results corresponding to Figure 6

Method	# Coeffients	Time(s)	RMS Error	# Iterations
GWP ℓ_1	597	53	0.05	12
GWP $\ell_{0.1}$	214	39	0.06	12
Curvelet ℓ_1	675	4001	0.08	512

curvelet result to the right. Note that the images are thresholded to be in the range $[-0.0001, 0.0001]$ in order to better see the details.

The same parameters were used for the both representations, with $\sigma = 0.03$. The Gaussian wave packet representation required 83 synthesis operations and 56 analysis operations, yielding a total time of 528 seconds. The corresponding number for the curvelet representation were 188 synthesis operations and 122 analysis operations, with a total time of 1453 seconds.

Since the Gaussian wave packet implementations are optimized, and the curvelet implementation is not, it is not fair to compare the times directly. However, we not that the Gaussian wave packet representations required fewer operations for convergence. More importantly, judging from Figure 5, we deem that result wise, the usage of Gaussian wave packets is superior to the usage of curvelets for this missing data example.

5.2. Sparse image representations. In this section we provide some numerical experiments regarding the compression of functions in two variables. To begin with, we will use Algorithm 2 to solve (4.1). Algorithm 1 is then used to solve the sub-problems (4.12).

To measure the quality of our approach we compare our results with the ones obtained by using a curvelet representation. As the way we solve to (4.1) makes use of the analytic properties of the Gaussian wave packets, we need to use a different numerical tool to solve the counterpart of(4.1) for the curvelet representation. Using the simple iterative thresholding algorithm, cf. (4.3), or a version of Algorithm 1 would not be fair, as it would be very computationally expensive. To make a more fair comparison, we have used the *SPG-LASSO* routine of the *SPGL1* package [5, 32], as one of the current state of the arts methods for solving large scale problems if this kind. The formulation used in the *SPG-LASSO* routine differs slightly from that of (4.1). If A is the matrix that contains the "basis functions" that we wish to use, then, for a given choice of parameter τ, *SPG-LASSO* solves

$$\text{argmin}\|Ax - f\|_2^2 \quad \text{subject to } \|x\|_1 \leq \tau. \tag{5.2}$$

It is easy to see that there is a one–to–one relation between the parameter τ above the the parameter

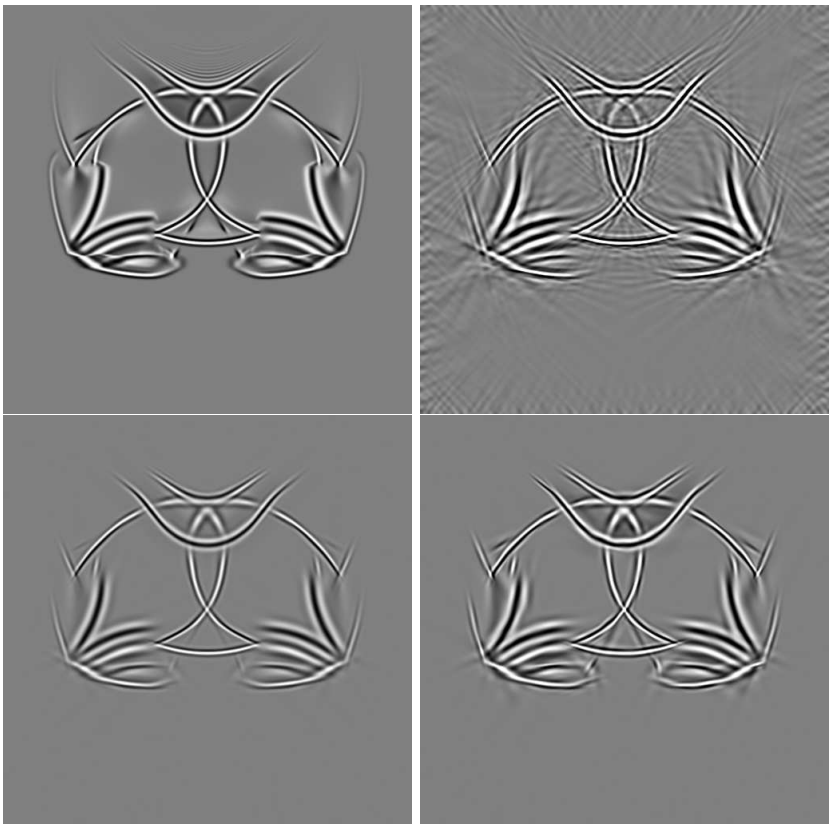

FIG. 6. *Compression comparison between Gaussian wave packets and curvelets. The original image has a range of [-0.5 0.5], and all images are shown in the range [-0.25 0.25]. Top left: original image, Top right: curvelet (ℓ_1) representation with 675 coefficients, Bottom left: Gaussian wave packet (ℓ_1) representation with 597 coefficients, Bottom right: Gaussian wave packet ($\ell_{0.1}$) representation with 214 coefficients*

TABLE 5.2
Table of results corresponding to Figure 7

Method	# Coeffients	Time(s)	RMS Error	# Iterations
GWP ℓ_1	1855	132	0.10	20
GWP $\ell_{0.1}$	901	87	0.11	20
Curvelet ℓ_1	2054	3315	0.10	434

μ in the formulation

$$\operatorname{argmin}\|Ax - f\|_2^2 + \mu\|x\|_1, \tag{5.3}$$

cf. [32]. In Figures 6, 7 we show some results of compression. In the top left Figures 6, Figure 7 we show two *snap shots* from a two dimensional wave propagation problem. Given a heterogeneous velocity model, we have numerically solved the wave equation with a time dependent point source by a standard finite difference method. The two snap shots in the top left of Figures 6, 7 contain the numerical solutions for two time instances; where Figure 6 corresponds to the earlier of the two time instances. The heterogeneous velocity model has been chosen to create a complex wave pattern, even for very simple point source functions. For the simulations presented here, we have used a one-dimensional Gaussian wave packet as a point source. In the bottom left panels of Figures 6, 7 we show the results obtained by solving 4.1. In the top right panel the counterpart results using curvelets are depicted. We have chosen the parameters μ in (4.1) and τ in (5.2) so

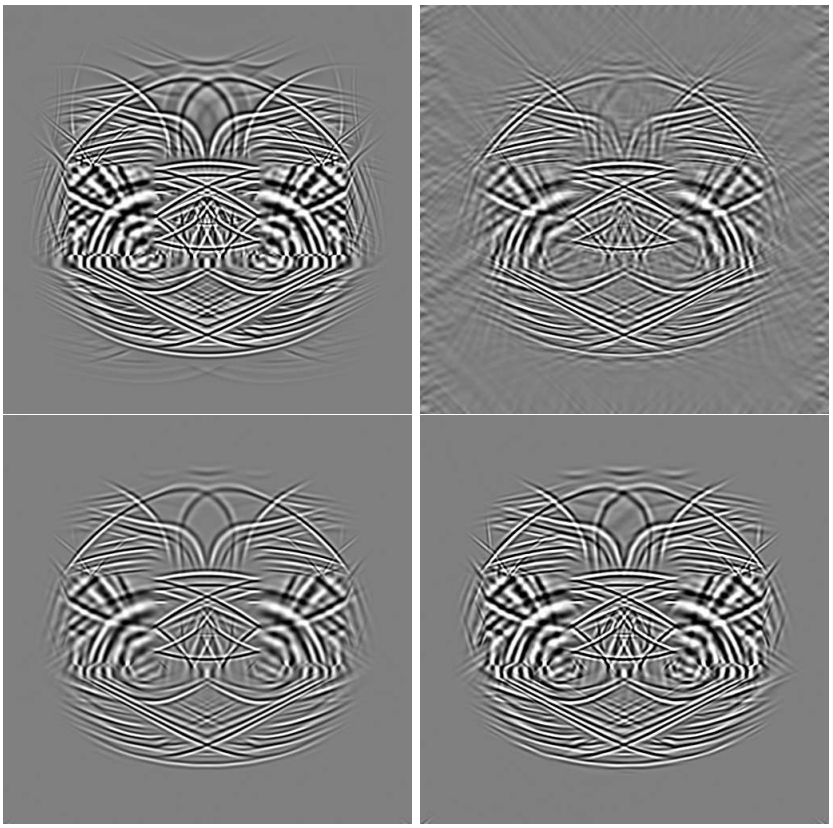

FIG. 7. *Compression comparison between Gaussian wave packets and curvelets. The original image has a range of [-0.5 0.5], and all images are shown in the range [-0.25 0.25]. Top left: original image, Top right: curvelet (ℓ_1) representation with 2054 coefficients, Bottom left: Gaussian wave packet (ℓ_1) representation with 1855 coefficients, Bottom right: Gaussian wave packet ($\ell_{0.1}$) representation with 901 coefficients*

that the solutions have about the same number of non-zero coefficients (number of basis function). In both the examples, we have allowed for slightly more curvelet coefficients than Gaussian wave packet coefficients.

In addition to the two ℓ_1 formulations (4.1) and (5.2), we explore the case of ℓ_p, $p < 1$ in the formulation (4.18). We have used Algorithm 2 with (4.12) replaced by (4.18), and Algorithm 1 with the weights w_j^k in step 4 replaced by the weights given by (4.19), so solve the subproblems of the form (4.18) corresponding to line 5 in Algorithm 2. The results for the same choice of μ for the Gaussian wave packet solutions to (4.1) are shown in the bottom right of Figures 6, 7. For these cases we have used $p = 0.1$ (where p again refers to the ℓ_p-like formulation (4.18)).

In Tables 5.1, 5.2, we show the number of coefficients used in the representation, the computational time, the standard deviation between the original and the compressed representations, and the number of times that the either one of the analysis and synthesis operators, for the respective cases, has been applied. For the Gaussian wave packets, a total of 12 and 20 applications of either $\mathcal{V}$ and $\mathcal{V}^*$ has been applied, for the respective examples. This corresponds to 6 and 10 "outer" iterations of Algorithm 2, respectively. For the curvelet representations, substantially more applications of the corresponding synthesis and analysis operations are required, 434 and 512 respectively for the two examples considered. The large difference in the required number of operator applications, makes the curvelet approach subtantially slower[3]. We recognize that the times presented in Tables

[3]However, we note that we would expect those numbers to be substantially higher if, e.g., an iterative soft thresholding method would have been used instead of *SPG-LASSO* would have been used.

TABLE 5.3
Parameters used to construct the wave packet parameter set Γ used in the simulations of the section.

η_k			28,81	40,74	57,62	81,49	115,24	162,97
			28,81	40,74	57,62	81,49	115,24	162,97
	14,41	20,37	28,81	40,74	57,62	81,49	115,24	162,97
α_k			1	1	1	1	1	1
			2	2	2	2	2	2
	1	1	1	1	1	1	1	1
β_k			4	4	4	4	4	4
			2	2	2	2	2	2
	2	2	2	2	2	2	2	2

5.1, 5.2 for the curvelet cases, could have been less if the *C++* versions would have been used.

However, although it seems like the Gaussian wave packet approach seems to perform superior to the curvelet approach, the primary motivation for the comparisons in this section concerns the quality of the compressed representations. Here we note that Gaussian wave packet results seems to agree visually much better with original data than the curvelet representations. Just as in Figure 3, we see that the "tails" of the curvelets produce disturbing artifacts. We also note large boundary artifacts for the curvelets.

For the simulations with Gaussian wave packets we have used the parameter set in (3.9) with $\theta_\ell = \ell\pi/32$, $\mathbb{Z} \ni \ell = [0, 32)$, and $\eta_{k,\ell} = \eta_k(\cos(\theta_\ell), \sin(\theta_\ell))$, $\alpha_{k,\ell} = \alpha k$, and $\beta_{k,\ell} = \alpha k$, with η_k, α_k and β_k chosen as in Table 5.3. We have also used $\epsilon = 1e-3$, and $\sigma = 1.0$. The simulation was made using 512x512 data sets.

It is interesting to note that using $\ell_{0.1}$ ($p = 0.1$ in (4.18)) instead of ℓ_1 ($p = 1$ in (4.18))in the formulation for the Gaussian wave packets we obtain fairly good results with only half as many coefficients. A drawback is that the reconstruction has slightly more distinctive artifacts. On the other hand, the $p = 0.1$-case has better amplitude agreement than the $p = 1$-case. However, this amplitude correction problem can easily be resolved using $p = 1$ as a variable selection procedure to select which coefficients to use, and thereafter use ordinary least squares with the selected coefficients.

One important application that we have in mind is the decompoition of data originating from measurements of solutions of the wave equation (e.g., seismic data). In this case the primary interest is not in the actual compression ratio that can be achieved but in the sparsity that can be used in the analysis of the data. In this way, the Gaussian wave packet representations are natural, as it is easy to relate physical properties to the reconstructions.

6. Conclusions. We have shown that Gaussian wave packets are useful for representing functions with wave-like features. They satisfy several appealing invariance properties that are useful for many reasons, for example they provide analytic formulas for the Gram matrix. For the continuous transform, we have provided an inversion formula and illustrated the good decay properties associated with Gaussian wave packets. Moreover, we have shown how to construct fast algorithms for a discrete transform, and how to choose precision parameters associated with it. Using the analytic properties of the Gaussian wave packets to compute elements of the Gram matrix directly, we have constructed a new numerical algorithm for obtaining sparse recovery for very redundant representations, without having to compute full analysis and synthesis operations.

REFERENCES

[1] F. Andersson, M.V. de Hoop, H.F. Smith, and G. Uhlmann. A multi-scale approach to hyperbolic evolution equations with limited smoothness. *Comm. Partial Differential Equations*, 33(4-6):988–1017, 2008.

[2] F.G. Meyer andl R.R. Coifman. Brushlets: A tool for directional image analysis and image compression. *Applied and Computational Harmonic Analysis*, 4:147–187, 1997.

[3] J.-P. Antoine, R. Murenzi, P. Vandergheynst, and S.T. Ali. *Two-dimensional wavelets and their relatives*. Cambridge University Press, 2004.
[4] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and Ö. Yılmaz. Sparco: A testing framework for sparse reconstruction. Technical Report TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver, October 2007.
[5] E. van den Berg and M.P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007.
[6] G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comput. Harmon. Anal.*, 2(4):363–381, 1995.
[7] A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51:34–81, 2008.
[8] V. Bucha. Gaussian packet prestack depth migration. part 3: Simple 2-d models: Seismic waves in complex 3-d structures. Technical Report 19, Charles University, 2009.
[9] E.J. Candès and D.L. Donoho. Continuous curvelet transform. II. Discretization and frames. *Appl. Comput. Harmon. Anal.*, 19(2):198–222, 2005.
[10] E.J. Candès, D.L. Laurent, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Model. Simul.*, 5(3):861–899, 2006.
[11] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.
[12] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.
[13] I. Daubechies, M. Fornasier, and C. S. Gntrk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.
[14] I. Daubechies and M. Defriseand C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.
[15] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14(6):1368–1393, 1993.
[16] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.
[17] H.G. Feichtinger and T. Strohmer. *Gabor analysis and algorithms: theory and applications*. BirkHuser, 2003.
[18] J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Trans. Inform. Theory*, 50(6):1341–1344, 2004.
[19] K. Gröchenig. *Foundations of time-frequency analysis*. Birkhäuser, 2001.
[20] A. Grossmann, J. Morlet, and T. Paul. Transforms associated to square integrable group representations. I. General results. *J. Math. Phys.*, 26(10):2473–2479, 1985.
[21] K. Guo and D.Labate. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.*, 39(1):298–318, 2007.
[22] N.R. Hill. Prestack gaussian-beam depth migration. *Geophysics*, 66:1240–1250, 2001.
[23] L. Klimeš. Notes on summation of gaussian beams and packets: Seismic waves in complex 3-d structures. Technical Report 14, Charles University, 2004.
[24] A.K. Louis, P. Maass, and A. Rieder. *Wavelets: theory and applications*. Wiley, 1997.
[25] M.M. Mäkelä and P. Neittaanmäki. *Nonsmooth optimization*. World Scientific Publishing Co. Inc., River Edge, NJ, 1992. Analysis and algorithms with applications to optimal control.
[26] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.
[27] E.P. Simoncelli and W.T. Freeman. The steerable pyramide: a flexible architechture for multi-scale derivate computation. *IEEE Conf. Imag. Proc.*, 3:444–447, 1995.
[28] H.F. Smith. A parametrix construction for wave equations with $C^{1,1}$ coefficients. *Ann. Inst. Fourier (Grenoble)*, 48(3):797–835, 1998.
[29] R. Tibshirani T. Hastie and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.
[30] A. Beckand M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
[31] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.
[32] E. van den Berg and M.P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.
[33] R.-S. Wu, Y. Wang, and M. Luo. Beamlet migration using local cosine basis. *Geophysics*, 73(5):S207–S217, 2008.